В. М. Антонов

Сучасні комп'ютерні мережі

"МК-Прес"
Київ, 2005

ББК 32.97
А72
УДК 681.324

Рецензенти:
Попов Ю. Д. — доктор технічних наук, професор Київського національного університету ім. Тараса Шевченка;
Сбітнєв А. І. — доктор технічних наук, професор Київської національної Академії оборони України;
Ріппа С. П. — доктор технічних наук, професор Національної Академії Державної Податкової Служби України.

Антонов В. М.
А72 Сучасні комп'ютерні мережі. — К.: "МК-Прес", 2005. — 480 с., іл.
ISBN 966–96415-5-1

У цьому фундаментальному виданні розглянуто широке коло питань стосовно сучасних мережевих інформаційно-комунікаційних технологій: концепції та принципи побудови мереж; класифікація мережевого апаратного і програмного забезпечення; функціональне призначення протоколів тощо. Подані основи побудови корпоративних, нейронних та віртуальних мереж. Розглянуті засоби захисту у мережах. Крім того, приділена увага таким перспективним технологіям як дистанційна освіта в Internet, мови програмування для Internet, Web-дизайн тощо.

Це видання можуть використовувати як студенти вищих навчальних закладів, так і спеціалісти в галузі нових інформаційних технологій.

ББК 32.97

Антонов Валерій Миколайович

Сучасні комп'ютерні мережі

Головний редактор: Ю. О. Шпак
Комп'ютерна верстка: Ю. О. Шпак
Дизайн обкладинки: М. В. Шашкова

Підписано до друку 01.03.2005. Формат 70 × 100 1/16.
Папір газетний. Друк офсетний. Ум. друк. л. 38,9. Обл.-вид. л. 28,4.
Тираж 1300 екз. Замовлення № 5-360

ПП Савченко Л.О., Україна, м.Київ, тел./ф.: (044) 517-73-77; e-mail: info@micronika.com.ua.
Свідоцтво про внесення суб'єкта видавничої справи до Державного реєстру видавників, виготівників та розповсюджувачів видавничої продукції: серія ДК №51582 від 28.11.2003 р.

Надруковано в ЗАТ "ВІПОЛ". 03151, м. Київ, вул. Волинська, 60

ISBN 966–96415-5-1

Присвячую моїй улюбленій
онучці Амірі та онуку Юнесу

"Щоб полонити всіх, думка, як і жінка, має бути і розумна, і вродлива"

(Народна мудрість)

"У словах не опишеш усі барви світу.
У розмовах не висловиш всієї глибини мудрості.
Той, хто судить за словами, згубить себе.
Той, хто прив'язаний до слів, зіб'ється зі шляху"

(Умень)

Зміст

Подяки

"Наукою нехай займуться мужні,
Щоб всім останнім вистачило світла"
(Шекспір)

Моїй дружині — за неймовірну терплячість до моєї постійної відсутності у повсякденному родинному житті та спілкуванні і "зникненню" на довгий час роботи над рукописом у віртуальний світ таких непрозорих і карколомних комп'ютерних понять як мережеві технології, які тільки те й вміють, що залишати не виліковуванні "рани" у справі виховання третього покоління наших родинних нащадків.

Сбітнєву Анатолію Івановичу — за велику толерантність до мене на межі людського комфортного спілкування, за "активацію" та стимуляцію у мене на різних етапах написання книги прискіпливого аналізу та синтезу таких нечітких і "розмитих" понять як комп'ютерні мережі та сучасні інформаційні технології, а також за можливість вільно користуватись його безмежною за глибиною та широтою — інтелектуальністю з життєвих і професійних проблем.

Студентам та аспірантам факультету кібернетики Київського національного університету імені Тараса Шевченка — за активну і щиру співпрацю у підготовці тексту книги та перевірці прикладів у рукопису.

Керівництву Університету — за надану мені неймовірну але цікаву можливість на лекціях і практичних заняттях перевіряти на студентах нові ідеї, думки, технології стосовно питань, що включені до книги.

Вступ

"Немає в світі мук сильніше муки слова"
(Надсон С. Я.)

Ідея об'єднання комп'ютерів у мережу для спільної обробки даних виникла через невеликий період часу після того, як розпочалося інтенсивне використання комп'ютерів. Головна ідея об'єднання комп'ютерів у мережу — зробити ресурси одного комп'ютера досяжним для іншого і забезпечити засоби обмеження доступу до розподілених ресурсів, а також забезпечити обмін даними між користувачами, що працюють на різних комп'ютерах.

Використання сучасних комп'ютерних та інформаційних технологій відкриває нові можливості у сфері менеджменту, маркетингу, банківській та фінансовій діяльності, мультимедіа, а також у сфері наукових досліджень, освіти та ефективної підготовки кадрів. На сучасному етапі актуальною потребою є об'єднання комп'ютерних ресурсів у мережу передачі даних, яка підтримує поточні та майбутні інформаційні потреби. Витрати на організацію та підтримку функціонування обчислювальної мережі у повній мірі окупаються перевагами, що пропонуються новими мережевими технологіями.

Ефективна організація функціонування комп'ютерних (обчислювальних) мереж — задача, яка у теперішній час хвилює широке коло користувачів. По своїй практичній діяльності автору даної роботи найбільш близькі питання організації і застосування обчислювальних мереж у навчальному процесі. Використання ефективно працюючої мережі в автоматизованому навчальному класі дозволяє значно підвищити якість учбового процесу за рахунок прискорення доступу та збільшення числа активних робочих місць, а також за рахунок створення систем дистанційного навчання. Налагодження програмного забезпечення, оптимальна організація передачі даних, грамотна та ефективна диспетчеризація — це те, що дозволяє використовувати в сучасних комп'ютерних класах найновіші автоматизовані учбові курси, експертні системи, інтелектуальні автоматизовані робочі місця викладача та учня.

Мережеві технології забезпечують інтеграцію центральних та локальних систем в єдину логічну мережу. Інтеграція систем робочих груп з системами на основі архітектури мікро-, міні- та великих ЕОМ дозволяє скоротити капітальні вкладення в ці системи та зробити їх ресурси доступними всім користувачам мережі. Користувачі можуть здійснювати доступ до даних, прикладних програм та обчислювальних ресурсів цих ЕОМ так же легко як і до локально-глобального серверу. Канали зв'язку регіональних комп'ютерних мереж дозволяють об'єднувати в одну мережу віддалені одна від одної системи, також забезпечуючи прозорий доступ до ресурсів. Комп'ютерні мережі органічно вписуються в розподілені інформаційні процеси, що є характерною ознакою для сучасних інформаційно-комунікаційних систем.

Основні скорочення, використані в книзі

АЛП	арифметико-логічний пристрій
АРМ	автоматизоване робоче місце
АС	автоматизована система
АСУ	автоматизована система управління
БД	база даних
БЗ	база знань
БМ	база моделей
ВІС	велика інтегральна схема
ГОМ	головна обчислювальна машина
ДАП	двоноправлена асоціативна пам'ять
ЕОМ	електронно-обчислювальна машина
ЕП	електронна пошта
ЕС	експертна система
ІОМ	інформаційно-обчислювальна мережа
ІС	інтегральна схема
КТЗ	комплекс технічних засобів
ЛОЗП	локальний оперативно запам'ятовуючий пристрій
ЛОМ	локальна обчислювальна мережа
МА	мережевий адаптер
МІК	мережева інтерфейсна карта
МПД	мережа передачі даних
НМ	нейронна мережа
НФ	непрограмуючий фахівець
ОЗП	оперативний запам'ятовуючий пристрій
ООП	об'єктно-орієнтоване програмування
ОПР	особа, яка приймає рішення
ОС	операційна система
ОТ	обчислювальна техніка
ПЕОМ	персональна електронно-обчислювальна машина
ПЗ	програмне забезпечення
ПЗА	персональний засіб автоматизації
ПЗП	постійний запам'ятовуючий пристрій
ПК	персональний комп'ютер
ПП	прикладна програма
ПР	прийняття рішення
ПТЗ	програмно-технічні засоби
ПТЗА	програмно-технічні засоби автоматизації
РС	робоча станція
СППР	система підтримки прийняття рішення
СУБД	система управління базами даних
СУБЗ	система управління базами знань
СУБМ	система управління базами моделей
ТЗ	технічне забезпечення; технічне завдання
ЦП	центральний процесор

3D	3-Dimensions — тривимірний
API	Application Programming Interface — інтерфейс прикладного програмування
ATM	Asynchronous Transfer Mode — асинхронний режим передачі
BBS	Bulletin Board System — електронна дошка оголошень
CGI	Common Gateway Interface — загальний шлюзовий інтерфейс
CSA	Client-Server Architecture — архітектура "клієнт-сервер"
DNS	Domain Names System — система доменних імен
FR	Frame Relay — ретрансляція кадрів
GII	Global Information Infrastructure — глобальна інформаційна інфраструктура
GUI	Graphical User Interface — графічний інтерфейс користувача
HTTP	Hypertext Transport Protocol — протокол передачі гіпертексту
IP	Internet Protocol — протокол Internet
IT	Information Technologies — інформаційні технології
LAN	Local Area Network — локальна мережа
MAN	Metropolitan Area Network — регіональна мережа
NIC	Network Interface Card — мережева інтерфейсна карта
OSI	Open System Interconnection — взаємодія відкритих систем
SMTP	Simple Mail Transfer Protocol — протокол передачі поштових повідомлень
SNA	System Network Architecture — системна мережева архітектура
TCP	Transmission Control Protocol — протокол керування транспортуванням
UDP	User Datagram Protocol — протокол дейтаграм користувача
URL	Uniform Resource Locator — уніфікований покажчик інформаційного ресурсу
VLAN	Virtual Local Area Network — віртуальна локальна мережа
WAN	Wide-Area Network — глобальна мережа
WWW	World Wide Web — "всесвітня павутина"

Частина I

Загальна теорія мереж

Розділ 1

Концепції та принципи побудови мереж

"Читати без роздумів — шкідливо.
Роздум без читання — небезпечно."
(Народна мудрість)

1.1. Комп'ютерні технології обробки даних

Автоматизовані технології переробки інформації — це цілеспрямовані процеси обробки даних за допомогою сучасних засобів автоматизації з використанням централізованого, децентралізованого і розподіленого функціонально-технологічного підходу.

Централізована обробка даних характеризується наявністю одного головного засобу автоматизації (Mainframe, Host-комп'ютер), який використовує спільнота користувачів (рис. 1.1).

Рис. 1.1. Централізована обробка даних: К — користувач (i = 1, 2 ..., n)

Децентралізована обробка даних характеризується наявністю багатьох головних засобів автоматизації і багатьох спільнот користувачів (рис. 1.2).

Розподілена обробка даних — це обробка даних одного користувача одним персональним засобом автоматизації (Laptop, palmtop, desktop, мережевий комп'ютер, X-термінал, PDA, ...), які за допомогою засобів зв'язку об'єднані в єдину мережу на основі сервер-комп'ютера (рис. 1.3).

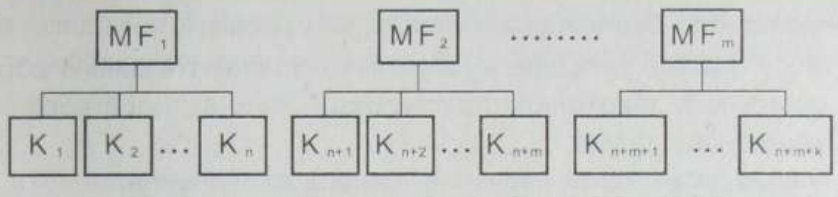


Рис. 1.2. Децентралізована обробка даних: MF = Mainframe

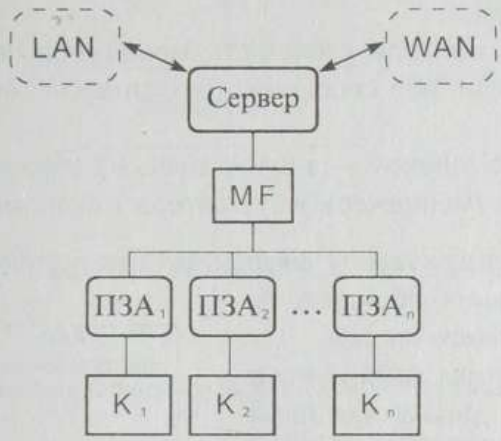


Рис. 1.3. Розподілена обробка даних або технологія клієнт-сервер:
ПЗА — персональний засіб автоматизації; LAN — локальна мережа; WAN — глобальна мережа

Автоматизоване робоче місце (АРМ) — умовна назва організаційно-технологічних, економіко-методичних, технічних і програмних засобів для вирішення завдань (реалізації функцій) не програмуючого фахівця в реальному часі із застосуванням інформаційного, функціонального, методологічного, алгоритмічного, математичного, лінгвістичного, методичного, правового та кадрового забезпечення. Таким чином, АРМ — це ідеологія автоматизації функцій будь-якого фахівця з використанням персональних засобів обробки інформації (рис. 1.4).

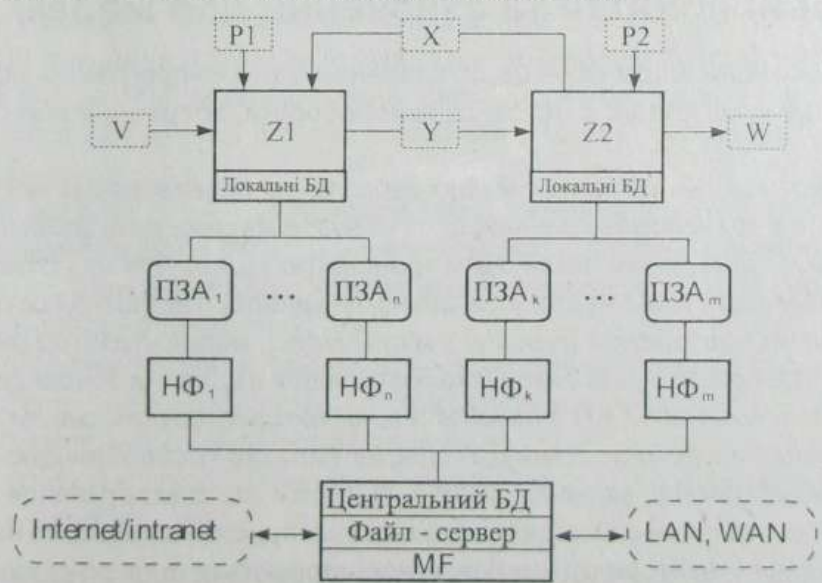


Рис. 1.4. АРМ-технологія: НФ — не програмуючий фахівець; БД — банк даних; Y — керуючі дії; X — зворотний зв'язок; V – вхідна інформація; W — вихідна інформація; P1, P2 — перешкоди; Z1 — керуюча система; Z2 — система, якою керують

АРМ складається з наступних загальних трьох блоків:

- блок вводу, аналізу, контролю, коригування, впізнання і первинної обробки даних;
- блок організації, зберігання, структуризації, пошуку, перетворення, інтерпретації, модифікації даних;
- блок виводу, візуалізації, навчання і самовдосконалення.

Класи АРМ (за повнотою функцій, що реалізуються) поділяються на:

- однофункціональні;
- функціональні (бухгалтера, економіста, менеджера тощо);
- багатофункціональні або споріднені (бухгалтера + економіста, менеджера + банкіра і тощо);
- інтегровані або комплексні — з орієнтацією на автоматизацію функцій фахівців одного об'єкта (менеджера + бухгалтера + економіста + банкіра + і тощо).

АРМ можуть бути реалізовані у вигляді: систем підтримки і прийняття рішень; експертних, інтелектуальних, гібридних, генетичних, нейронних та інших систем.

Основні характеристики АРМ: багатократне використання в динамічних (змінних) умовах, надійність, секретність, захищеність, тиражність (масовість), ефективність, продуктивність, динамічна (генетична) змінність алгоритму, відкритість, інтелектуальність та експертність.

Складність (вартість) реалізації АРМ підвищується у три рази при переході з класу в клас (рис. 1.5.).

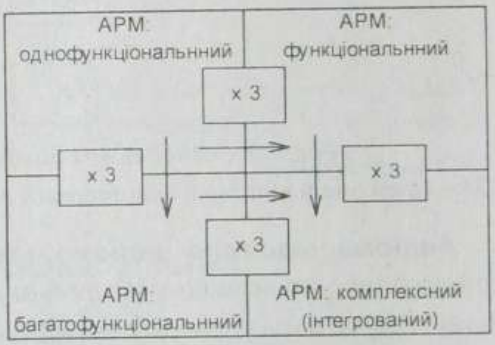


Рис. 1.5. Вартість реалізації АРМ

1.2. Основні поняття та означення для мереж

Обчислювальна мережа — це об'єднання (технічно-програмно-інформаційно) двох або більше комп'ютерів з метою спільної обробки, зберігання або передачі даних.

Шлюз (*gateway*) — це пристрій, що забезпечує перетворення несумісних між собою протоколів транспортного рівня; це спецвузол мережі, який забезпечує зв'язок мереж. Шлюзи можуть також виконувати трансляцію протоколів на чотирьох верхніх рівнях базової моделі OSI. Часто у загальному випадку так позначають сукупність таких пристроїв як повторювач (*repeater*), міст (*bridge*), маршрутизатор (*router*).

Шлюзи — це пристрої, що використовуються для з'єднання зовсім різних мереж, наприклад LAN з WAN або LAN з mainframes, які використовують зовсім інші або абсолютно несумісні протоколи обміну. У даному випадку треба повністю перетворювати весь потік інформації, включаючи коди, формати, методи керування тощо.

Приклад підключення у складній мережі різних пристроїв показано на рис. 1.6.

Мости, маршрутизатори та шлюзи використовуються для об'єднання у єдину мережу декількох різнорідних мереж, які використовують різні протоколи обміну нижнього рівня, тобто вони забезпечують так звану "прозорість" мережі для протоколів вищого рівня. Ці пристрої повинні бути дещо складніші за трансівери (*transeiver*), по-

вторювачі та концентратори, тому що повинні здійснювати складну обробку інформації і тому їх у більшості випадків реалізують на базі спеціальних ПК або робочих станцій (*work stations*).

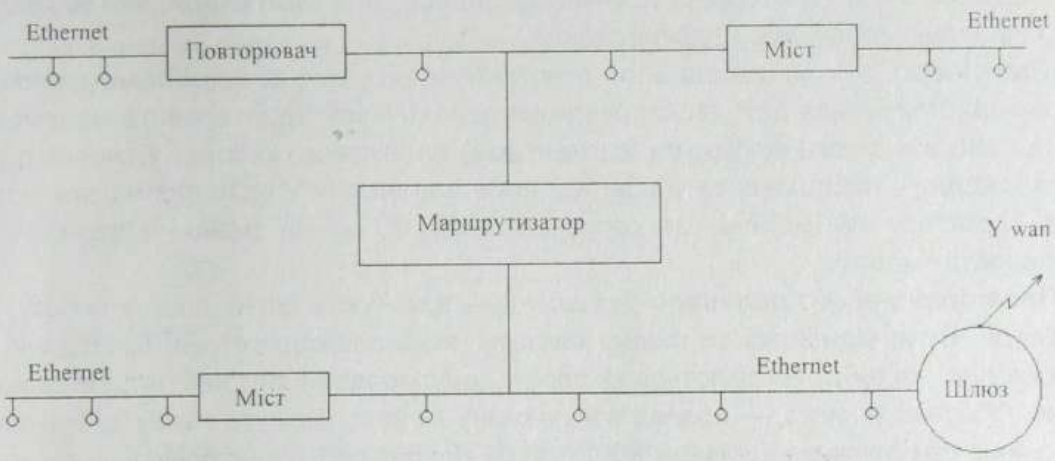


Рис. 1.6. Підключення повторювачів, мостів, маршрутизаторів та шлюзів

Мости (*bridges*) — це порівняно нескладні пристрої, що організовують обмін між мережами з різними стандартами обміну, наприклад Ethernet, Arcnet, TokenRing тощо, а також між декількома сегментами однієї мережі з метою розподілення навантаження між ними. Міст — це спецвузол мережі, який аналізує всі пакети даних, що проходять скрізь нього у обох напрямках та забезпечує при необхідності безпеку даних, надійність мережі тощо.

Маршрутизатори (*routers*) — це більш складні пристрої, ніж мости. Вони використовуються для того, щоб вибрати оптимальний шлях (маршрут) для кожного пакета. Це робиться для сильно розгалужених мереж з метою обходу пошкоджених ліній або для зменшення навантаження на окремі сегменти мережі. Використовуються для зв'язку однорідних сегментів (мереж з однією технологією).

Гібридні маршрутизатори (*brouters*) — об'єднують функції моста і звичайного маршрутизатора. Маршрутизатор має "свою" мережеву адресу (на відміну від моста), керує потоками даних і різними службами у мережі, підвищує безпеку мереж, сегментує дані тощо.

Перемикач (*switch*) — це пристрій, який використовується для перемикання (перенаправлення) пакетів даних у мережах. В якості перемикача можуть бути і мости і маршрутизатори, і мережеві комп'ютери (*host-computer*). Для перемикача існує безліч назв. Наведемо деякі з них:

- IMP (Interface Message Processor) — інтерфейсний процесор повідомлень;
- CC (Communication Computer) — комп'ютер для зв'язку;
- PS (Pocket Switch) — пакетний перемикач;
- DSE (Data Switching Exchange) — вузол обміну та перемикання даних;
- Node — вузол.

Комутатор — це пристрій, що автоматично розпізнає програмні пакети та перетворює дані або у логічні групи (для Virtual LAN) для одного середовища передачі

даних, або в одну різнорідну групу для різних середовищ передачі даних; здійснює фрагментацію даних.

Вузол (*node*) мережі (або абонент мережі) — комп'ютери або інші пристрої мережі: персональні комп'ютери, спецмікроконтролери, міні-комп'ютери, великі комп'ютери, спеціальні робочі або графічні станції.

Трансівери (*transceiver* від англ. *transmitter* + *receiver*) або ***прийомопередавачі*** — використовуються для двонаправленої передачі між адаптером та мережевим кабелем або між двома відрізками (сегментами) мережевого кабелю. Основна функція трансівера — посилення сигналів або перетворення їх у іншу форму для поліпшення характеристик мережі. Це пасивний пристрій, що не змінює інформаційних характеристик мережі.

Повторювачі або ***репітери*** (*repeaters*) — виконують більш просту роботу, ніж трансівери. Вони відновлюють форму сигналу, який пошкоджується, проходячи через довгі лінії, та використовуються як прості двонаправлені ретранслятори сигналів мережі. Основна їх мета — подовжити довжину мережі. Можуть також здійснювати гальнонічне розгалуження сегментів мережі та відновлювати синхронізацію сигналів. Повторювач — пасивний пристрій мережі.

Концентратори або ***хаби*** (від англ. *hub*) — це спеціальні пристрої, до яких підключаються робочі станції (PC) мережі. Використовуються для підключення декількох абонентів мережі. З точки зору обробки інформації, вони бувають пасивні та активні (рис. 1.7).

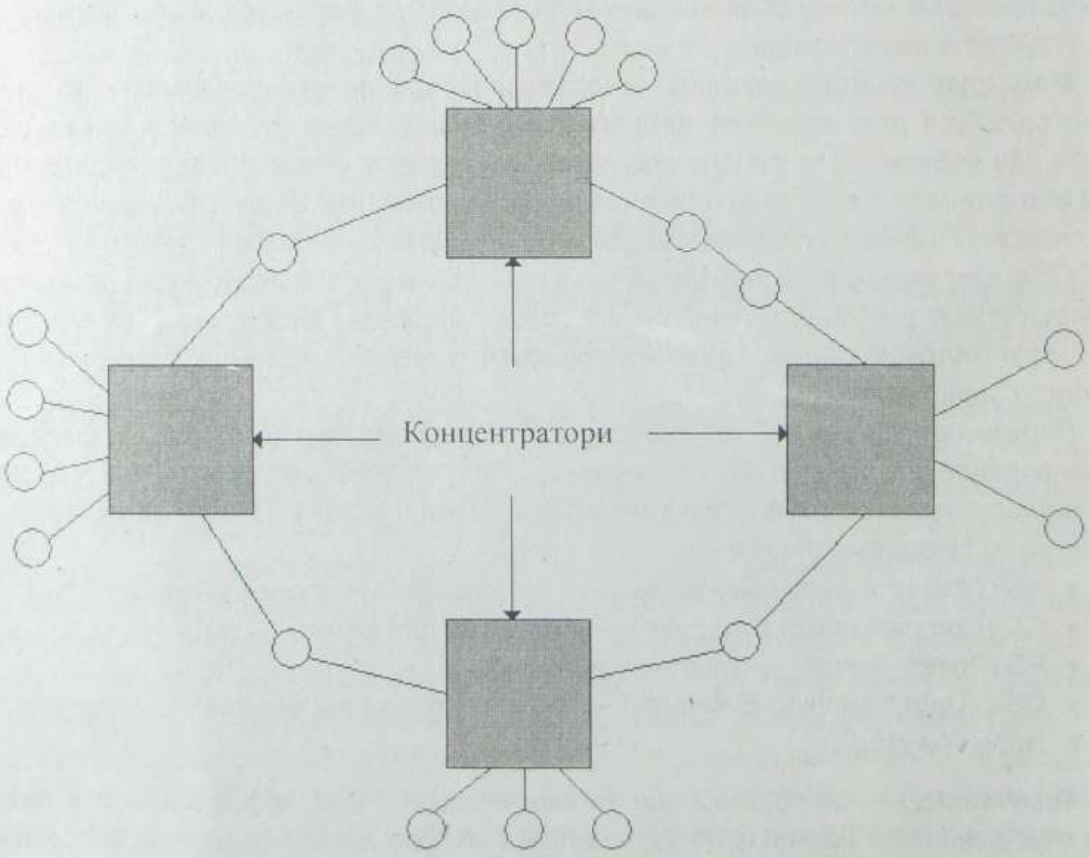


Рис. 1.7. Підключення концентраторів у мережу типа "кільце"

Головною функцією концентратора є ретрансляція повідомлень будь-якої передаючої PC до усіх сегментів, що підключені до мережі. Сегменти з різною топологією ("зірка", "шина", "дерево", "кільце" тощо) можуть за допомогою хаба об'єднуватись в одну LAN.

Підключення репітера та пасивного концентратора наведено на рис. 1.8.

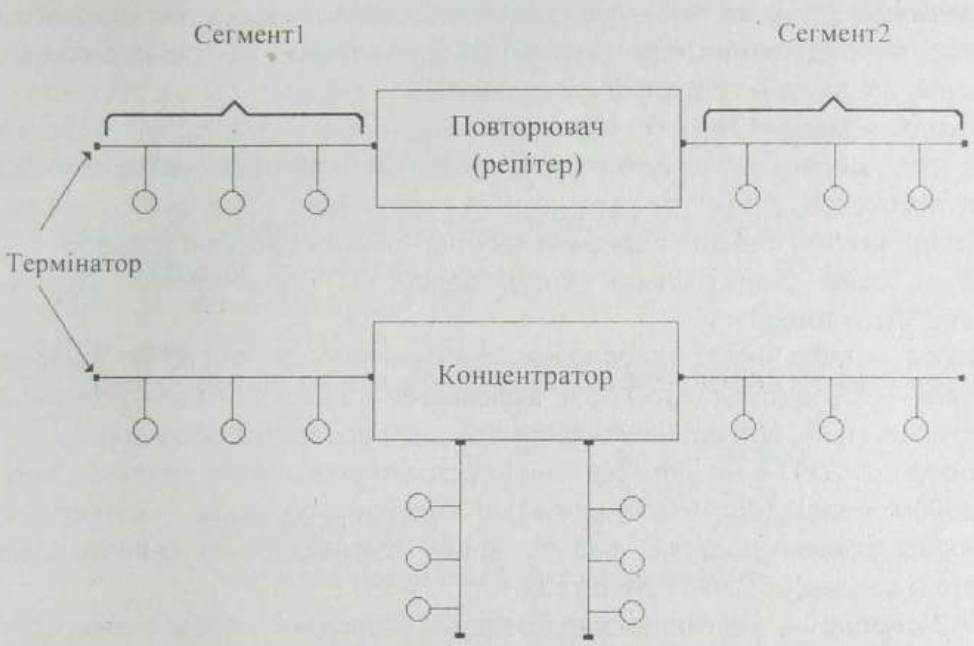


Рис. 1.8. Підключення репітера та пасивного концентратора

Мережевий адаптер (інтерфейсна карта, мережевий процесор, контролер) — це програмно-технічний пристрій (засіб), який перетворює дані у цифрову форму і забезпечує їх безпомилкову передачу по мережі; використовується для з'єднання комп'ютера і середовища передачі даних з урахуванням прийнятих для даної мережі правил обміну інформацією.

Мережевий комп'ютер (*Host-computer*) — використовується в якості перемикача або керуючого програмно-технічного пристрою.

Конектор (*connector*) або ***Т-конектор*** — це звичайний трійник, що дозволяє підключати до комп'ютера два додаткових відрізка кабелю, які мають спецроз'єми. Конектор підключається до мережевого адаптера і використовується у мережах з топологією "загальна шина" для зв'язку сегментів мереж (відрізків кабелю), які не можуть бути довшими 185 м (технічне обмеження).

Конвертор (*converter*) — це пристрій, що використовується для з'єднання мереж з різними типами кабелів, але з однаковою швидкістю передачі даних.

Термінатор (*terminator*) — це заглушка на кінці кабелю (опору 50 Ом) у мережах топології "загальна шина", тобто, якщо комп'ютер крайній у мережі, то у вільний кінець Т-конектора повинен бути встановлений BNC-термінатор (BalaNCed terminator), а інший кінець конектора повинен бути заземлений.

Брандмауер (*firewall*) — це система захисту інформації у мережі, котра частіше за все обмежує доступ у корпоративні мережі (Intranet) з Internet.

Шинний арбітраж — це програмно-технічний комплекс, що встановлює правила за якими комп'ютер у мережі впізнає, що шина вільна і можна передавати дані.

Протокол (*protocol*) — це сукупність правил та форматів (стандарти), що визначають взаємодію об'єктів у мережі; це загально прийнятий набір команд, що вказує, як оперувати з даними у мережі і як двом мережевим комп'ютерам (клієнту та серверу) зв'язатися між собою та "розуміти" один одного.

Трафік (*traffic*) — потік даних, що передаються по мережі.

Транзакція (*transaction*) — дискретна операція в межах комп'ютерної системи, (наприклад, впорядкування користувачів або поновлення інвентарного опису); послідовність дій, що використовуються у сукупності.

Топологія мережі (*network topology*) — це геометрична форма мережі або фізичне розташування комп'ютерів у мережі відносно один одного. Топологія мережі визначає просторову структуру і розміщення комп'ютерів, об'єднаних у мережу, а також пристрої, які потрібні для з'єднання кабелю. Існують наступні топології: "загальна шина" (*Bus*), "зірка" (*Star*), "кільце" (*Ring*), "дерево" (*Tree*), коміркова, чарункова або мережева (*Mesh*) тощо.

Адреса — набір чисел, що однозначно визначають деякий об'єкт у мережі.

Пакет — група (сукупність) бітів, включаючи адресу, дані і контрольні суми, що передаються як ціле, або одиниця інформації, що передається по мережі.

Сервер (*server*) — це універсальний програмно-технічний пристрій, який надає користувачам мережі багатофункціональні послуги, наприклад, комп'ютер-сервер, файл-сервер, сервер додатків, сервер друку, комунікаційний сервер, Data Ware House (архів-сервер), СПАЗ-сервер тощо.

СПАЗ-сервер — це мережеве програмно-апаратне забезпечення для будь-якого вузла (рівня) мережі у відповідності до архітектури мережі, яке має для кожного рівня специфічний протокол та інтерфейс.

Архітектура мережі (*network architecture*) — це структура комп'ютерної мережі або її частини з урахуванням дисципліни (правил) з'єднання та їх топології. Це також основна структура комп'ютерної мережі, включаючи технічне та програмне забезпечення, функціональні рівні, інтерфейси та протоколи, які використовуються при встановленні зв'язку та забезпеченні надійної передачі даних. Прикладами стандартної мережевої архітектури є запропонована міжнародною організацією по стандартизації семирівнева модель ISO/OSI (Open Systems Interconnection), яка визначає структуру взаємодії відкритих систем, та модель SNA (Systems Network Architecture) компанії IBM.

Метод доступу (*access method*) — це набір правил, що визначають як використовувати мережу різним споживачам (користувачам); він реалізується на фізичному рівні і часто залежить від топології мережі.

Існують стандартні та нестандартні методи доступу. До стандартних відносять наступні методи доступу (тип стандарту прийнято називати ім'ям мережі прототипу):

- Ethernet (компанія Xerox) або ISO 8802-3 — метод випадкового доступу;
- ARCNet (компанія Data Point) або ISO 8802-4 — метод "маркерна шина";
- TokenRing (компанія IBM) або ISO 8802-5 — метод "маркерне кільце";
- CambridgeRing (університет Кембріджу) або ISO 8802-7 — метод "тактоване кільце", SMA/CD (Carrier Sense Multiple Access/Collision Detection) (множинний доступ з прослуховуванням несучої/знаходження колізій).

До нестандартних відносяться деякі методи випадкового та детермінованого доступу Gigabit Ethernet, ISDN (Integrated Service Digital Network) тощо.

Т-конектор — це звичайний трійник, що дозволяє приєднати до комп'ютера два відрізки кабелю, які мають спеціальні роз'єми. Цей трійник дозволяє будувати мережі типу Ethernet топології "загальна шина" чи "кільце". Т-конектор зазвичай входить у комплект поставки мережевої карти. Якщо комп'ютер останній у мережі топології "загальна шина", то у відповідний кінець Т-конектора повинен бути вставлений BNC- термінатор з опором 50 Ом.

1.2.1. Кабель або типи ліній зв'язку

Існують наступні середовища передачі інформації в мережі:

- неекранована вита пара (рис. 1.9);
- екранована вита пара (рис. 1.10);
- коаксіальний кабель (рис. 1.11);
- волоконнооптичний кабель, радіоканал, інфрачервоний канал тощо.

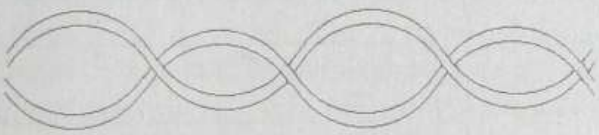


Рис. 1.9. Неекранована вита пара проводів

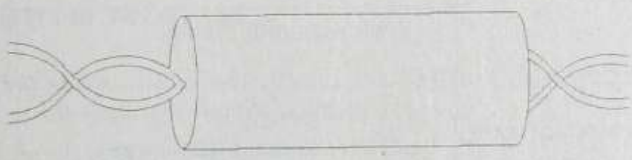


Рис. 1.10. Екранована вита пара

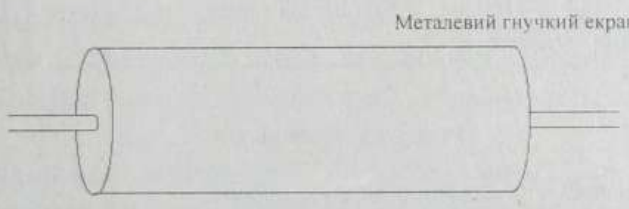


Рис. 1.11. Коаксіальний кабель

З'єднання комп'ютерів мережевим електричним кабелем наведено на рис. 1.12.

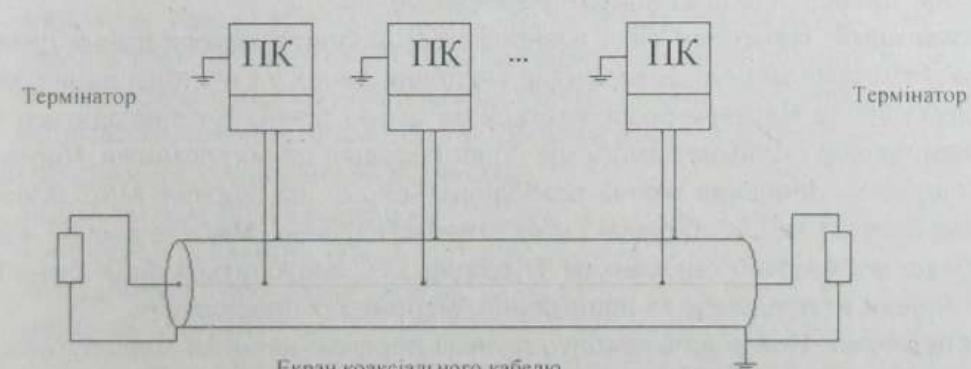


Рис. 1.12. З'єднання комп'ютерів у мережу: опір згоди ("узгодження"); хвильовий опір

Технологія побудови мережі характеризується відповідним методом доступу до каналів передачі даних у мережах і позначається своєю характерною назвою: Ethernet, Fast Ethernet, Gigabit Ethernet, TokenRing, 100VG-AngLan, ARCNet, FDDI тощо.

1.3. Модель OSI

Організація ISO розповсюдила стандарт на модель архітектури обчислювальних мереж під назвою OSI (Open System Interconnect — взаємодія відкритих систем). Більшість виробників намагаються дотримуватись моделі OSI. Модель OSI розподіляє комунікаційні функції в LAN на сім рівнів. Взаємозв'язок рівнів один з одним здійснюється за допомогою добре визначених інтерфейсів (рис. 1.13).

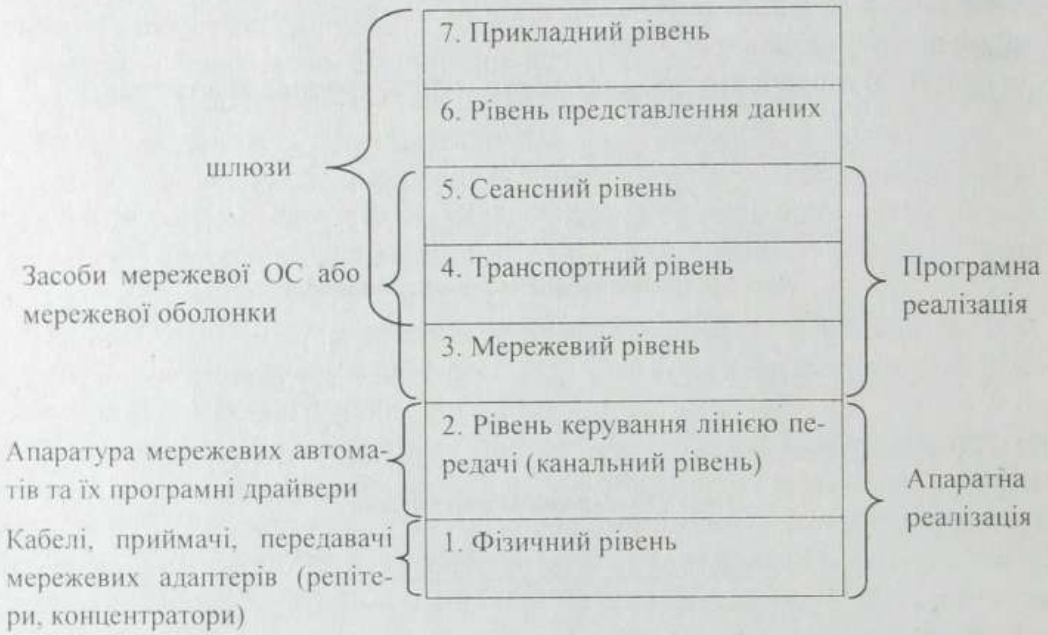


Рис. 1.13. Модель OSI

Розглянемо призначення кожного з семи рівнів:

- **Фізичний**. У цій частині модель ISO визначає фізичні, механічні та електричні характеристики ліній зв'язку, що складають LAN. Можна вважати цей рівень таким, що відповідає за апаратне забезпечення.
- **Канальний**. На цьому рівні визначаються правила використання фізичного рівня вузлами мережі. Електричне уявлення даних у LAN (біти даних, методи кодування та маркери розпізнаються на цьому рівні). Тут знаходяться та виправляються (шляхом вимоги ще одної передачі даних) помилки. Через свою складність канальний рівень розподіляється на два підрівні: MAC (Media Access Control) та LLC (Logical Link Control). Підрівень MAC зв'язаний з доступом до мережі та її керуванням. Підрівень LLC знаходиться вище рівня MAC і зв'язаний з передачею та прийманням мережевих повідомлень.
- **Мережевий**. Цей рівень виконує функції переключення та маршрутизації пакетів. Він відповідає за адресацію та передачу пакетів.

- **Транспортний**. Коли в процесі обробки знаходиться більше, ніж один пакет, транспортний рівень контролює порядок проходження компонентів повідомлення. Якщо приходить дублікат прийнятого раніше повідомлення, то цей рівень розпізнає повторення та ігнорує повідомлення.
- **Сеансний**. Функції цього рівня полягають у координації зв'язку між двома програмами, що працюють на різних робочих станціях. Це відбувається у вигляді добре структурованого діалогу. До цих функцій входять створення сеансу, керування передачею та прийомом пакетів повідомлень під час сеансу та припинення сеансу.
- **Рівень представлення даних**. Перетворює дані з внутрішнього формату комп'ютера в інший формат. Така ситуація може виникнути в LAN з різними ПК (PC, Macintosh, DEC, Next, Burrough, та ін.), яким необхідно обмінюватися даними.
- **Прикладний**. Цей рівень є проміжним між прикладною програмою та процесами моделі OSI.

Повідомлення, призначене для передачі через комп'ютерну мережу потрапляє в модель OSI у даній точці, проходить до рівня 1 (фізичного), пересилається на іншу робочу станцію, проходить від рівня 1 у зворотному порядку до досягнення прикладної програми на іншій робочій станції через її прикладний рівень.

1.4. Компоненти мереж

У загальному випадку базові компоненти мереж поділяються на дві великі групи: апаратні або технічні засоби та програмні.

До апаратних засобів відносяться:

- комп'ютери (Mainframe, Host, міні- та мікро-ЕОМ, персональні та мережеві ЕОМ, Internet ЕОМ, Notebook, Palmtop, Desktop, сервер, робоча станція, графічна станція PDA тощо), шлюзи, повторювачі, мости, маршрутизатори, перемикачі, комутатори, хаби, конектори, термінатори, брандмауери, модеми тощо;
- мережеві адаптери (контролери, мережеві плати) з відповідними методами доступу: Ethernet, ARCNet, TokenRing (стандартними/нестандартними);
- кабельні системи (вита пара, коаксіальний кабель, оптоволокно, тощо);
- некабельні системи або бездротові (радіо-, супутниковий зв'язок);
- мережеві процесори (контролер моноканалу, локальний оперативно запам'ятовуючий пристрій (ЛОЗП), контролер прямого доступу до ЛОЗП, контролер припинень, мікропроцесор, арбітр локальної шини, порт керування адаптером, таймер, генератор тактових імпульсів тощо).

До програмних компонент відносяться:

- мережеві операційні системи;
- комунікаційні пакети;
- спеціалізовані програмні засоби;
- мережеві утиліти і т. ін.

Приклад взаємозв'язку компонентів мережі наведено на рис. 1.14.

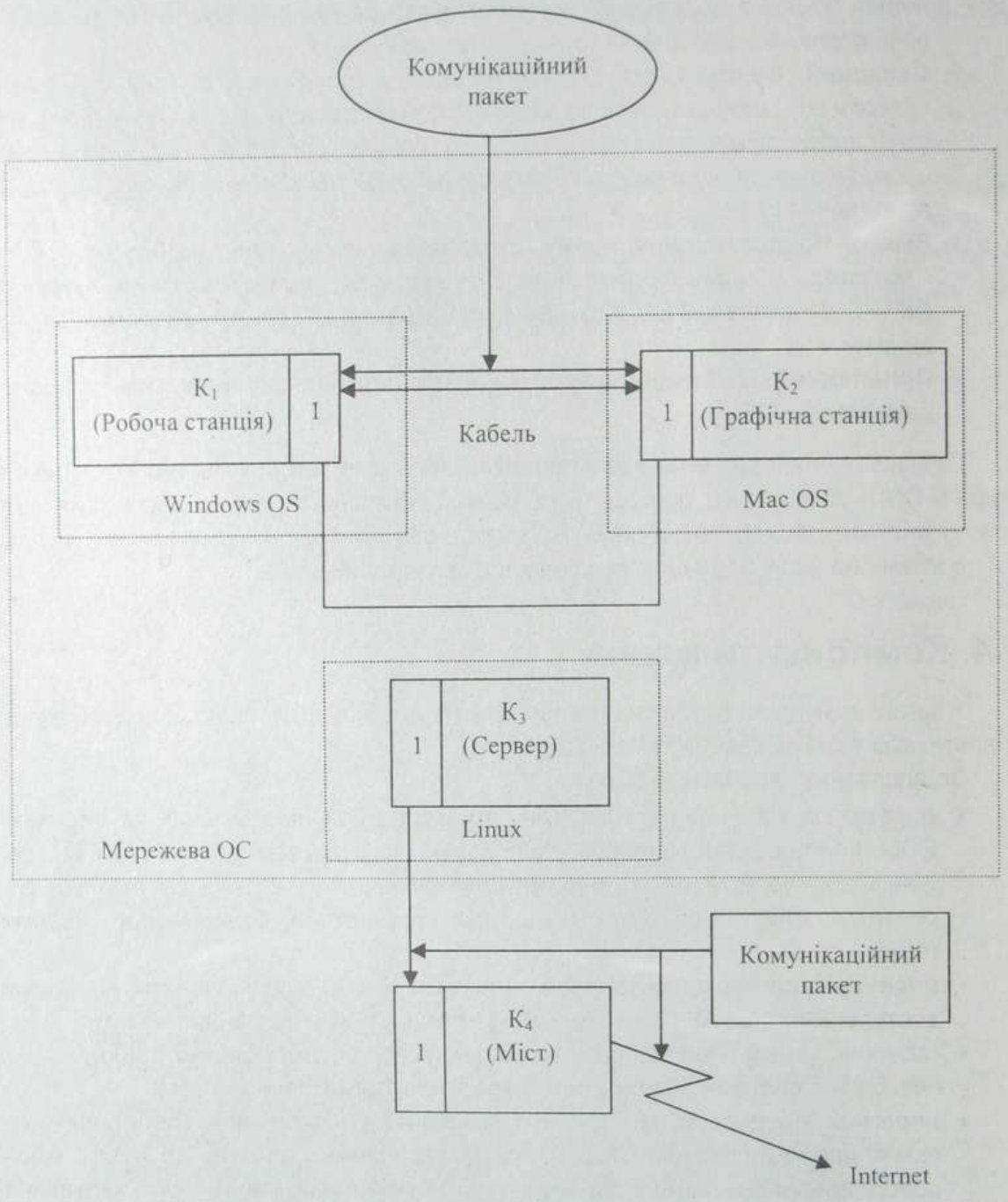


Рис. 1.14. Взаємозв'язок технічно–програмних компонентів мережі (приклад LAN): К — комп'ютер; 1 — мережева карта

Розглянемо деякі апаратні та програмні компоненти детальніше.

1.4.1. Апаратні компоненти мережі

Робочі станції

Робочі станції (*work station*) — це спеціалізовані комп'ютери, що входять до складу мереж, на яких працюють користувачі, які не виконують постійно функції розподілення своїх ресурсів. У загальному значенні — це комплекс обчислювальної апаратури та апаратури введення-виведення, який може використовувати окрема особа. Часто термін "робоча станція" вживають стосовно потужного віддаленого комп'ютера з найсучаснішими обчислювальними та графічними характеристиками.

Конкретні моделі робочих станцій можуть дуже відрізнятись одна від одної залежно від класу задач, що вирішує користувач. Для потужної робочої станції є характерними значні обсяги оперативної пам'яті, високопродуктивний процесор або декілька процесорів, значний обсяг дискової пам'яті.

Вимоги до робочих станцій:

- надійність, стабільність роботи (ці характеристики часто важливіші за швидкість);
- наявність специфічного (унікального) програмного забезпечення;
- реалізація робочих (конкретних) задач типу "mission critical", тобто задач, для яких некритична швидкодія їх виконання, але критична надійність апаратного забезпечення та час неперервної роботи;
- використання ОС типу UNIX, Linux, Windows, FreeUnix тощо;
- вирішення задач конкретного користувача (фахового споживача), тобто робоча станція — це інструментарій вирішення задач конкретної галузі науки та/або практики. У даному випадку спеціалісту не потрібна універсальність або багатофункціональність;
- багатопроцесорність.

Графічні станції

Графічні станції характеризуються спеціалізованим технічно-програмним забезпеченням та реалізацією наступних функцій:

- реалізація та виготовлення реклами, презентацій, анімаційних фільмів;
- обробка великих відеопотоків; теле– та відеомонтаж; обробка некомпресованих відеопотоків; робота з великими двомірними та тримірними зображеннями у реальному часі; оперування графічними файлами великого розміру (наприклад, 5000x5000 пікселів) у реальному часі тощо.

Файловий сервер

Файловий сервер — це окремий комп'ютер, який обслуговує всі комп'ютери, що працюють у конкретній мережі. Він забезпечує спільне використання файлів, що знаходяться на його накопичувачах.

Файлові сервери — це високопродуктивні комп'ютери, характерною особливістю яких є концентрація технічних вирішень для забезпечення швидкого пошуку файлів: перш за все — висока тактова частота процесорів, значний обсяг оперативної пам'яті, швидкодіючі накопичувачі. Разом з тим, файлові сервери часто оснащуються

монохромним монітором та недорогою клавіатурою, тому що вони не використовуються інтерактивно користувачами.

Сервери повинні бути високоякісними та високонадійними машинами, тому що при обслуговуванні всієї комп'ютерної мережі вони багаторазово виконують роботу звичайної робочої станції. Їхні накопичувачі повинні бути високонадійними та мати значний термін роботи.

У так званих одноранговых мережах файл-сервер використовується ще й як робоча станція. Ясно, що вимкнення такої робочої станції-сервера заблокує роботу всієї мережі. Обчислювальні ресурси такої робочої станції, яку в цьому випадку називають невиділеним сервером, також доведеться розділяти на функції забезпечення роботи файл-сервера та робочої станції.

Файловий сервер може використовувати іншу операційну систему, ніж робочі станції. Прикладом мережевої операційної системи, яка працює тільки на файловому сервері, є ОС NetWare компанії Novell.

Мережеві кабелі

Мережеві кабелі мають багато модифікацій та типів залежно від пропускної спроможності мережі. Використовуються як тонкі коаксіальні кабелі (їх називають Thinnet або CheaperNet), так і товсті коаксіальні кабелі (Thicknet), екрановані виті пари (STP — Shielded Twisted Pair), неекрановані виті пари (UTP — Unshielded Twisted Pair), оптоволоконні кабелі. Тип кабелю та тип мережевого адаптера відповідають один одному, їх вибір залежить від вимог, які повинна задовольняти мережа для забезпечення нормального функціонування інформаційної системи.

За допомогою кабелю кожна робоча станція з'єднується з іншими робочими станціями та з файловим сервером. В одних випадках єдиний кабель проходить від вузла до вузла, послідовно з'єднуючи всі робочі станції та всі файлові сервери. Така конфігурація називається "спільна шина" (топологія "шина", топологія "дейзі-ланцюг").

В інших випадках до кожної робочої станції підходить власний кабель з центрального вузла, наприклад файлового сервера — таку топологію ще називають "зіркою". Іноді кабелі періодично розгалужуються у вузлових точках, створюючи зіркоподібне дерево.

Конфігурація "дейзі-ланцюга" вимагає найменшої кількості кабелю, але важче піддається діагностиці та ремонту при порушеннях нормальної роботи. Якщо необхідно прокладати кабелі через стіни та стелю, то може так статися, що проведення мережі виявиться найдорожчою частиною витрат. У кожному місці розгалуження мережі необхідно використовувати спеціальні з'єднувачі для розгалужень кабелів. Іноді необхідно також використовувати додаткове обладнання, наприклад: повторювачі, розгалужувачі, пристрої доступу та інше специфічне обладнання.

Останнім часом для передачі мережевих сигналів між комп'ютерами використовуються інфрачервоні промені або радіохвилі.

Мережеві адаптери

Мережеві адаптери — це спеціальні пристрої, які дозволяють пересилати машинні дані в мережі через мережеві кабелі. Карти мережевих адаптерів установлю-

ють в кожній робочій станції та в кожному файл-сервері. Робоча станція відсилає запит через мережевий адаптер до файлового серверу та отримує відповідь через мережевий адаптер, коли файловий сервер готовий передати чергову частину файлу. Такі запити та відповіді схожі на читання та запис файлів на диск у персональному комп'ютері.

Одночасно тільки два мережевих адаптери можуть обмінюватись інформацією один з одним. Це означає, що інші робочі станції повинні чекати своєї черги. Такі затримки звичайно не дуже помітні, тому при роботі мережі складається враження про одночасний доступ усіх користувачів до файл-сервера.

На картах адаптерів LANtastic існує два роз'єми, один з яких призначений для вхідного кабелю, а інший — для вихідного. На картах адаптерів Ethernet можуть бути або Т-подібний роз'єм, або 15-контактний роз'єм, який за формою нагадує букву D, або роз'єм типу телефонної розетки, а іноді — комбінація всіх трьох перерахованих роз'ємів. На картах адаптерів Token Ring існує 9-контактний роз'єм, а іноді — ще й додатковий телефонний роз'єм.

Карти з двома або більше типами роз'ємів надають можливість вибору з широкого асортименту мережевих кабелів. Наприклад, карта адаптера Token Ring з двома роз'ємами дозволяє використовувати мережевий кабель типу екранованої витої пари (STP) або неекранованої витої пари (UTP), або телефонний дріт.

Карта адаптера приймає всі повідомлення, які передаються по кабелю та вибирає з них тільки ті, що безпосередньо призначені цій робочій станції. Отримане повідомлення затримується доти, доки робоча станція не буде готова приймати повідомлення. У випадку, коли робоча станція збирається послати повідомлення на файловий сервер, адаптер зачекає появи перерви в потоці даних мережі та надасть своє повідомлення у створене вікно. Адаптер також автоматично здійснює перевірку цілісності передачі повідомлення, і якщо будуть знайдені помилки — ініціалізує повторення передачі.

Швидкодія мережі визначається величиною мегабіт за секунду. Враховуючи, що один байт інформації складається з восьми біт, то для того щоб визначити, скільки символів (байтів) за секунду теоретично спроможна пропустити мережа, необхідно величину швидкодії мережі розділити на вісім.

Припустімо, що необхідно передати інформацію з повністю заповненої дискети формату 3,5" ємністю 1,44 Мбайт через мережу. Максимальна швидкодія мережі дорівнює 4 Мбіт за секунду. Поділивши 4 Мбіт за секунду на 8, ми визначимо, що мережа спроможна передати 500 Кбайт даних за 1 секунду. Така швидкодія приблизно дорівнює швидкості передачі даних середнього накопичувача на жорстких магнітних дисках. Таким чином, передача даних з дискети могла б зайняти лише декілька секунд.

На практиці максимальна швидкодія мережі ніколи не реалізується. Фактично мережа не може працювати швидше, ніж найповільніша з її компонент. Якщо забезпечується передача 1,44 Мбайт даних від накопичувача до файл-сервера, то витрачений час буде включати не тільки час на передачу даних, але й час на читання цих даних з диска робочої станції, час на опрацювання даних у робочій станції, а також час на запис даних на диск сервера. Швидкість передачі даних з дискового накопичувача станції найімовірніше буде найменш повільною компонентою, з якою дані передаються на сервер. Запити інших користувачів мережі, які працюють разом у ме-

режі, будуть змішуватись, тому сумарний час передачі буде ще більший. Таким чином робоча станція не може передавати дані мережею швидше, ніж вона читає їх з дискет.

Оскільки вартість мережевих адаптерів залежить від їх швидкодії та типу, при створенні мережі необхідно враховувати обсяги використання мережевих ресурсів. Наприклад, у невеликій мережі, яка має лише чотири робочих станції та один сервер, швидкість передачі даних не має великого значення, тому для такої мережі слід вибрати дешеві 16-бітні мережеві карти. Однак, якщо до одного кабелю підключено сто робочих станцій, то швидкодія адаптера стає важливим параметром.

Адаптер LANtastic

Фірма Artisoft випускає адаптери двох видів: Ethernet та LANtastic. Адаптер LANtastic являє собою оригінальну розробку, працює зі швидкістю до 2 Мбіт за секунду, використовує чотирижильний кабель, який по черзі з'єднує робочі станції одна з одною.

Адаптер Ethernet

Адаптери Ethernet належать до найпоширенішого класу адаптерів, які випускаються багатьма фірмами. Вони використовуються в комп'ютерних системах на UNIX-платформах, комп'ютерах різних виробників (Macintosh, IBM-сумісних тощо) та різних операційних системах. Ці адаптери мають три модифікації (ThinNet, UTP та ThickNet), залежно від товщини кабелю. Адаптери модифікації ThickNet можуть працювати на більших відстанях, ніж інші, але вони значно дорожчі.

Швидкість роботи мережі з Ethernet досягає 10 Мбіт за секунду. При відсутності передачі даних у мережі (запитів до сервера та відповідей від нього) мережа з адаптерами Ethernet знаходиться в стані спокою. Після передачі повідомлення на робочу станцію кабель знову буде вільний.

Припустимо, що одна з робочих станцій намагається запитати щось у сервера в той час, як сервер надсилає відповідь іншій станції. Виникає конфлікт. В такій ситуації обидва комп'ютери — сервер та робоча станція — припинять обмін і будуть намагатись повторити передачу.

Для усунення конфліктів мережеві адаптери Ethernet використовують механізм, який називається CSMA/CD (Carrier Sense Multiple Access/Collision Detection — багатоваріантний доступ з виявленням несучої частоти та усуненням конфліктів). У такій ситуації адаптери переходять у режим чекання протягом певного проміжку часу, після чого повторюють спробу передачі. Цей метод дуже ефективний для невеликих трафіків у мережі та дозволяє одному з комп'ютерів випереджати інші.

Збільшення трафіку в мережі викликає нові конфлікти й відповідно збільшує час реакції на запити, що призводить до зменшення швидкодії мережі. Фактично мережа на основі Ethernet починають витрачати більше часу на вирішення конфліктів, ніж на передачу даних. Для вирішення цієї проблеми дві фірми (IBM та Texas Instruments) розробили мережу з новим типом адаптерів Token Ring.

Адаптер ARCNet

Адаптери цього типу почала випускати фірма Data Point Corporation, але зараз ARCNet-сумісні карти випускає багато інших фірм. Ці адаптери мають невисоку

швидкодію, але вони некритичні до невеликих помилок при встановленні зв'язку. Вони відомі своєю високою надійністю, а проблеми, що виникають легко діагностуються. Адаптери ARCNet дешевші, ніж Ethernet та працюють приблизно за таким принципом, як і адаптери Token Ring, але мають значно меншу швидкість, а саме близько 2,5 Мбіт за секунду.

Адаптери Token Ring

Адаптери цього типу поступаються вартістю тільки адаптерам для оптоволоконних мереж. Token Ring може використовувати екрановані та неекрановані виті пари. Витрати на мережу з Token Ring виправдані, коли мережа має напружений трафік з великою кількістю робочих станцій, особливо якщо використовується потужний центральний комп'ютер у великих компаніях з великою мережею.

Швидкість обміну LAN з адаптерами Token Ring може сягати 100 Мбіт за секунду, тоді як звичайна швидкодія лежить в межах 4 або 16 Мбіт за секунду. У мережі Token Ring навіть за відсутності трафіку всі робочі станції передають одна одній спеціальний сигнал, що називається маркером.

Маркер (*token*) — це унікальний структурований об'єкт даних або повідомлення, що безперервно циркулює по мережі, реєструючи її поточний стан. Це повідомлення є ознакою того, що мережа вільна. Якщо робоча станція на момент прийому маркера не має потреби в передачі, то вона просто передає маркер наступній по порядку станції. Робоча станція може передати своє повідомлення в мережу тільки в тому випадку, коли вона одержала маркер. Якщо LAN зайнято, а робоча станція хоче переслати повідомлення на сервер чи іншу робочу станцію, то необхідно дочекатись надходження маркера. Тільки після цього станція зможе надіслати своє повідомлення. Це повідомлення буде передаватись через інші робочі станції та сервери доти, доки не повернеться на станцію-відправника, після чого відправник надсилає в мережу маркер про те що мережа знову вільна.

Під час циркуляції повідомлення одна з робочих станцій (або файлових серверів) розпізнає, що повідомлення адресовано їй і починає його обробку. Маркеру для проходження по мережі необхідно дуже мало часу, навіть для мережі на 100 чи 200 станцій. Крім того, є можливість присвоїти пріоритети певним робочим станціям та файловим серверам таким чином, щоб вони мали більш частий доступ до мережі.

У результаті схема з передачею маркера є більш стійкою до високого рівня трафіку, ніж схема з визначенням конфліктів типу Ethernet. Карти Token Ring та ARCNet не сумісні одна з одною, але ARCNet використовує схожу схему передачі маркера для управління доступом до робочих станцій і сервера до мережі. Іноді станції пропускають маркер або втрачають його. Щоб цього не трапилось, вони слідкують одна за іншою та використовують складну процедуру регенерації втраченого маркера. Схема Token Ring дещо складніша, ніж Ethernet, тому карти адаптерів для неї відповідно дорожчі.

Для об'єднання різних мереж у глобальну мережу необхідні, по-перше, наявність зв'язку між ними, а по-друге, — наявність умов для обміну інформацією між ними. Для об'єднання неоднорідних мереж використовуються спеціальні пристрої, які забезпечують вирішення усіх завдань, що виникають при інтеграції різних локальних і глобальних мереж. Такими пристроями є мости, маршрутизатори та шлюзи.

Мости

Мости (*bridges*) оперують даними на високому рівні й мають відповідне призначення. По-перше, вони призначені для об'єднання різних мережевих сегментів, що мають різні мережеві середовища (наприклад для об'єднання сегмента з оптоволоконним кабелем і сегмента з коаксіальним кабелем). Мости також можуть бути використані для зв'язку сегментів, які мають різні протоколи низького рівня (фізичного та канального), тобто можливе використання мостів для зв'язку сегментів мережі, як з однаковими протоколами, такими як два сегменти Ethernet, так і для зв'язку сегментів з різними протоколами, такими як два сегменти з Token Ring та Ethernet відповідно.

Мости часто мають властивість бути прозорими для протоколів високого рівня. При передачі даних між двома сегментами вони можуть використовувати третій сегмент, який навіть не буде розуміти даних, що проходять через нього. Проміжний сегмент існує виключно для маршрутизації даних. Також мости дозволяють пристроям та сегментам, які використовують один протокол (TCP/IP або NetBIOS), обмінюватись даними незалежно від фізичного рівня цих мереж.

Мости аналізують, фільтрують, направляють повідомлення, намагаючись знизити трафік сегментів, до яких вони підключені. Цим пояснюється їхня важлива роль у об'єднанні мереж. При виявленні перевантаженості трафіку фізичного сегмента мережі можливе його розділення на два фізичних сегмента за допомогою моста, який обмежить трафік кожного з них, не завантажуючи сегмент даними, адресованими іншому. Мости часто виявляються повільними пристроями, оскільки їм доводиться витрачати багато часу на виявлення та аналіз адрес повідомлень, а також на перевірку пакетів та направлення їх за заданими адресами. Але вони мають комунікаційні якості, завдяки яким стають дуже корисними в комп'ютерних мережах із змішаними протоколами.

Маршрутизатори

Маршрутизатори (*routers*) не мають такої властивості до аналізу повідомлень, як мости, але вони можуть приймати рішення про вибір оптимального шляху для даних між двома мережевими сегментами.

Мости приймають рішення про адресацію кожного з надісланих пакетів даних: переправляти його через міст, чи ні залежно від адреси призначення. Маршрутизатори вибирають із таблиці маршрутів найкращий для даного пакету повідомлень. У полі зору маршрутизаторів знаходяться лише пакети, адресовані до них попередніми маршрутизаторами, у той час, як мости повинні обробляти всі пакети в сегменті мережі, до якої вони підключені.

Тип топології або протоколу рівня доступу до мережі не має значення для маршрутизаторів, тому що вони працюють на рівень вище, ніж мости (мережевий рівень моделі OSI). Маршрутизатори часто використовуються для зв'язку між сегментами з однаковими протоколами високого рівня. Найпоширенішим транспортним протоколом, який використовують маршрутизатори, є IPX фірми Novell.

Необхідно пам'ятати про те, що для роботи маршрутизаторів необхідний один і той же протокол у всіх сегментах, з якими він зв'язаний. При об'єднанні мереж з різ-

ними протоколами краще використовувати мости. Для управління завантаженістю трафіку сегмента мережі також можна використовувати мости.

Гібридні маршрутизатори

Гібридний маршрутизатор (*brouter*) — це гібрид моста та звичайного маршрутизатора. Гібридні маршрутизатори, які часто не зовсім вірно називають багатопротокольними маршрутизаторами, демонструють великі переваги маршрутизаторів та мостів у дуже складних комп'ютерних мережах. Справжні багатопротокольні маршрутизатори не володіють "мостовими" перевагами цих приладів, а просто працюють як звичайні маршрутизатори, не більше ніж з одним протоколом. Гібридні маршрутизатори можуть приймати рішення про те, чи можливо відтворити маршрут для пакета з даним протоколом. Після цього вони виконують маршрутизацію тих повідомлень, для яких це можливо, а для інших служать мостом.

Гібридні маршрутизатори — це складні та дорогі пристрої, які важко встановлювати, але для складних неоднорідних мереж вони є найкращим рішенням.

Шлюзи

Шлюзи (*gateway*) оперують на верхніх рівнях моделі (OSI) (сеансному, представлення та прикладному). Це найбільш розвинутий метод підключення мережевих сегментів та комп'ютерних мереж до центральних ЕОМ. Потреба у мережевих шлюзах виникає при необхідності об'єднання двох систем, які мають зовсім різну архітектуру. Наприклад, шлюз доводиться використовувати для з'єднання LAN з протоколом TCP/IP та великої ЕОМ зі стандартом SNA. Ці дві архітектури не мають нічого спільного і тому необхідно повністю переводити весь потік даних, що проходять між двома системами.

1.4.2. Програмні компоненти мережі

При використанні мереж необхідно встановити спеціальне програмне забезпечення, яке, по-перше, забезпечить розподілення дискового простору, файлів, засобів друку, периферійних пристроїв, а по-друге, надасть можливості використання мережевих ресурсів.

Усі операційні системи можна розділити на мережеві і немережеві. Найбільш поширені мережеві ОС, які інтегрують в собі мережеві можливості та надають змогу безпосередньо працювати в локальних і глобальних мережах:

- **Windows 2000/2003** — операційна система для організації одноранговых корпоративних мереж. Дуже проста з точки зору встановлення, налагодження та супроводження, має зручний інтерфейс користувача. Має можливість підключення до виділених серверів на базі Windows NT у межах LAN.
- **Windows XP** — сучасний освітній стандарт, що включає комфортний інтерфейс користувача, удосконалену систему роботи з ресурсами комп'ютера, підтримку специфікації Plug And Play і 32-бітних програм. До цієї операційної системи включені інтегровані засоби доступу до Internet через комутований зв'язок, засоби доступу до платної мережі Microsoft Network, сервісне програ-

мне забезпечення (ПЗ) для роботи з модемом та електронною поштою; має російськомовну та україномовну версію.

- **OS/2 Warp Connect** — розвиток потужних операційних систем OS/2 та OS/2 Warp компанії IBM. Включає в себе можливості підключення до Internet через комутовані лінії, сервісне ПЗ для роботи з електронною поштою, перегляду новин та Web-браузер. Разом з пакетом під назвою Peer to Peer/2 дозволяє легко інтегрувати робочу станцію в локальну мережу Windows 2000. Разом з пакетом IBM TCP/IP for OS/2 дозволяє інтегрувати робочу станцію в мережі робочих UNIX-станцій на основі протоколу TCP/IP.
- **OS/2 Warp Server** — мережева операційна система, що інтегрує в собі OS/2 Warp та окремий продукт OS/2 LAN Server. З точки зору швидкодії та можливостей, виграє порівняно з Windows NT або Novell Netware.
- **Novell NetWare** — операційна система для файл-серверів. Має потужні можливості по організації та адміністрації файлових систем на серверах. На сьогодні Novell NetWare є найпопулярнішою системою на ринку мережевих операційних систем для серверів.
- **Windows NT** — мережева операційна система компанії Microsoft, спланована як альтернатива UNIX. Працює на платформах, що мають у основі процесори Intel, PowerPC, Alpha, MIPS. Дозволяє будувати на своїй основі сервери як для локальних, так і для глобальних мереж. Підтримує багатопроцесорні машини. Може працювати, використовуючи протоколи TCP/IP, SPX/IPX, NetBIOS, AppleTalk та декілька інших.
- **Personal Netware** фірми Novell з інтегрованими можливостями організації однорангових мереж невеликого офісу (2–20 комп'ютерів) і примітивною багатозадачністю. При наявності в мережі сервера, функціонуючого під Novell Netware, він легко інтегрується і працює в мережі як робоча станція.
- **System 7**, **System 8** (Copland) — операційні системи для комп'ютерів Apple Macintosh та PowerMac. Включають інтегровану підтримку мереж AppleTalk, дружній інтерфейс користувача. Комп'ютери що працюють під управлінням System 7 або System 8 дуже легко інтегруються в мережі AppleTalk, прості у використанні, доступні непрофесійному користувачу.
- **UNIX** — загальна назва для великої кількості систем на базі відкритої архітектури таких, як Solaris (SUNsoft), AIX (IBM), SCO Open Desktop та UnixWare (Santa Cruz Operations), FreeBSD та BSD (Berkley Software Distribution), HP–UX (Hewlett Packard), Linux (вільно розповсюджувана) та багатьох інших. Ці ОС об'єднуються в єдину групу через загальну для всіх них архітектуру та принципи побудови, основані на відкритих стандартах. Всі вони мають інтегровані можливості для функціонування в локальних і глобальних мережах, використані в якості операційних систем серверів і робочих станцій. Слід зазначити, що операційна система UNIX з самого початку свого існування створювалася як справді мережева ОС. Мережеві можливості в тілі довільного клону UNIX не виглядають, так би мовити, стороннім тілом, а є невід'ємною складовою частиною системи.

Немережеві ОС для об'єднання в мережі повинні використовувати спеціальне ПЗ, що дає змогу прикладним програмам користувача використовувати мережеві ресурси.

З усієї розмаїтості комунікаційного та сервісного ПЗ, що розповсюджується як на західному, так і на українському ринку, можна окремо згадати такі пакети:

- **Microsoft TCP/IP for Windows** — набір драйверів для побудови мереж Windows на базі протоколу TCP/IP. Включає утиліти TELNET для доступу до віддалених комп'ютерів і FTP для передачі файлів з іншого комп'ютера.
- **Windows Longhorn** — ОС з графічним ядром Avalon; з новою файловою системою WinFS, системою розробки Web-програм на основі технології .NET — Indigo; з розвинутою службою моніторингу, системою забезпечення безпеки — Palladium; з системою пошуку документів Yukon тощо.
- **Longhorn Milestone 6** — ОС з поліпшеним візуальним інтерфейсом на базі файлової системи Windows Future Storage.
- **Lindows OS** — інтегрована ОС на базі ОС Debian Linux та KDE з можливостями безкоштовного сервісу Internet-телефонії (SIPhone) тощо.
- **Windows .NET, X-Windows** — комбіновані ОС.
- **SUNsoft PC NFS 5.0** — пакет для інтеграції DOS і Windows у мережі на основі UNIX-серверів фірми SUN Microsystems.
- **Eudora, Eudora Light** — програмні пакети для роботи з електронною поштою Internet. Пакет Eudora Light розповсюджується за принципом ShareWare. Eudora є комерційним програмним забезпеченням. Сьогодні комерційний варіант вважається найкращою програмою для роботи з електронною поштою Internet. Включає можливості для роботи в мережі CompuServe, підтримує поштові протоколи POP3 та SMTP. Комерційна версія підтримує стандарт захисту Kerberos, кодування бінарних файлів у стандартах UUE, BinHex та MIME, перевірку орфографії. Працює на платформах Apple та Intel.
- **Netscape Navigator** — браузер, що підтримує стандарти мови HTML (Hyper Text Markup Language), а також його розширення. Існує у варіантах для Windows, а також для більшості клонів UNIX. Має дуже зручний інтерфейс користувача. Пакет має модулі для роботи з електронною поштою та перегляду новин.
- **Internet Explorer** — Web-браузер компанії Microsoft, що розповсюджується безкоштовно. Підтримує стандарти мови HTML, а також його розширення за допомогою технології OLE. Працює під керуванням ОС Windows.
- **NewsExpress** — програма перегляду новин Internet. Вважається однією з найкращих програм цього класу. Підтримує стандарти MIME, UUE кодування бінарних повідомлень, використовує стандарти POP3, SMTP, NNTP.
- **Trumpet Winsock** — комплекс програм та утиліт для доступу до Internet через комутовані лінії.

Окрім цього, існує дуже багато пакетів для доступу до BBS (Bulletin Board Systems — вузлів безкоштовної мережі FIDOnet), систем для адміністрування мереж, моніторингових та захисних комплексів та інших мережевих та службових програмних засобів.

1.5. Класифікація комп'ютерних мереж

Комп'ютерні мережі класифікують за наступними ознаками:

1. Географічна площа:
   - локальні — LAN (Local Area Network);
   - регіональні — MAN (Metropolitan Area Network);
   - глобальні — WAN (Wide-Area Network);
   - тощо.
2. Сфера застосування:
   - офісні (Office LAN);
   - промислові (Industry LAN);
   - фірмові (intranet);
   - побутові (Home AN);
   - приватні (Private AN);
   - загальнодоступні (Public AN)
3. Комплекс архітектурних рішень, що виражається у фірмовій назві:
   - Ethernet/Fast Ethernet;
   - Token Ring;
   - ARCNet;
   - 100VG – Any LAN тощо;
4. Топологія:
   - шинна;
   - кільцева;
   - зіркоподібна;
   - деревоподібна (ієрархічна);
   - повнозв'язна.
5. Фізичне середовище передавання:
   - мережа з симетричним кабелем;
   - мережа з коаксіальним кабелем;
   - мережа з волоконно-оптичним кабелем;
   - мережа з інфрачервоним каналом;
   - мережа з мікрохвильовим каналом;
   - мережа з витою парою;
   - бездротові мережі (радіо-, супутникові канали).
6. Метод доступу до фізичного середовища:
   - мережі з опитуванням;
   - мережі з маркерним доступом;
   - мережі з суперництвом (з пріоритетами);
   - мережі з уставлянням регістра;
   - метод множинного доступу з контролем несучої та виявлення колізій (конфліктів).
7. Набір протоколів (або протокольний стек):
   - мережі TCP/IP;
   - мережі SPX/IPX;
   - тощо.

Схематично класифікацію мереж наведено на рис. 1.15.

Рис. 1.15. Класифікація мереж

В загальному випадку всі комп'ютерні мережі можна поділити на два види:

- локальні обчислювальні мережі (LAN);
- глобальні обчислювальні мережі (WAN).

У свою чергу поняття WAN об'єднує регіональні, корпоративні (Intranet), всесвітні мережі. Останнім часом при класифікації мереж виділяють додатково віртуальні, нейронні, інтелектуальні мережі тощо.

Локальна обчислювальна мережа — обчислювальна мережа, що розташована на невеликій території і використовує орієнтовані на цю територію засоби та методи передачі даних. Ряд особливостей дозволяє виділити LAN в окремий клас обчислювальних мереж. До цих особливостей відносяться:

- розміщення LAN на невеликій території — це означає, що максимальна відстань між довільними двома мережевими пристроями не перевищує 5–10 кілометрів;
- прості методи модуляції сигналу, можливість передачі немодульованих сигналів, низький рівень помилок і прості інтерфейсні пристрої внаслідок малих відстаней;
- відсутність обмежень, що присутні у глобальних мережах, відкритих для широкого кола користувачів;
- легкість зміни конфігурації та самого середовища передачі даних;

- мала вартість самої мережі передачі даних порівняно з вартістю підключених до неї пристроїв.

Загальна структура LAN показана на рис. 1.16.

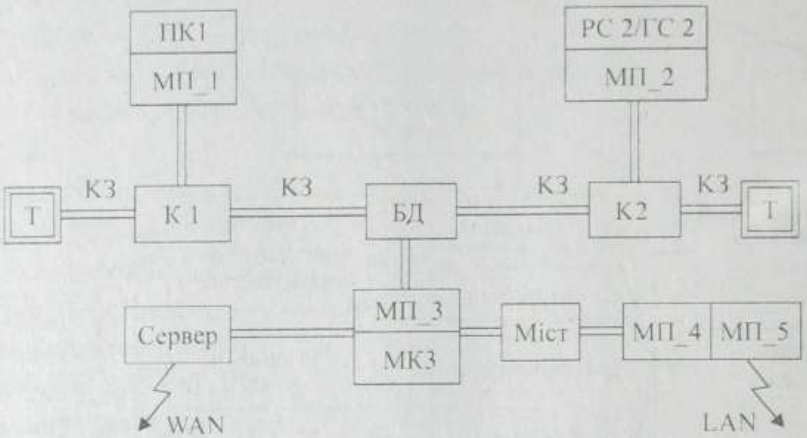


Рис. 1.16 . Локальна мережа для трьох комп'ютерів: ПК — персональний комп'ютер; РС — робоча станція; Т — термінатор; К — конектор; БД — блок доступу; КЗ — канал зв'язку; МП — мережевий процесор (мережевий адаптер); МК — мережевий комп'ютер; ГС — графічна станція

Глобальна обчислювальна мережа — обчислювальна мережа, що розташована на великій території (більше 10 кілометрів) і не включає в себе єдиного для всіх абонентів високошвидкісного каналу передачі даних. Для WAN не є характерним розподілення пристроїв окремих комп'ютерів, її дуже важко модифікувати (прикладом може бути перехід Internet на новий протокол передачі даних), відносно висока вартість встановлення і обслуговування, якщо використовуються високошвидкісні апаратні засоби.

Регіональні та сукупність корпоративних мереж — це глобальні мережі, що класифікуються відносно до свого географічного положення та призначення. ***Регіональні мережі*** — це мережі, що охоплюють деякий географічний регіон (штат, область, країну). ***Корпоративні мережі*** — мережі, створені корпораціями для внутрішнього використання і передачі корпоративної інформації. Оскільки корпорації можуть бути як національними, так і транснаціональними, їхні мережі охоплюють різні за розмірами географічні регіони, але завжди зберігають одну характерну рису, а саме відсутність або жорстку обмеженість доступу для широкого кола користувачів.

Засобами такого обмеження є, наприклад, так звані ***брандмауери*** (*firewalls*) — спеціальні програмні комплекси, встановлені на виділеному для цих цілей комп'ютері (що є бажаним варіантом), або на сервері чи маршрутизаторі, які дозволяють контролювати всі пакети, що виходять із всесвітніх мереж та надходять до корпоративних і навпаки. Брандмауери також дозволяють встановлювати обмеження на шляху передачі окремих груп пакетів, здійснювати моніторинг пакетів, відновлювати початкову та кінцеву адресу пакетів та багато іншого.

Окремим видом WAN є ***всесвітні мережі***. Окремим випадком всесвітньої мережі є мережа Internet, яка фактично виконує об'єднуючі функції. Завдяки тому, що Internet фактично виконує функції об'єднання різних мереж в одну, вона набула дуже широкого розповсюдження. Через відсутність централізованого керування окремі по-

стачальники послуг мають змогу дуже швидко реагувати на досягнення науково-технічного прогресу та швидко адаптуватися до нових ринкових умов.

➢ Глобальна мережа Internet розглядається в частині II книги.

Віртуальна локальна мережа (Virtual LAN) — це локальна мережа з логічною структурою, інваріантною фізичній, але яка побудована на концепції локальної сегментації користувачів (рис. 1.17).

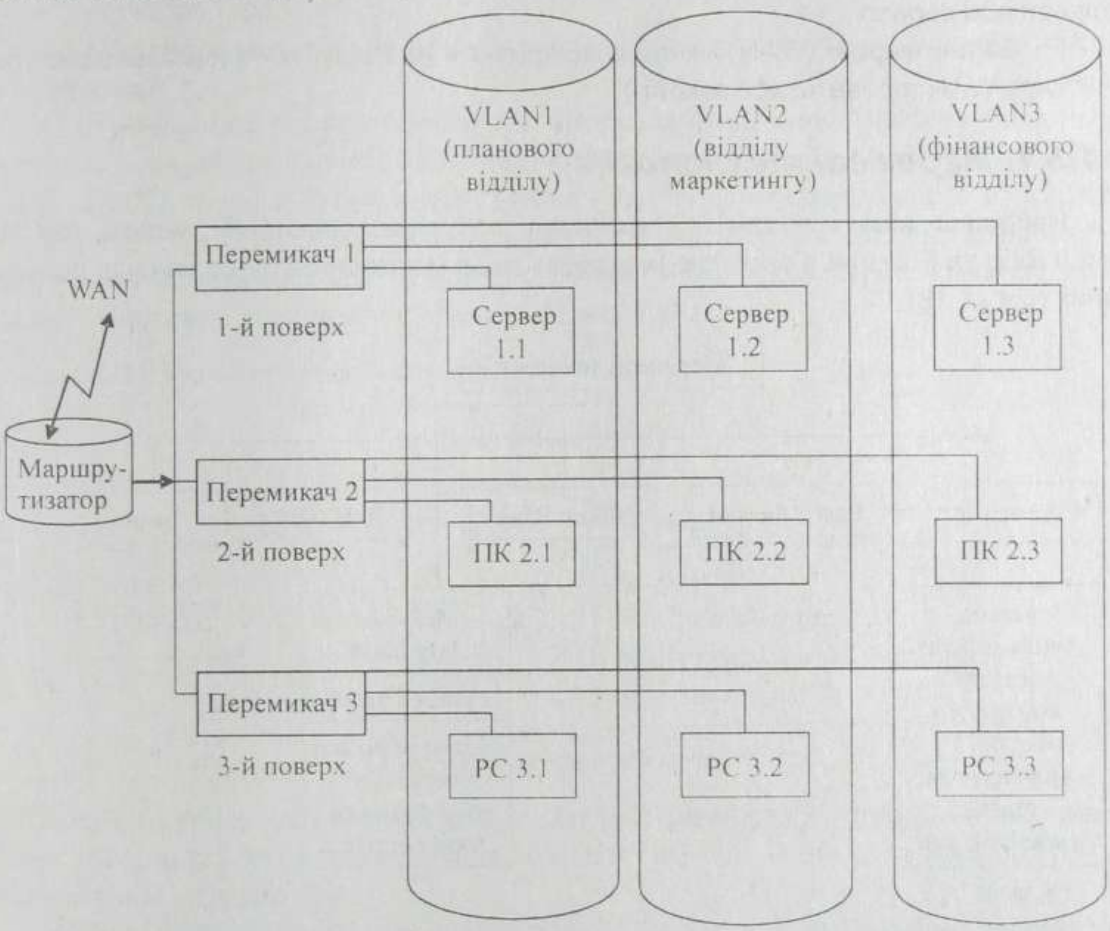


Рис. 1.17. Віртуальна локальна мережа об'єкта: Сервер — файл-сервер; ПК — персональний комп'ютер, PC — робоча станція

Основою VLAN є перемикачі (*switch*), які дозволяють об'єднувати користувачів і створювати всередині об'єкта (організації, установи, фірми) багато незалежних робочих груп.

Нейромережа (*neural network*) — це тип системи штучного інтелекту, яка змодельована подібно до нейронів (нервових клітин) у біологічній нервовій системі і націлена на моделювання процесів обробки, вивчення та запам'ятовування інформації мозком. Нейромережу розробляють як систему обробки взаємопов'язаних елементів з обмеженим числом входів (подібним імпульсним рецепторам нейрона) та одним виходом (подібним сінапсу, через який нервовий імпульс "іде" до наступного нейрона). Нейромережі впроваджуються на апаратно-програмних рівнях.

➢ Нейронні та віртуальні мережі розглядаються в частині III книги.

Гомогенні LAN — це локальні мережі, що побудовані на однорідних (однотипних) комп'ютерах.

Гетерогенні LAN — це локальні мережі, що побудовані на різнорідних (різнотипних) комп'ютерах.

Одноранговa LAN — це локальна мережа, в якій всі комп'ютери рівноправні, і ця мережа не має в своєму складі виділених серверів.

Багаторангова LAN — це локальна мережа, що має в своєму складі виділені сервери або сервер.

Глобальні мережі (WAN) інколи класифікують на Public WAN (загальнодоступні), та Private WAN (приватні або закриті).

1.5.1. Мережеві технології

Найбільш відомі в світі є технології побудови локальних мереж ARCNet, Token Ring та Ethernet. Головною їх відмінністю є метод доступу до каналів передачі даних (рис. 1.18).

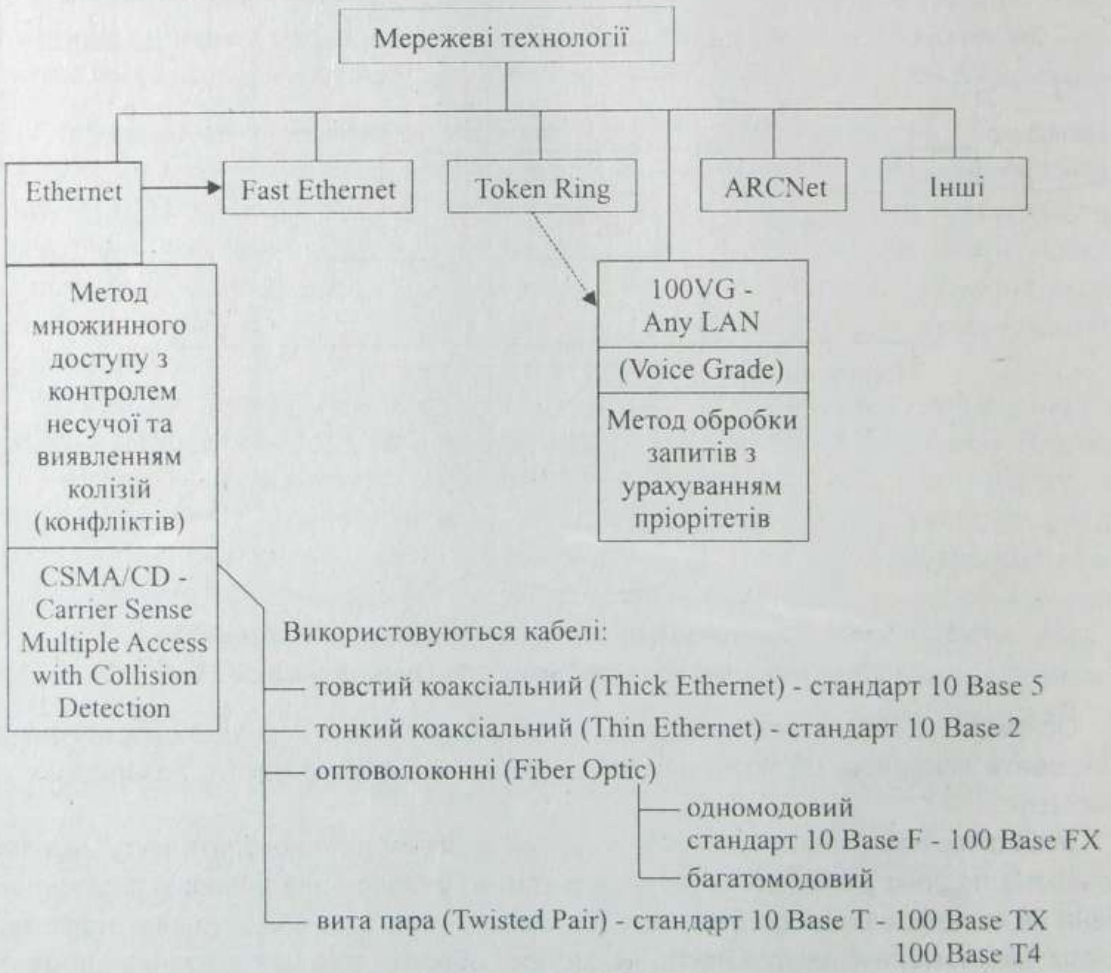


Рис. 1.18. Мережеві технології

1.5.2. Архітектура мереж

Архітектура мережі (*network architecture*) — це основна структура комп'ютерної мережі, яка включає в себе:

- технічне забезпечення;
- функціональні рівні;
- інтерфейси;
- протоколи (правила);

що використовуються при встановленні зв'язку та забезпеченні надійної передачі інформації.

Архітектуру мережі розробляють з метою забезпечення технічно-програмних пристроїв (комп'ютерів) категоріальними й фізичними стандартами, які б регламентували складні процеси встановлення зв'язків та передачі інформації без конфліктів.

Існують різноманітні архітектури мереж, серед яких міжнародно прийнятою є семирівнева модель ISO/OSI, та модель компанії IBM — SNA (System Network Architecture — системна мережева архітектура) (рис. 1.19).

➢ Модель OSI розглядається вище в пункті 1.3.

Рис. 1.19. Архітектура мереж

Кінцевою метою розвитку будь-якої архітектури мережі є створення стандартів зв'язку, які дадуть змогу різнотипним комп'ютерам вільно та чітко (безконфліктно) обмінюватись інформацією.

Модель ISO/OSI (International Organization for Standartization/Open System Interconnection — Міжнародна організація з питань стандартизації та взаємодії відкритих систем) — це багаторівнева архітектура для стандартизації рівнів обслуговування та типів взаємодії між комп'ютерами.

Кожен рівень будується на основі стандарту більш низького рівня. Найнижчий з семи рівнів (фізичний) має справу тільки з апаратними зв'язками. Найвищий (прикладний) — з програмними зв'язками (на рівні прикладних програм).

Рівні ISO/OSI виконують наступні функції:

1. фізичний рівень — апаратні зв'язки.
2. рівень зв'язку між даними або каналами кодування — адресування та передача інформації;
3. мережевий рівень — керування транспортуванням повідомлень, тобто стандартизація зв'язків типу "комп'ютер-комп'ютер";
4. транспортний рівень — коректне постачання та якість обслуговування;
5. сеансний рівень — встановлення та координація зв'язку;

6. рівень представлення або подання — форматування та подання даних;
7. прикладний рівень — обмін інформацією між програмами.

Модель SNA — модель архітектури мережевих систем, або модель зв'язку у мережах (компанія IBM). Модель пропонує свої стандарти обміну інформацією між комп'ютерами різних моделей та її обробку. Модель SNA передбачає п'ять рівнів комунікації, кожний з яких має свою функцію. SNA не передбачає можливості використання апаратного та програмно-прикладного рівнів.

Рівні SNA виконують наступні функції:

1. рівень менеджменту — несе відповідальність за "візуальне" завдання, тобто представлення даних, менеджмент інтерфейсу між мережею і користувачем;
2. рівень керування потоками даних — супровід потоку даних під час сеансу зв'язку;
3. рівень керування передачею — статус та темп сеансу зв'язку;
4. рівень керування маршрутами — включає керування даними;
5. рівень керування зв'язками даних — адресування даних під час передачі.

Порівняння сумісних та несумісних рівнів у моделях SNA та ISO/OSI наведено на рис. 1.20.

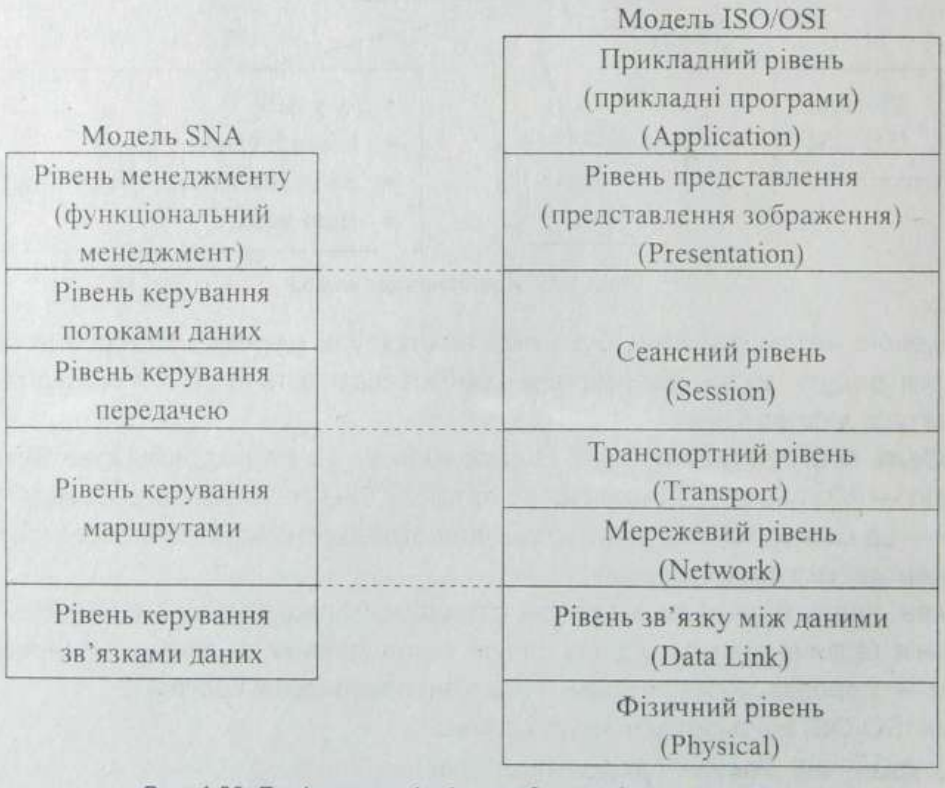


Рис. 1.20. Порівняння рівнів у моделях SNA та ISO/OSI

Протоколи та еталонна модель ISO/OSI наведена на рис. 1.21. Інші моделі архітектури характеризуються реалізацією конкретних систем відомих компаній. Так існують мережеві архітектури за назвою реалізованих мереж: Bay SIS (Bay Networks), CiscoFVsion (Cisco Systems), Switched Virtual Networking (IBM), Synthesis (Cabletron

Systems), Transcend Networking (3 COM), Vivid Switched Virtual Routing Architecture (Newbridge Networks), Policy Management Architecture (Net Manage) тощо.

<table>
<tr><td>7</td><td>Прикладний</td><td rowspan="8">SNA</td><td rowspan="8">DECnet</td><td rowspan="3">NES</td><td rowspan="8">NetWare</td><td rowspan="8">Windows</td></tr>
<tr><td>6</td><td>Представлення</td></tr>
<tr><td>5</td><td>Сеансний</td></tr>
<tr><td>4</td><td>Транспортний</td><td>TCP/IP</td></tr>
<tr><td>3</td><td>Мережевий</td><td rowspan="3" colspan="2">Ethernet
Token Ring
ARCNet
FDDI</td></tr>
<tr><td>2</td><td>Канальний</td></tr>
<tr><td>1</td><td>Фізичний</td></tr>
</table>

Рис. 1.21. Протоколи та еталонна модель ISO/OSI

У загальному випадку на мережеву архітектуру впливає дуже багато факторів. Перелічимо деякі з них:

- апаратні засоби (концентратори, маршрутизатори, комутатори і т. ін.);
- комунікаційні технології (Fast Ethernet, ATM, віртуальні мережі тощо);
- платформи мережевого адміністрування (HP OpenView, IBM NetView, Novel NMS, SunNet Manager, Cisco Works, Spectrum, Transcend, Vivid Sysmem Manager тощо);
- програмні засоби маршрутизації, мережевого моніторингу та аналізу трафіка (Optivity, Internetwork Operating Sysmem, SMON, Empower, NetDirector, Traffic Tuner, RMON, VNA, Vivid Sysmem Software Release і т. ін.);
- тощо.

1.5.3. Топологія мереж

Топологія мережі — це конфігурація локальної мережі, що утворюється з'єднаннями між пристроями. За назвою та функціями, існує багато топологій, наприклад: повнозв'язна, функціонально зв'язна, неповнозв'язна і т. ін. Інколи їх називають магістральна або шинна, "зірка", "кільце", ієрархічна тощо.

Топологія мережі тісно пов'язана з процедурою обміну даними між комп'ютерами (робоча станція, графічна станція, мережевий комп'ютер, Internet-комп'ютер тощо). Правила, котрі встановлюють порядок виконання цих процедур, називають ***протоколами***. Існують спеціальні міжнародні стандарти, що регламентують порядок обміну даних у мережах або методи доступу (Ethernet, ARCNet, TokenRing і т. ін.).

Термін "топологія мережі" відноситься до шляху, по якому дані переміщаються по мережі. Існують три основних види топології: "загальна шина" (рис. 1.22), "зірка" (рис. 1.23) і "кільце" (рис. 1.24).

Локальна мережа може використовувати одну з перелічених топологій. Це залежить від кількості комп'ютерів, їхнього взаємного розташування та інших умов.

Можна також об'єднати декілька локальних мереж, виконаних із використанням різних топологій, у єдину локальну мережу. Може бути, наприклад, деревоподібна топологія, або ієрархічна чи типу "сітка".

Топологія "загальна шина"

Топологія "загальна шина" (рис. 1.22) припускає використання одного кабелю, до якого підключаються всі комп'ютери мережі. У випадку загальної шини кабель використовується спільно всіма станціями по черзі. Приймаються спеціальні міри для того, щоб при роботі з загальним кабелем комп'ютери не заважали один одному передавати і приймати дані.

Вихід із ладу окремих комп'ютерів не порушить працездатності мережі в цілому. Але пошук несправностей у кабелі ускладнений. Крім того, тому що використовується лише один кабель, у випадку його обриву порушується робота всієї мережі.

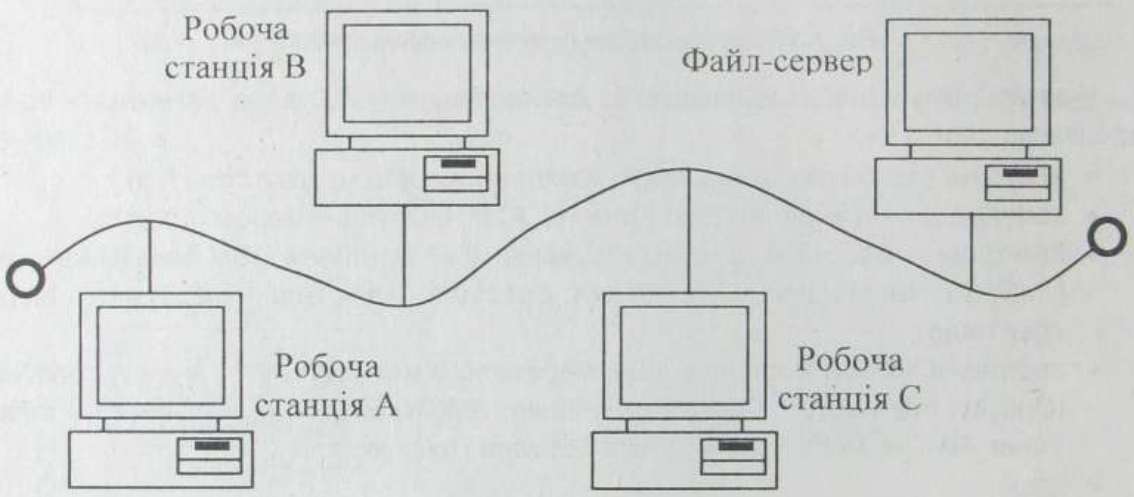


Рис. 1.22. Топологія мережі "загальна шина"

Топологія "зірка"

В разі топології "зірка" (рис. 1.23) кожний комп'ютер через спеціальний мережевий адаптер підключається кабелем до об'єднуючого пристрою. При необхідності можна об'єднувати разом декілька мереж із топологією "зірка", при цьому утворюються розгалужені конфігурації мережі. З погляду надійності ця топологія не є найкращим рішенням, тому що вихід з ладу центрального вузла призведе до припинення роботи всієї мережі. Проте при використанні топології "зірка" легше знайти несправність у кабельній мережі.

Топологія "кільце"

У випадку топології "кільце" (рис. 1.24) дані передаються від одного комп'ютера до іншого як би по естафеті. Якщо комп'ютер одержить дані, призначені для іншого комп'ютера, він передає їх далі по кільцю до тих пір, поки вони не попадуть на "свій" комп'ютер.

Рис. 1.23. Топологія мережі "зірка"

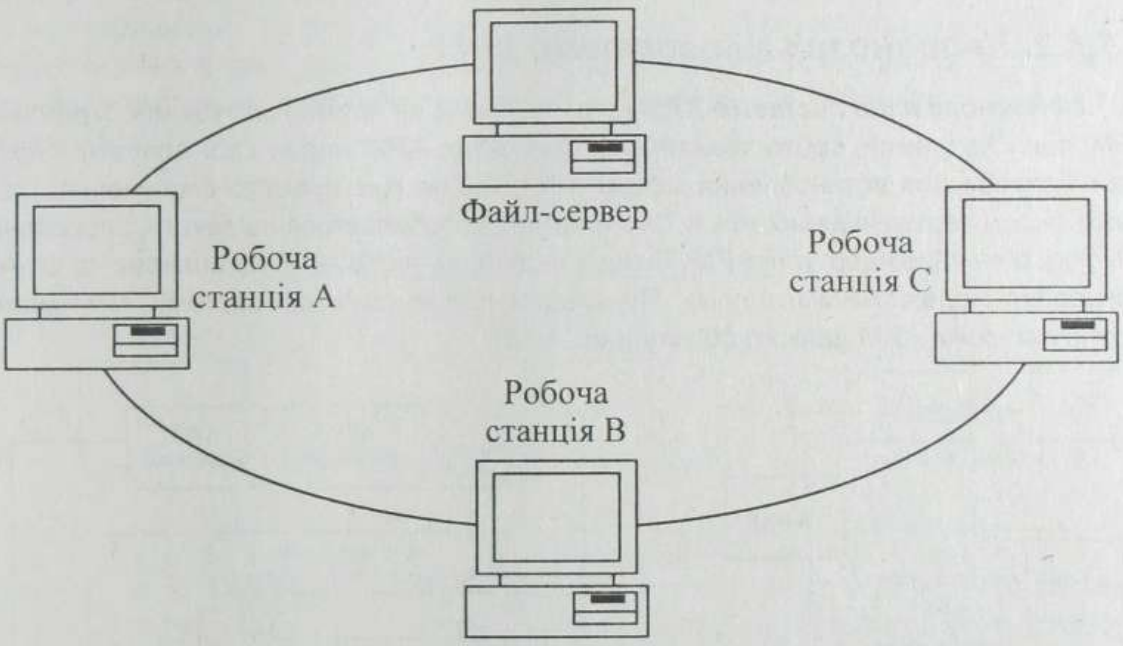


Рис. 1.24. Топологія мережі "кільце"

1.6. Топологія мереж АРМ

Нижче розглянуті варіанти організації мереж АРМ з урахуванням існуючого досвіду проектування мереж ЕОМ.

1.6.1. Повнозв'язна система АРМ

Повнозв'язна система АРМ характеризується зв'язком типу "кожне із кожним" по виділеному каналу (рис. 1.25). Відмова якого-небудь АРМ не впливає на функціонування системи АРМ в цілому при наявності дублювання в реалізації функцій на різних АРМ. Проте це значно ускладнює систему великою кількістю зв'язків. Прикладом повнозв'язної системи АРМ може служити фрагмент мережі АРМ (рис. 1.26).

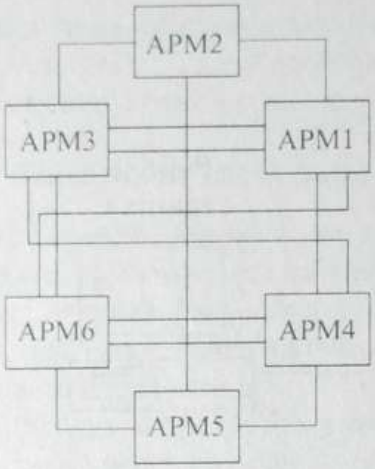


Рис. 1.25. Повнозв'язна система АРМ

Рис. 1.26. Приклад повнозв'язної системи АРМ

1.6.2. Неповнозв'язна система АРМ

Неповнозв'язна система АРМ (рис. 1.27) має як прямий зв'язок між окремими АРМ, так і непрямий, тобто транзитний. Транзитне АРМ надає свої апаратні і програмні засоби для встановлення зв'язку АРМ, що не має прямого сполучення. При необхідності передачі даних між АРМ інформація розбивається на пакети, і передача здійснюється через доступні АРМ. Вихід з ладу будь-якого АРМ не впливає на функціонування мережі АРМ в цілому. Прикладом неповнозв'язної системи АРМ може служити мережа АРМ деякого об'єкту (рис. 1.28).

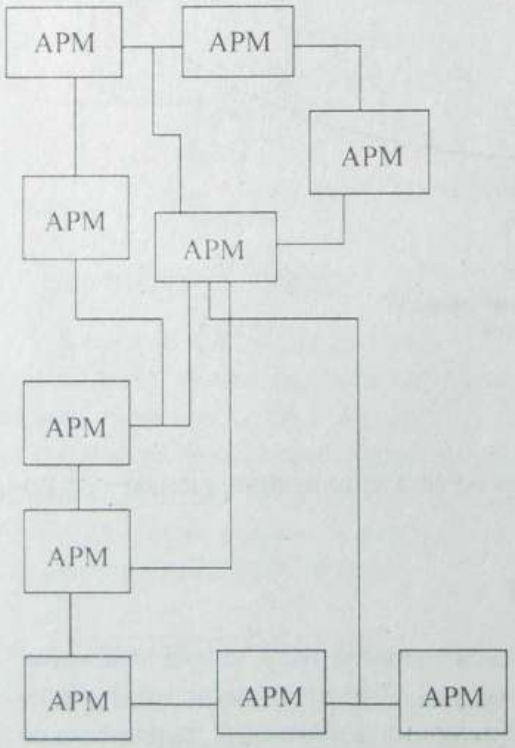


Рис. 1.27. Неповнозв'язна система АРМ

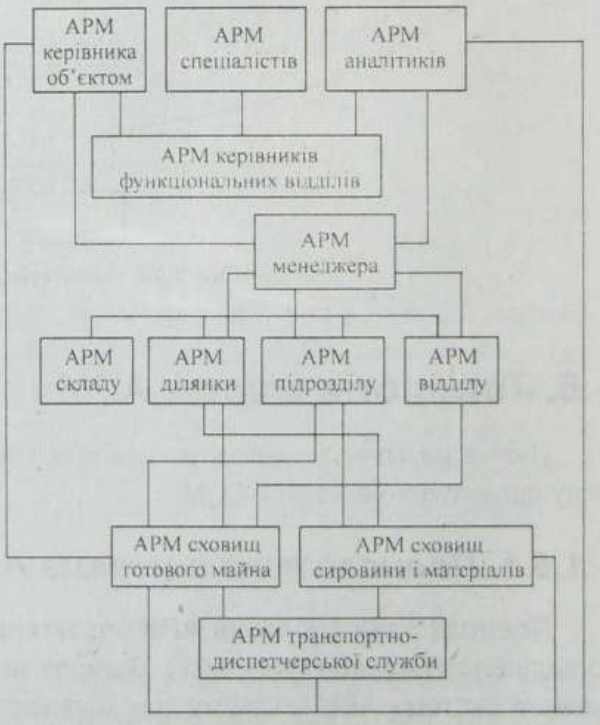


Рис. 1.28. Приклад неповнозв'язної системи АРМ

1.6.3. Регулярна та нерегулярна мережа АРМ

Регулярна мережа АРМ характеризується сполученням АРМ з усіма сусідніми АРМ — граничними АРМ, пов'язаними з поточним АРМ згори і знизу (праворуч і ліворуч). Мережа АРМ стає дуже складною через велику кількість ЕОМ. Вихід з ладу будь-якого АРМ не впливає на функціонування мережі в цілому. При відмові основних ліній можуть використовуватися резервні шляхи передачі між АРМ.

Нерегулярна мережа АРМ не має певної структури сполучення ЕОМ. Як правило, вона застосовується в географічно розподілених системах, де лінії зв'язку визначають топологію зв'язків. При відмові ліній використовується часткове резервування.

1.6.4. Ієрархічна конфігурація мережі АРМ

Ієрархічна конфігурація має деревоподібну структуру зв'язків АРМ (рис. 1.29). По мірі наближення до шпиля дерева збільшуються потужності задіяних ЕОМ. АРМ нижнього рівня в такій системі орієнтовані на конкретне застосування, тобто на рішення цілком певних задач. верхній рівень виконує функції управління і координації мережі АРМ.

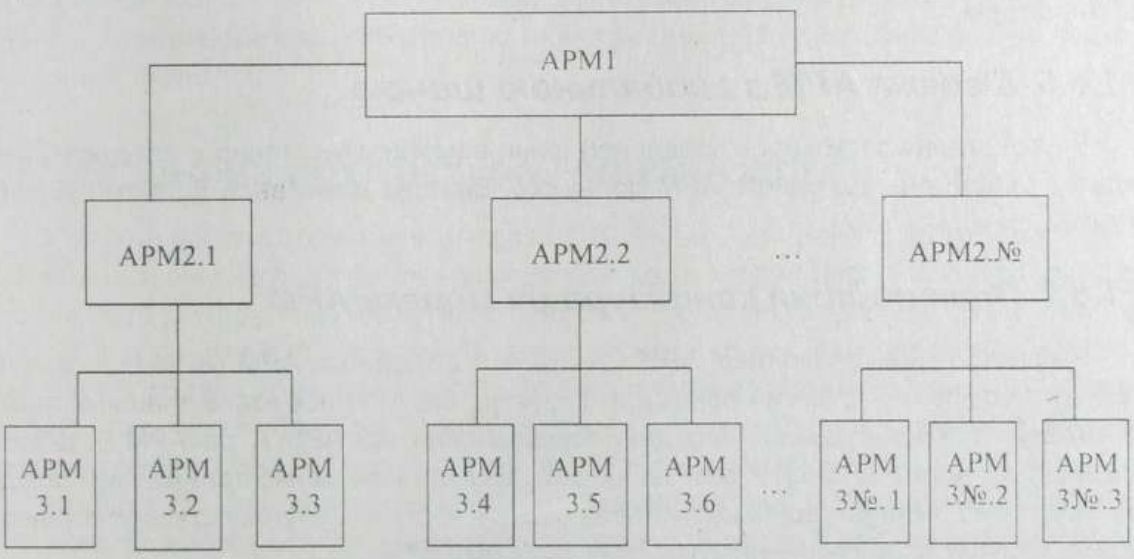


Рис. 1.29. Ієрархічна структура мережі АРМ

На нижньому рівні мережі АРМ звичайно використовують найпростіші комп'ютери з обмеженою кількістю периферійних пристроїв, а верхній рівень повинен бути оснащений ЕОМ з потужнім процесором. В таких системах АРМ нижнього рівня часто — підготовчо-обробляючі; АРМ середнього рівня — професійно-спеціалізовані, АРМ верхнього рівня — індивідуальні АРМ стратегічного характеру.

Ієрархічна конфігурація використовується у системах управління складними процесами і збирання даних. Функції спеціалізовані на нижчих рівнях ієрархії і більш узагальнені на верхніх. При відмовах АРМ нижчих рівнів працездатність системи знижується, а при відмові на більш високих рівнях ієрархії відбуваються чималі порушення.

1.6.5. Петля АРМ

Петля (або кільцевий зв'язок) АРМ наведена на рис. 1.30.

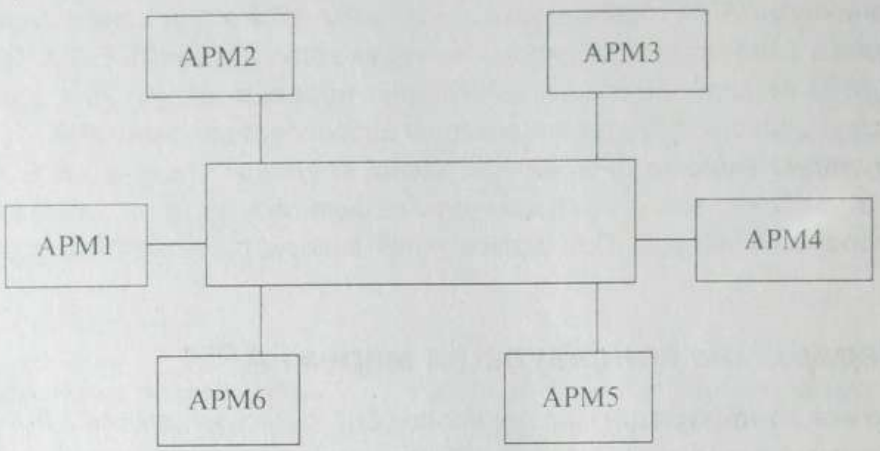


Рис. 1.30. Петля АРМ

Потоки даних можуть проходити у двох напрямках. Система АРМ стійка в разі відмови однієї лінії при подвійній петлі і непрацездатна в разі єдиної однонапрямкової магістралі.

1.6.6. Мережа АРМ з глобальною шиною

Використання загальної (глобальної) шини вимагає спеціального алгоритму передачі повідомлень від одного АРМ до іншого. Відмова шини веде до відмови всієї системи (див. рис. 1.22).

1.6.7. Зіркоподібна конфігурація мережі АРМ

У зіркоподібній конфігурації АРМ (див. рис. 1.23) одне із АРМ (АРМ-Ц — файл-сервер) утворює центр зірки і працює як керуюче. Це АРМ пов'язане з іншими лініями зв'язку. При необхідності передачі повідомлення від АРМ А до АРМ В зв'язок здійснюється через АРМ-Ц. У разі зайнятості АРМ-Ц (передачею-прийманням повідомлення) повідомлення від інших АРМ займають чергу і обслуговуються у встановленому порядку апаратурно-програмними засобами АРМ-Ц.

1.6.8. Система АРМ з розподіленою пам'яттю

Система АРМ з розподіленою пам'яттю наведена на рис. 1.31.

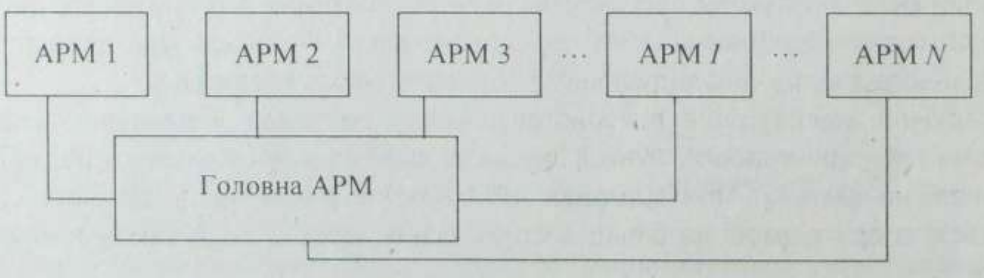


Рис. 1.31. Система АРМ з розподіленою пам'яттю

Найбільш загальний засіб сполучення АРМ — взаємодія одного з іншим шляхом розміщення повідомлень в пам'яті головного АРМ, що доступна для всіх. Головною характеристикою системи є можливість використання пам'яті у вигляді каналу та запам'ятовуючого пристрою.

1.6.9. Гомогенні та гетерогенні мережі АРМ

Мережі АРМ можна також поділити на гомогенні та гетерогенні. ***Гомогенні мережі АРМ*** — це сукупність АРМ, що базуються на однотипних ЕОМ.

Гомогенна мережа АРМ, в якій всі ЕОМ однотипні, значно простіша за архітектурою і засобами зв'язку. Життєвість такої мережі вища, ніж гетерогенної, бо в ній кожний процесор може взяти на себе функції будь-якого іншого, що вийшов з ладу. Така мережа АРМ характеризується однорідністю взаємодіючих елементів. Міра гомогенності впливає на загальну структуру системи АРМ, на вибір ПЗ, на засоби зв'язку АРМ, інколи — на загальні засоби управління функціонуванням мережі АРМ.

Гетерогенна мережа АРМ інколи може опинитися функціонально ефективніше гомогенної, бо вона допускає такий розподіл функції, при якому оптимально використовуються переваги кожного із типів ЕОМ у мережі АРМ. Проте проектування таких систем є більш складним, зокрема з урахуванням динамічного перерозподілу ресурсів різноманітних за архітектурою і конструкцією ЕОМ, що функціонують під керівництвом різних ОС.

1.7. Мережева архітектура "клієнт-сервер"

В мережевій архітектурі "клієнт-сервер" (CSA — Client-Server Architecture) обчислювальні задачі розподіляються між сервером та клієнтом. Задача виконується там, де це найбільш ефективно. Наприклад, на робочій станції необхідно отримати вибірку з таблиці по багатьом умовам. В цьому випадку клієнт відсилає серверу запит, який той обробляє і пересилає клієнту лише отримані результати. Таким чином, клієнт зайнятий тільки обробкою отриманих результатів, а його робота не буде перешкоджати обміну інформацією по мережі між іншими клієнтами.

Важлива різниця обчислень в системі CSA від методів їх організації на великих ЕОМ полягає в тому, що клієнт є "інтелектуальним". Завдяки цьому з'являється можливість оптимізації обчислювального процесу за рахунок раціонального розподілення задач між робочою станцією та сервером.

Найбільш розповсюдженими застосуваннями архітектури CSA є реляційні бази даних. Реалізацію свого продукту у вигляді системи CSA пропонують більшість постачальників баз даних. В таких продуктах запити до бази даних (на мові SQL) спрямовують на сервер, який обробляє запит і повертає результат клієнту. В межах цієї архітектури дорогі комп'ютери з великою обчислювальною потужністю можуть спільно використовуватись декількома клієнтами, що дозволяє максимально задіяти доступні ресурси. Крім того, централізується та росте ефективність управління даними. Рішення CSA інтегрує в собі переваги і великих ЕОМ (централізоване управління і обробка даних), і персональних комп'ютерів.

Наведемо особливості архітектури CSA:

- Не потребує графічного інтерфейсу користувача. Логіка застосування CSA залишається незалежною від інтерфейсу з користувачем, тому зовсім необов'язково використовування графічного інтерфейсу. Разом з тим, такі застосування часто реалізуються на базі популярних операційних систем, які використовують GUI (Graphical User Interface).
- Не обов'язково орієнтована на бази даних.
- Не збільшує швидкість розробки програм.
- Не збільшує повторного використання програмного коду.
- Програми архітектури CSA не керуються подіями. Не дивлячись на те, що логічні моделі CSA оптимально підходять для систем, які керуються подіями, сама архітектура ніяким чином не припускає обов'язкового використання орієнтованих на події операційних систем. З іншого боку, ці системи прекрасно підходять для реалізації клієнт-серверної архітектури.

На основі архітектури CSA народилася і швидко розвивається четверта хвиля еволюції ІС (з початку 90-х років), яка відрізняється від попереднього наявністю ієрархічності, поверненням до централізованої обробки та єдиним керуванням ресурсами.

Робота в неоднорідній обчислювальній мережі є однією з важливих переваг архітектури "клієнт-сервер". Вона може функціонувати в мережах, до складу яких входять комп'ютери, що працюють під керівництвом різних операційних систем та/або побудованих на різних апаратних платформах. Для адміністраторів таких мереж дуже важливо забезпечити взаємодію всіх робочих станцій в процесі роботи. Це досягається завдяки підтримці сервером власних включень.

Основною ідеєю є положення, що кожне клієнтське середовище повинне мати "внутрішню" підтримку. Замість використання шлюзів і введення нових протоколів кожне широко розповсюджене мережеве програмне забезпечення повинно підтримуватися сервером без яких-небудь змін. Це відкриває перед розробниками нові можливості. Вся корпоративна обчислювальна система може працювати як єдине ціле незалежно від обчислювальної платформи, що використовується.

Інколи виникає необхідність в тому, щоб програма працювала на декількох платформах. Хоча це потребує значних зусиль з боку розробника, отримані в результаті переваги перекриють витрати. Наведемо загальні вимоги до розробки таких застосувань:

- можливість виконання на декількох платформах;
- однакові інтерфейси і логіка роботи на всіх платформах (мається на увазі тільки подібність схем екрана, елементів меню і діалогової інформації; при цьому необхідно використовувати стиль, характерний для відповідної операційної системи, але інформація, яка надається користувачу, повинна бути максимально узгоджена між різними платформами);
- інтегрованість з "рідною" операційною середою;
- однакова поведінка на різних платформах;
- узгоджена підтримка незалежно від платформи.

Реалізувати програми одночасно в декількох середовищах дуже непросто, тому з'явилися інтегровані середовища розробки, які полегшують цю задачу. Такі середо-

вища як Windows Open System Architecture (WOSA) та Win32 корпорації Microsoft або загальне відкрите програмне середовище UNIX COSE та AppWare Foundation компанії Novell значно полегшують перенесення програм між різними середовищами. Багато компаній приймають рішення про вкладення коштів саме враховуючи підтримку багатьох платформ.

Розподілені обчислення передбачають розподіл робіт між декількома машинами. Обробляти дані (чи запити), що поступають з клієнтських систем, можуть і декілька комп'ютерів, тобто розподілені обчислення можна роздивлятися як обчислення з одним клієнтом і декількома серверами.

Розподілені обчислення вигідні і користувачу, і компаніям, оскільки вони є одним з видів обчислень типу "клієнт-сервер", в результаті чого користувачі отримують збільшену пропускну здатність мережі і можливість багатозадачної роботи, а компанія в цілому має вигоди за рахунок максимального використання обчислювальних ресурсів, що знижує витрати і збільшує ефективність. Обчислення в неоднорідному середовищі дають можливість всім системам взаємодіяти між собою і перетворити ці системи і мережі в єдине ціле.

В архітектурі CSA прикладна програма розподіляється між двома компонентами: клієнтом і сервером, кожен з яких виконує окремі специфічні функції. Оскільки обробка може здійснюватися в будь-якому місці мережі, обчислення типу "клієнт-сервер" надають можливість ефективного масштабування.

Існують поняття "товстого" та "тонкого" клієнта. Основна ідея архітектури "тонкого" чи "прозорого" клієнта складається в тому, що потужні програми виконуються на потужному сервері, а "тонкий" клієнт лише користується результатами цього виконання. Проілюструємо цю ідею на прикладі використання архітектури "клієнт-сервер" для баз даних (рис. 1.32)

На рис. 1.32 зображена послідовність дій для виконання запиту SQL програмою "тонкого" клієнта:

- клієнт передає запит на виконання SQL-запиту;
- сервер отримує запит SQL, виконує його та готує результат виконання для "тонкого" клієнта;
- клієнт отримує результат запиту SQL.

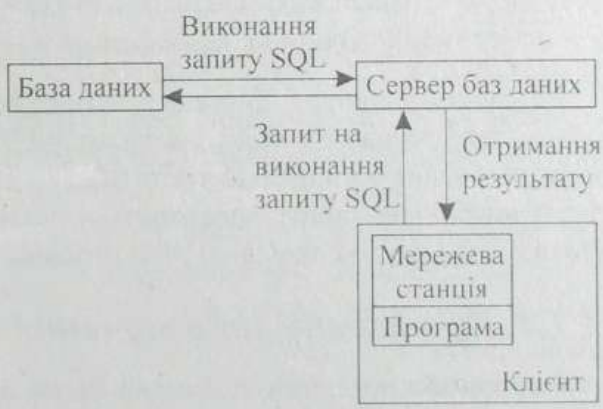


Рис. 1.32. Дворівнева архітектура "клієнт-сервер"

Переваги такого підходу до виконання програм стають очевидними у ситуації, коли база даних з яких-небудь міркувань (наприклад, міркувань безпеки або розмірів бази даних) знаходяться на сервері.

По-перше, якщо база даних зберігається на сервері з міркувань обмеження доступу до інформації, то завантаження копії на мережеву станцію робить цю задачу дуже складною, якщо не неможливою. По-друге, якщо розміри бази даних досить суттєві, то завантаження цієї бази декілька разів на день може стати серйозною про-

блемою. По-третє, запит SQL на досить слабкій машині може виконуватись набагато довше, ніж це було передбачено розробником, а може й взагалі не виконатись. В такій ситуації існують два виходи: купити більш потужне апаратне забезпечення або скористуватися новою програмною технологією, яка дозволить побудувати більш потужну систему програм, що легко масштабуються.

На рис. 1.33 наведена класична архітектура обміну даними в мережі.

Рис. 1.33. Класична архітектура обміну даними в мережі

В теперішній час найбільш часто використовують три- та чотирирівневу архітектуру, що не в останню чергу пов'язано з розповсюдженням Internet (рис. 1.34)

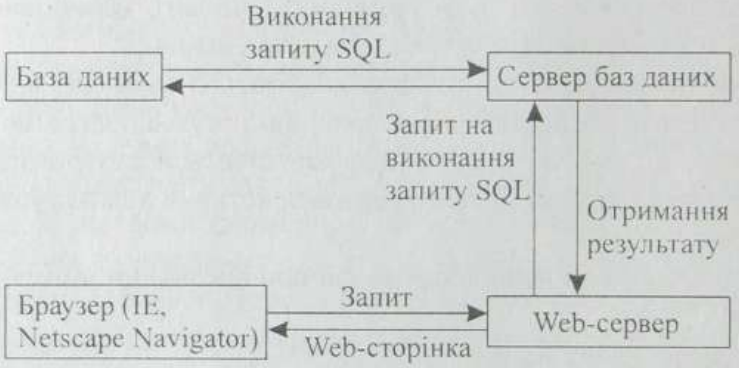


Рис. 1.34. Трирівнева архітектура "клієнт-сервер"

Розглянемо типову ситуацію: користувач завантажив з Web-сервера Web-сторінку, сформував та надіслав запит на Web-сервер. Web-сервер в свою чергу сформував запит та надіслав текст SQL на сервер баз даних якоїсь корпорації. Результат виконання запиту повертається по тому ж шляху та досягає користувача у вигляді комфортно оформленої Web-сторінки.

1.7.1. Програмне забезпечення

Розглянемо програми, побудованих на архітектурі "клієнт-сервер", що використовуються найчастіше. До цієї категорії можна віднести програми обробки електронної пошти (E-mail). Повідомлення не досягає адресата відразу, а проходить через ланцюг E-mail-серверів. Другий приклад. Коли користувач Internet користується якимнебудь Web-браузером (наприклад, Internet Explorer або Netscape Navigator), то він користується "тонким" клієнтом, який підключається до потужних серверів, що здатні зберігати десятки тисяч Web-сторінок та обробляти десятки тисяч повідомлень на добу.

Розглянемо програмне забезпечення побудови баз даних згідно архітектури "клієнт-сервер". На ринку серверних програм зараз існує багатий вибір серверів загального призначення, наприклад, продукт компанії Online Dynamic Server під назвою Informix, який представляє собою класичний SQL-сервер — результат об'єднання широкого спектру програмних продуктів. Цією компанією також пропонується об'єктно орієнтований продукт під назвою Universal Server.

Ще одна гілка продуктів — Sybase: SQL Server — сервер масштабу підприємства та SQL Anywhere — сервер масштабу робочої групи. Корпорація Microsoft поширює продукт SQL Server, що розроблявся сумісно з Sybase.

Серед програмного забезпечення, що дозволяє створити "тонкі" клієнтські програми існує також широкий вибір. В принципі, це дозволяє робити будь-яке сучасне середовище візуального програмування. Відмітимо потужний PowerBuilder, недорогі, але досить якісні продукти компаній Borland — Delphi та C++, та Microsoft — Visual Basic та Visual C++ .

1.7.2. Моделі побудови архітектури "клієнт-сервер"

Розглянемо моделі побудови архітектури CSA. Історично першою виникла дворівнева архітектура. Ця система складається з компоненти користувача та сервера баз даних (БД). До цього типу архітектури відносяться моделі RDA (Remote Data Acces — дистанційний доступ до даних) та DBS (Data Base Server — сервер баз даних). Вони звичайно використовуються при розробці не дуже відповідальних програм з досить простими транзакціями, наприклад, для системи підтримки прийняття рішень (СППР) або для розв'язання задач на рівні підрозділу. Розглянемо ці моделі докладніше.

Модель RDA

Модель RDA (рис. 1.35) суттєво відрізняється від моделі FS (File Server — файл-сервер) характером доступу до інформаційних ресурсів. Коди компонента представлення і прикладного компонента об'єднані і виконуються комп'ютером-клієнтом. Доступ до інформації підтримується або операторами спеціальної мови (наприклад, SQL), або запитами функцій певної бібліотеки (в цьому випадку є спеціальний інтерфейс API).

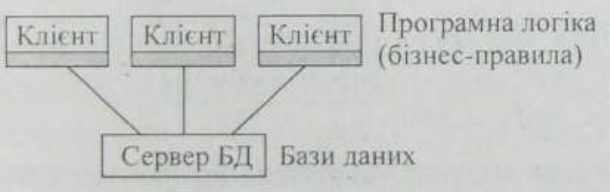


Рис. 1.35. Модель RDA

Клієнт відправляє запит до інформаційних ресурсів (наприклад, до баз даних) через мережу віддаленому серверу. На сервері функціонує ядро системи управління базами даних (СУБД), яке обробляє запити, виконуючі описані в них дії, і повертає клієнту результат, оформлений як блок даних. Ініціаторами маніпуляцій з даними є прикладні програми, що виконуються на комп'ютерах-клієнтах. Ядру СУБД відводиться пасивна роль обслуговування запитів та обробки даних.

Модель RDA дозволяє позбутися наступних недоліків, притаманних системам з централізованою архітектурою: мейнфреймам і файловим серверам (модель FS):

- перенос компонента представлення та прикладного компонента на комп'ютери-клієнти значно розвантажує сервер БД, зменшує кількість процесів в операційній системі сервера;
- сервер БД звільняється від невластивих йому функцій — процесор (чи процесори) сервера повністю завантажуються операціями обробки даних, запитів і транзакцій;
- в порівнянні з моделлю FS, різко знижується навантаження мережі, по ній передаються тільки запити і оброблені результати запитів;
- в порівнянні з архітектурою мейнфреймів, клієнтські місця стають "інтелектуальними" — на відміну від терміналів, вони мають свої власні ресурси, тому обчислювальна потужність сервера (як правило, потужної і дорогої машини) використовуються тільки для обробки даних.

Модель DBS

Модель DBS (рис. 1.36) в останній час стала досить популярна. Як правило, саме вона реалізується в реляційних СУБД, таких як InterBase, Oracle, Informix, Ingres, Sybase. Крім перерахованих вище переваг, її основу складає механізм процедур, що зберігаються — засобів програмування SQL-сервера.

Процедури зберігаються в словнику БД, можуть розподілятися між декількома клієнтами та виконуватися на тому ж комп'ютері, на якому функціонує SQL-сервер.

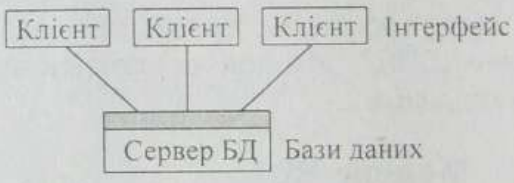


Рис. 1.36. Модель DBS

В моделі DBS компонент представлення виконується на комп'ютері-клієнті, в той час, як прикладний компонент, оформлений як набір процедур, що зберігаються, виконується на сервері БД. Там також знаходиться і ядро СУБД.

Крім переваг, притаманних моделі RDA, модель DBS має ряд власних переваг:

- можливість централізованого адміністрування;
- додаткове зниження трафіка за рахунок виконання процедур, що зберігаються на SQL-сервері;
- розподіл процедур, що зберігаються, між декількома прикладними програмами дозволяє організувати задачі підтримки цілісності даних незалежно від прикладних програм, що використовують ці дані.

В останній час все більшого поширення набуває трирівнева архітектура CSA. Вона дозволяє створювати найбільш гнучкі і ефективні системи. Ця модель має назву AS (Application Server — сервер прикладних програм).

Модель AS

В моделі AS (рис. 1.37) процес, який виконується на комп'ютері-клієнті, відповідає за інтерфейс з користувачем. Звертаючись до прикладного компонента, розміщеному на сервері, цей процес грає роль програми-клієнта (Application Client). При-

кладний елемент, реалізований як група процесів, що виконують прикладні функції, називається програмним сервером (Application Server). Всі операції над інформаційними ресурсами можуть здійснюватись відповідним компонентом, розміщеним на іншому сервері, по відношенню до якого AS грає роль клієнта.

В моделі AS, як правило, реалізована трирівнева система розподілу функцій, де прикладний компонент виділений як найважливіший ізольований елемент прикладної програми, для роботи якого використовуються універсальні механізми багатозадачної операційної системи і стандартизовані інтерфейси з іншими компонентами.

Рис. 1.37. Модель AS

В принципі, архітектура CSA не обмежується трьома рівнями і може бути багаторівневою.

Для моделей RDA і DBS фундаментом є дворівнева схема розподілу функцій. В моделі RDA прикладні функції реалізуються програмою-клієнтом, а в моделі DBS відповідальність за їх виконання бере на себе ядро СУБД. В першому випадку прикладний компонент зливається з компонентом представлення, в другому — інтегрується з компонентом забезпечення доступу до інформаційних ресурсів.

В моделі AS реалізована трирівнева схема розподілу функцій, де прикладний елемент виділений як ізольований компонент прикладної програми. Модель AS є фундаментом для моніторів обробки транзакцій.

1.7.3. Засоби розробки програм в архітектурі CSA

Важливу роль у розробці прикладних програм в архітектурі CSA має інструментарій, яким володіє розробник. Процес розробки в даному випадку є, фактично, процесом декомпозиції прикладних програм та розподілу їх між клієнтом і сервером. Чим більше рівнів в системі, тим складнішим стає завдання програміста. Це завдання можна виконувати з використанням мов програмування різних поколінь.

Під засобами розробки третього покоління (3GL) розуміють мови C, C++, Visual C++, Pascal тощо. Програми, написані на цих мовах, відрізняються високою ефективністю, однак розробка такими засобами коштує надзвичайно дорого.

Засоби розробки четвертого покоління (4GL) дозволяють приховати складності взаємодії операційної системи з базою даних. Розробник керує компонентами більш високого рівня і тому може відносно легко і швидко будувати прикладні програми. Однак такі програми, як правило, дуже громіздкі і часто менш ефективні. Сьогодні існує досить багато інструментальних засобів розробки 4GL, наприклад: Gupta SQL Windows, Powersoft PowerBuilder, а також програмні продукти компанії Progress та Oracle. Більшість з них підтримують платформи Windows, UNIX, OpenVMS, Macintosh.

В більшості випадків, це — оптимальний варіант для корпоративних користувачів. Інструментальні засоби четвертого покоління дозволяють абстрагуватись від

особливостей операційної системи і конкретних баз даних, що робить прикладні програми переносними між різними платформами.

Особливе місце серед інструментарію розробки програм в архітектурі CSA займають так звані CASE-системи, які можуть моделювати логіку задачі та дозволяють розробляти програми, алгоритм роботи яких і організація даних максимально оптимізовані. Подібні продукти випускають багато компаній, однак вони надзвичайно дорогі й виправдовують себе лише в великих корпоративних системах.

Засоби розробки п'ятого покоління (5GL) призначені для проектування надзвичайно складних програм. Розробник оперує тільки компонентами, які моделюють процеси взаємодії клієнта з сервером, причому основну масу програмного коду генерує засіб розробки, тобто розробник може створити програму, не написавши жодного рядку програмного коду. Такі продукти логічно представляють значні компоненти програми і дозволяють розробнику візуально комбінувати їх в єдине ціле. Такі програми можуть забезпечити зв'язок з базами даних, відеообробкою зображень і передачею повідомлень. Складність створення програм архітектури "клієнт-сервер", так би мовити, "маскується" від розробника. Прикладом такого продукту є Visual AppBuilder компанії Novell.

Висока гнучкість цих засобів розробки має і свої негативні сторони. Розробникам доводиться вивчати нову технологію (ООП, CASE-методи, принципи моделювання даних і оптимізації бізнес-процесів BPR).

Так чи інакше, будь-які засоби обробки не усувають необхідності у старанному плануванні та структуруванні процесу створення прикладних програм.

Значним фактором є вартість засобів розробки. Якщо найбільш розповсюджені продукти (такі як PowerBuilder чи Delphi), коштують 1–2 тисячі доларів, то витрати на більш потужні системи складають від 25 до 100 тисяч доларів.

1.7.4. Практичне використання архітектури CSA

Переходячи до питання практичного використання архітектури CSA, розглянемо, як її поява вплинула на планування та розвиток бізнес-процесів.

В галузі промисловості, яка виникла після 1980р. (сотовий телефонний зв'язок, промислові інкубатори, тестові медичні лабораторії тощо), мейнфрейми рідко відігравали головну роль, на відміну від архітектури "клієнт-сервер".

Перепроектування бізнес-процесів (BPR — Business Process Reengineering) є ключовим фактором, що обумовлює зміну архітектури обчислень в інформаційних технологіях. BPR визначається як "фундаментальне переосмислення і радикальне перепроектування бізнес-процесів з метою досягнення значного покращення у критично важливих сучасних критеріях продуктивності, таких як вартість, якість, сервіс і швидкість".

Для розуміння взаємодії бізнес-стратегій з архітектурою інформаційних технологій скористуємося моделлю, яку створив Джон Хендерсон з Массачусетського технологічного інституту.

В даній моделі основна бізнес-платформа являє собою набір стратегій, ринків, технологій виробництва продуктів і ресурсів, вибраний організацією як такий, що відповідає встановленій меті. Звідси випливає, що бізнес-архітектура — це той набір товарів та послуг, організаційних структур, процесів керування, розподілу ресурсів,

цінностей та стимулів, що є необхідним для впровадження основної бізнес-платформи. Під відповідною основною інформаційною платформою (IT — Information Technologies) розуміється ряд адекватних комп'ютерних технологій, доступних компанії, і засоби, якими ці технології можуть бути реалізованими для підвищення конкурентноздатності.

Архітектура IT — це визначений набір архітектур та IT-продуктів, вибраний для реалізації основної IT-платформи, а також інфраструктури підтримки, механізми прийняття рішень і адміністративні механізми, що використовуються для розгорнення цих структур.

З появою архітектури "клієнт-сервер" малось на увазі, що її реалізація дасть не тільки якісь явно невідчутні переваги, але й реальну економію витрат. Але насправді реалізацію систем CSA можна поставити на один рівень з традиційними системами. В деяких випадках вони навіть обходяться дорожче, і хоча загальна вартість таких систем приблизно однакова, супутні витрати можуть відрізнятись.

На сьогодні, обчислення типу "клієнт-сервер" — це дуже цінне рішення. Воно дозволяє підвищити ефективність роботи користувача, гарантує інтеграцію самого користувача в загальне корпоративне обчислювальне середовище, зберігаючи при цьому незалежність за рахунок того, що робоча станція користувача має автономні обчислювальні ресурси.

Обчислення CSA вигідні всім користувачам, які пов'язані з інформаційними системами компаній. Користувачам надаються гнучкість, продуктивність і необхідні ресурси. Для компанії це вигідно тому, що забезпечується ефективне використання ресурсів і економляться кошти. Користувачі виграють за рахунок можливості використання потужних обчислювальних ресурсів сервера.

Використання архітектури CSA надає багато переваг. Так, дорогі комп'ютери з великою обчислювальною потужністю можуть використовуватися колективно. Обчислювальна потужність клієнта зростає пропорційно обчислювальній потужності сервера. Завдяки цьому поступове старіння клієнтських машин може бути компенсоване збільшенням потужності сервера, що обійдеться значно дешевше. Забезпечуючи на сервері централізоване управління і доступ до логічних чи фізичних засобів, розробники можуть створювати інтегровані прикладні програми.

1.8. Протоколи мереж та методи доступу

Розглянемо класифікацію та основні характеристики протоколів, що використовуються в сучасних комп'ютерних мережах. Класифікація базових протоколів наведена на рис. 1.38.

Універсально-спеціалізовані протоколи:

- **HTPP** — Hyper Text Transfer Protocol — протокол передачі гіпертексту при взаємодії користувачів з Web-сервером);
- **FTP** — File Transfer Protocol — протокол обміну файлами;
- **SLIP** — Serial Line IP — протокол, який дозволяє комп'ютерам використовувати протокол TCP/IP по телефонних лініях;
- **PPP** — Point-to-Point Protocol — це більш потужній варіант SLIP з ущільненням даних та корегуванням помилок.

- **PPTP** — Point-to-Point Tunneling Protocol — варіант PPP з підтримкою віртуальних мереж;
- **RTP** — Real Time Transport Protocol — протокол реального часу;
- **LDAP** — Lightweight Directory Access Protocol — спрощений протокол доступу до мережевого протоколу;
- **WebNFS** — протокол розширення набору протоколів NFS (прискорює передачу файлів і підвищує продуктивність Web-вузлів);
- **NFS** — Network File System — мережева файлова система;
- **RSVP** — Resource ReserVation Protocol — протокол резервування ресурсів;
- **Mbone** — Multybone — протокол для багатоадресної передачі;
- **CIFS** — Common Internet File System — протокол альтернативний FTP;
- **IPX** — Internetwork Packet Exchange — транспортний протокол для об'єднання існуючих мереж.

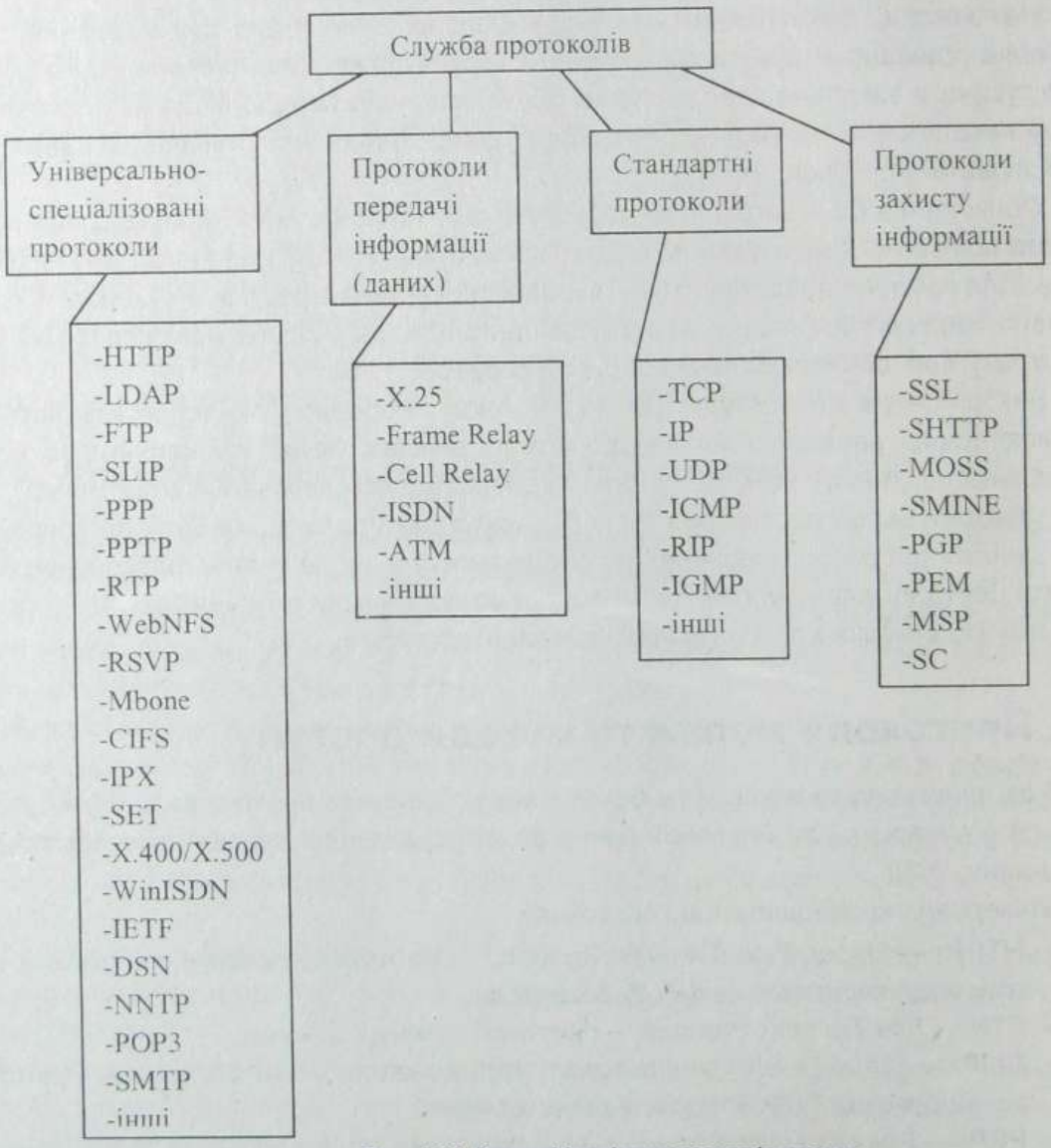


Рис. 1.38. Класифікація протоколів мереж

Протоколи передачі інформації (даних):

- **X.25** — протокол об'єднання передачі тільки даних по каналам зв'язку низької якості Hard Wave (HW) у мережу (існує з 1976р., займає перші три рівня моделі OSI);
- **Frame Relay** — дозволяє передавати не тільки дані, а й оцифрований голос, зображення. Це — протокол, що описує інтерфейс доступу до мереж із швидкою комутацією протоколів. Дозволяє об'єднати такі джерела інформації як телефони, відеокамери, мережі, комп'ютери;
- **ISDN** — Integral Services Digital Networks (цифрові мережі з комплексними послугами) — протокол для передачі: голосу, даних, зображень; забезпечує високу якість передачі, широкий спектр сервісів;
- **ATM** — Asynchronous Transfer Mode (Cell-Relay, B-ISDN) — асинхронний режим доставки інформації. Протокол (технологія) передачі по каналам зв'язку різнорідної інформації. Протокол ATM по структурі аналогічний протоколу ISDN, але відрізняється: швидкістю та методами передачі даних.

Стандартні протоколи:

- **TCP** — Transmission (Transport) Control Protocol — транспортний протокол чи протокол керування транспортуванням;
- **IP** — Internet Protocol — протокол адресації в Internet;
- **UDP** — User Datagram Protocol (протокол дейтаграм користувача) — подібний за функціями протоколу TCP, але простіший і менш надійний;
- **ICMP** — Internet Control Message Protocol (протокол керуючих повідомлень Internet) — керує мережевими повідомленнями про помилки);
- **RIP** — Routing Information Protocol (протокол маршрутизації інформації) — використовується при обміні інформацією в мережевих шлюзах, що використовують протокол IP;
- **IGMP** — Internet Group Messages Protocol — протокол групових повідомлень Internet;

Протоколи захисту інформації:

- **SSL** — Secure Sockets Layer (рівень захищених сокетів) — забезпечує захист Web-серверів на мережевому рівні;
- **SHTTP** — Secure Hyper Text Transport Protocol (захищений протокол HTTP) — забезпечує захист на прикладному рівні і використовується в Internet для зв'язку з віддаленими файлами підприємств;
- **MOSS** — Multi-part Object Security Standart — стандарт захисту багатоелементних об'єктів;
- S/MINE — протокол захисту даних в електронній пошті;
- **PGP** — Pretty Good Privacy ("досить добра конфіденційність") — набір алгоритмів та програм для високонадійного шифрування повідомлень з використанням ключів;
- **PEM** — Privacy Enchanced Messaging ("повідомлення підвищеної секретності") — протокол електронної пошти, що забезпечує шифрований зв'язок;
- **MSP** — Message Security Protocol — протокол захисту повідомлень;
- **SC** — Secure Courier ("таємний агент").

1.9. Концепції та принципи роботи з мережами

Більша частина мереж підтримується в належному стані завдяки зусиллям спеціалістів, що звуться системними (або мережевими) адміністраторами. Системний адміністратор звичайно відповідає за належний технічний стан мережі, правильне функціонування програмного забезпечення, наглядає за розподіленням і використанням ресурсів мережі тощо. Усі користувачі мереж (зокрема локальних) час від часу мають контактувати з адміністратором у таких випадках як порушення в роботі мережі, використання нового програмного забезпечення, вирішення виникаючих час від часу проблем.

При роботі в будь-якій мережі, по-перше, необхідно приєднатися до цієї мережі. Після фізичного підключення здебільшого треба зареєструватися в мережі для того, щоб можна було ідентифікувати користувача при проведенні різних мережевих транзакцій. Після реєстрації, яка звичайно включає в себе вхід у систему під якимось ім'ям, найчастіше необхідно ввести пароль користувача. Пароль звичайно задається або при першій реєстрації, або адміністратором мережі.

Слід зазначити, що в багатьох мережах, щоб забезпечити безпеку, зміна паролів користувачів відбувається регулярно (наприклад, у мережі NASA (державна організація США, що займається вивченням космосу) така зміна відбувається кожні 32 години, в результаті чого потенційні порушники не мають часу на довготривале втручання в роботу системи.

Якщо користувач успішно зареєструвався в мережі, то він може починати використовувати ресурси системи або надавати свої ресурси для використання іншим користувачам. Коли адміністратор розподіляє ресурси мережі, переважно призначаються користувачі, що мають права користування тим чи іншим ресурсом. Звичайно, для використання якогось ресурсу необхідно мати право на доступ до нього.

Варіанти підключення до мережі схематично наведені на рис. 1.39.

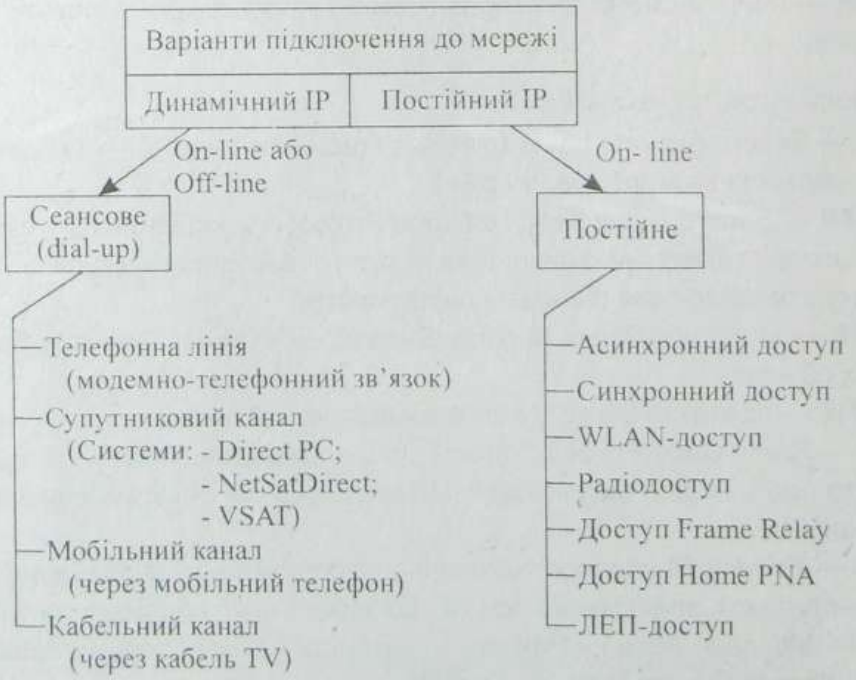


Рис. 1.39. Підключення до мережі: PNA — Personal Network Area; ЛЕП — лінія електропередач

При підключенні через мобільний канал використовується комп'ютер (Notebook, PDA) та мобільний телефон, що працює із спецпротоколами WAP (Wireless Access Protocol), GPRS (General Packed Radio Service) у мобільних мережах GSM, а також протокол Bluetooth ("блакитний зуб") для бездротового зв'язку в режимі GPRS+Bluetooth. Крім того може бути використаний інфрачервоний порт комп'ютера.

Асинхронне підключення здійснюється по телефонній лінії за технологією ADSL (Asynchronic Digital Service Line). При цьому лінія не зайнята для телефонних розмов.

Синхронний доступ реалізується по виділеному каналу: телефонна лінія (виділений телефонний номер); оптичний канал з цифровою лінією ISDN (Integrated Services Digital Network) або DSL (цифрова абонентська лінія).

WLAN-доступ — це приєднання LAN декількох користувачів (наприклад, одного будинку) з одним радіоканалом Radio Ethernet (антена на даху) до Internet.

Розглянемо приклад концептуальної моделі локальної кільцевої мережі інтелектуальних АРМ (рис. 1.40).

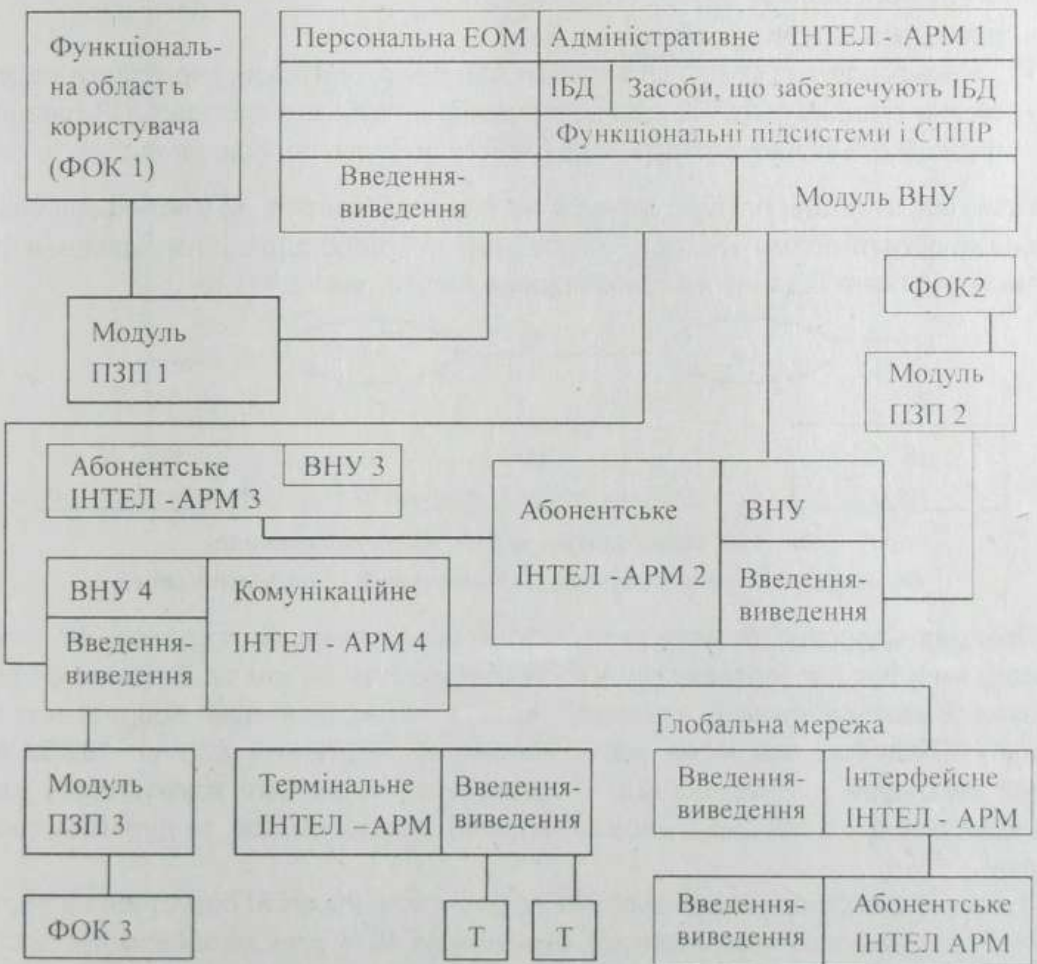


Рис. 1.40. Приклад концептуальної моделі локальної кільцевої мережі інтелектуальних АРМ

Ця система забезпечує засоби інтелектуального банку даних (ІБД), засоби навчання Інтел-АРМ і користувача, засоби діалогу з ІБД, організаційно-технологічні засоби. Система прийняття, підтримки і узгодження рішень (СППУР) використовується

керівником для аналізу і синтезу в умовах "керівничої" неозначеності. Модуль перетворення зовнішнього представлення (ПЗП) функціональної області користувача (ФОК) у внутрішнє формалізоване представлення на функціональному і аналітичному рівнях характеризується своїми інструментально-логічними засобами. Модуль перетворення внутрішнього представлення даних, знань, цілей Інтел-АРМ (ВНУ) використовується для мережевого представлення інформації. Зв'язок між Інтел-АРМ можливо здійснити як через комутативну телефонну мережу, так і через виділені телефонні лінії.

1.9.1. Принципи роботи з мережами

Перелічимо загальні принципи роботи з мережами: багаторівневий режим, комутація пакетів, адресація на основі міжмережевого протоколу, транспортування повідомлень, доменних імен тощо.

Багаторівневий режим роботи будь-якої сучасної мережі на основі моделі OSI будується на наступних рівнях:

- рівень апаратури та зв'язку;
- рівень базового ПЗ (ОС UNIX, Linux, Windows, браузери, спецутиліти тощо);
- рівень стандартного ПЗ для розширення і підтримки базового ПЗ (реалізація функцій електронної пошти, віддаленого доступу, пересилання файлів тощо).

Принцип комутації пакетів даних, а не комутації каналів як у телефонії реалізується маршрутизаторами, які "приймають рішення" щодо адреси направлення (перенаправлення) пакетів даних на основі адреси пакета (рис. 1.41).

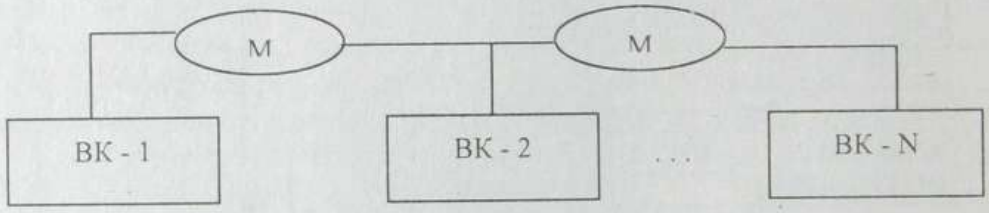


Рис. 1.41. Обмін даними на рівні комутації пакетів:
ВК — вузловий комп'ютер або UNIX-станція; М — маршрутизатор

Принцип адресації базується на основі міжмережевого протоколу IP (Internet Protocol), який був запропонований у 1974 р. Робертом Кеном та Вінтоном Сеффом. Протокол IP виконує функції "конверту", куди "вкладаються" дані. Конверт має свою унікальну IP-адресу, яка може мати, наприклад, наступний вигляд: **192.33.34.22**. Останні дві цифри у IP–адресі (**22**) — це адреса конкретного комп'ютера у мережі (найвищий рівень), а перший — номер (адреса) великої мережі, та ділянки у середині мережі.

Принцип транспортування окремих повідомлень (пакетів) базується на протоколі TCP (Transmission Control Protocol), який існує з 1974 року та забезпечує доставку повідомлень за потрібною IP-адресою. Відомо, що дані в мережі передаються порціями або пакетами (TCP–пакетами), тобто вся інформація, яку потрібно передавати, з технічних причин розподіляється на відносно невеликі пакети (приблизно по 1500 символів), які спочатку оформляються у вигляді порцій з відповідними IP-адресами, а потім передаються у мережу.

Інколи замість протоколу TCP використовують протокол UDP (User Datagram Protocol), який вважається простішим, ніж TCP, більш дешевим та використовується для пересилання коротких повідомлень. Але UDP не контролює пакети (порції), що губляться при пересиланні.

Принцип доменних імен DNS (Domain Names System) дозволяє замість числових імен комп'ютерів та мереж (IP-адреси) використовувати імена комп'ютерів, організацій, мереж тощо у вигляді так званих доменів. Наприклад, доменне ім'я може мати вигляд: **UX.CSO.UIUC.edu**, де **UX** — домен першого рівня, який характеризує ім'я реального комп'ютера (найнижчий рівень); **CSO** — домен другого рівня (відділ, в якому розташований комп'ютер); **UIUC** — домен третього рівня (ім'я університету де знаходиться відділ, в якому розташований комп'ютер); **edu** — домен четвертого рівня, що означає всі учбові заклади країни. Система доменних імен дозволяє використовувати не більше п'яти доменних рівнів. Схема ієрархії доменів представлена на рис. 1.42.

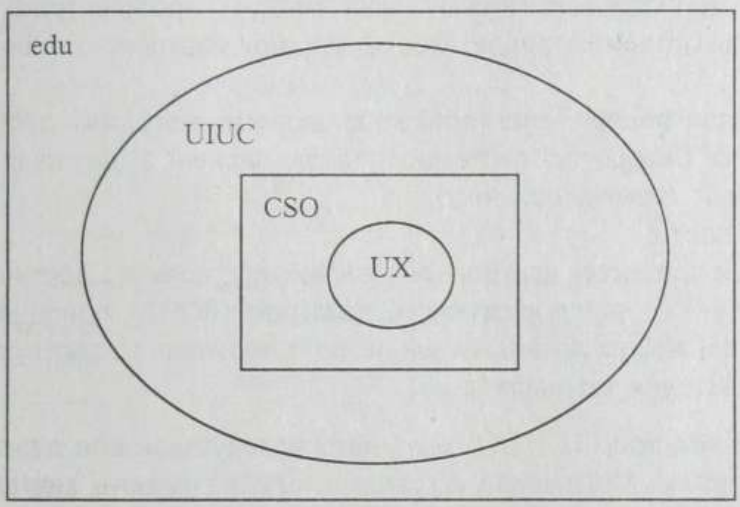


***Рис. 1.42. Схема ієрархії доменів для доменного імені** UX.CSO.UIUC.edu*

Розділ 2

Технічно-програмне забезпечення мереж

"Пам'ятай, що більшість слів — зайві"
(Народна мудрість)

2.1. Технічне забезпечення мереж

До технічного забезпечення мереж (мережевого обладнання) відносять наступні компоненти:

- апаратні: мейнфрейми; Host-ПК; міні-, мікро-ПК; робочі, графічні станції; мережеві комп'ютери, сервери; мости; маршрутизатори; концентратори (хаби) тощо;
- мережеві контролери (адаптери) з відповідними методами доступу;
- кабельні та бездротові системи, а також пасивні з'єднувачі, розгалужувачі, повторювачі, термінатори тощо;
- мережеві плати;
- мережевий процесор: контролери моноканалу, прямого доступу до локального оперативного запам'ятовуючого пристрою (ЛОПЗ), припинень; ЛОПЗ; мікропроцесор; арбітр локальної шини, порт керування адаптером, таймер, генератор тактових імпульсів тощо.

Мережевий адаптер (*LAN adapter*) використовується для підключення різних комп'ютерів у мережу. Мережевий адаптер інколи називають ***мережевою інтерфейсною картою*** (NIC — Network Interface Card).

Мережеві адаптери (МА) різних типів відрізняються:

- середовищами передачі даних, які вони підтримують;
- типом системної шини;
- продуктивністю;
- технологією, що вони підтримують тощо.

Мережевий адаптер встановлюється в середину комп'ютера для підключення його до мережі. Таким чином, кожен комп'ютер LAN використовує той чи інший тип мережевої інтерфейсної карти (МІК). МА "бере" дані з одного комп'ютера, перетворює їх у відповідний формат та пересилає їх по кабелю до МІК іншого комп'ютера. Плата цього іншого комп'ютера приймає дані, перетворює їх у форму, що прийнята для цього комп'ютера та пересилає їх у оперативну пам'ять.

Роботу МІК можна записати у вигляді восьми етапів:

1. Зв'язок комп'ютера з МІК.
2. Буферизація (для тимчасового зберігання даних).
3. Формування пакету.
4. Паралельно-послідовне перетворення.
5. Кодування-декодування.

6. Доступ до кабелю.
7. Квотування встановленого зв'язку.
8. Передача-прийом.

Інакше ці етапи можна назвати ступенями передачі даних з пам'яті одного комп'ютера у пам'ять іншого.

МІК поставляється з відповідними драйверами. Вони бувають 8-, 16- або 32-бітні.

У загальному випадку місце технічного забезпечення або мережевого обладнання можна показати у структурі всього мережевого ринку. Мережевий ринок іноді умовно поділяють на сегменти (рис. 2.1).

Рис. 2.1. Сегменти мережевого ринку

2.2. Модемне забезпечення

2.2.1. Можливості модему

За допомогою модему можна зв'язатися з іншим користувачем і обмінятися з ним файлами, незалежно від місця його розташування, або зіграти з ним у комп'ютерну гру, що підтримує модем. В разі факс-модему можна обмінюватися факсимільними повідомленнями.

Модем дозволяє користуватися послугами електронної дошки оголошень (BBS — Bulletin Board System), одержувати і приймати файли, спілкуватися з іншими користувачами, грати в ігри "on-line" (тобто, в режимі реального часу).

Можливе підключення до глобальних мереж, наприклад, FidoNet або Internet/Relcom/UANet. Підключившись до них, можна стати учасником множини телеконференцій, що дає можливість обмінюватися інформацією з різними людьми.

Коли комп'ютер використовують для обміну інформацією по телефонній мережі, необхідно мати пристрій, що може прийняти сигнал з телефонної мережі й перетворити його в цифрову інформацію. На виході цього пристрою інформація модулюється, а на вході — демодулюється, звідси й назва "модем". Призначення модему полягає в заміні сигналу, що надходить з комп'ютера (послідовність нулів та одиниць), електричним сигналом з частотою, що відповідає робочому діапазону телефонної лінії. Акустичний канал цієї лінії модем поділяє на рядки низької й високої частоти.

Смуга низької частоти застосовується для передачі даних, а смуга високої частоти — для прийому. Використовується багато методів кодування інформації, найбільш відомими з яких є метод FSK (Frequency Shift Keying — частотна маніпуляція) для швидкості передачі до 300 бод (***бод*** — одиниця швидкості передачі інформації, що дорівнює 1 біт/с) і метод PSK (Phase Shift Keying — фазова маніпуляція) для модемів зі швидкістю передачі до 2400 бод.

Метод FSK використовує чотири виділені частоти. При передачі інформації сигнал частотою 1070 Гц інтерпретується як логічний нуль, а сигнал частотою 1270 Гц — як логічна одиниця. При прийманні нуль відповідає сигналу 2025 Гц, а одиниця — 2225 Гц.

Метод PSK використовує дві частоти: для передачі даних — 2400 Гц, для прийому — 1200 Гц. Дані передаються по два біти, при цьому кодування здійснюється шляхом зсуву фази сигналу. Використовуються наступні зсуви фази для кодування:

- 0 градусів для поєднання бітів 00;
- 90 градусів для 01;
- 180 градусів для 10;
- 270 градусів для 11.

Існують також й інші види модуляції (DPSK, QAM, TCM). Модем виконують або у вигляді зовнішнього пристрою, який одним виходом приєднується до телефонної лінії, а іншим — до стандартного COM-порту комп'ютера (вихід RS232 згідно рекомендаціям CCITT V.24), або у вигляді звичайної плати, яка монтується на загальну шину комп'ютера. Внутрішні варіанти модемів можуть бути підключені як до звичайної шини ISA, так і до шини PCI.

Контролер модему — це спеціалізований мікрокомп'ютер типу Sc1107 або Sc1108, який містить восьмирозрядний АЛП, ПЗП 8 Кбайт, ОЗП 128 байт, таймер, командний регістр, контролер припинень, стек, порт введення-виведення. Якщо плата модему приєднана до системної шини ПК, тоді використовується "паралельний" контролер Sc1107. Якщо ж плата працює з комп'ютером шляхом Rs232, тоді використовується "послідовний" контролер Sc1108. В деяких платах функцію контролера виконує процесор 8031 із зовнішнім ПЗП (i2732, 2764) та мікросхемою 74ls373.

2.2.2. Міжнародні стандарти модемів

Найбільш поширені так звані Hayes-сумісні модеми, за ім'ям фірми-виробника одного з перших модемів. Такі модеми використовують At-команди (від англійського слова "attention" — "увага"), сумісні з модемом Hayes Smartmodem. Крім стандартного для всіх Hayes-сумісних модемів набору команд, кожний виробник зокрема пропонує користувачеві широкий спектр специфічних команд, що мають силу лише в моделях цієї фірми (наприклад, US Robotics, Rockwell, ZyXEL і т. ін.).

Крім сумісності набору команд, модем повинен відповідати деякому стандарту передачі інформації по телефонним лініям. Такими стандартами є рекомендації МККТТ (міжнародний консультативний комітет по телеграфії та телефонам), французький CCITT (Comite Consultatie International Telegraphique et Telephonique). В США й Канаді існує аналогічний стандарт Bell, майже ідентичний CCITT.

Найбільш поширені стандарти для модемів наведені в табл. 2.1.

Таблиця 2.1

Протокол	Швидкість передачі, біт/с (+/-0.01%)	Швидкість передачі, бод (+/-0.01%)	Модуляція	Частота, Гц
V.34	28.800	2400		
V.33	14.400	2400	128 -tcm	1800
V.33	12.000	2400	64 -tcm	1800
V.42bis	14.400	2400		1800
V.32bis	14.400	2400	128-tcm	1800
V.32bis	12.000	2400	64-tcm	1800
V.32bis	9.600	2400	32-tcm	1800
V.32bis	7.200	2400	16-tcm	1800
V.32	9.600	2400	32-tcm	1800
V.32 uncode	9.600	2400	16-qam	1800
V.32	4.800	2400	4-tcm	1800
V.29	9.600	2400	16-tcm	1700
V.29	7.200	2400	8-tcm	1700
V.29	4.800	2400	4-tcm	1700
V.27ter (Bell 208b)	4.800	1600	8-tcm	1700
V.27ter	2.400	1200	4-tcm	1800
V.26bis (Bell 208)	2.400	1200		
V.233bis		2400		
V.233 (Bell 202)		600		

Таблиця 2.1. Закінчення

Протокол	Швидкість передачі, біт/с (+/-0.01%)	Швидкість передачі, бод (+/-0.01%)	Модуляція	Частота, Гц
V.22bis	2.400	600	16-qam	1200 Orig. 2400 Ans.
V.22 (Bell 212a)	1.200	600	4-qam	1200 Orig 2400 Ans.
V.21 (Bell 103)		300	fsk	2025/2225 1070/1270
G3 Fax	У відповідності із стандартами T.30, V.17, V29,V.27ter			

Модеми, відповідні до стандартів із швидкістю до 2400 бод, можуть вільно обмінюватись інформацією. Треба зазначити, що рекомендація CCITT V.32 не є стандартом в повному розумінні цього слова, оскільки практично кожний великий виробник модемів швидкості вище 2400 бод додає один чи декілька специфічних протоколів передачі даних. Їх використання можливе тільки при зв'язку аналогічних модемів, але при цьому досягається надвисока швидкість передачі, завадостійкість й швидкість сполучення.

Найбільш поширеним і дешевим є протокол HST (High Speed Transfer — передача з високою швидкістю), розроблений фірмою US Robotics ще наприкінці 80-х років минулого сторіччя. Існують різновиди цього протоколу: H96, H14, H16, H19, H21, H28, які відрізняються лише швидкістю передачі інформації, яка відповідно складає 9600, 14400, 16800, 19200, 21600 та 28800 бод.

Завдяки дешевизні, широким можливостям модернізації, високій завадостійкості і швидкісним показникам протоколу HST, користувачі придбають широко відомі моделі US Robotics, такі як Sportster, Worldport, Courier.

Дуже поширені також модеми фірми ZyXEL, що мають специфічний протокол Zyx, який підтримує можливість передачі даних із швидкістю 19200 бод повним дуплексом. Велику популярність модеми ZyXEL отримали на початку 90 -х років виключно завдяки недосяжності для вітчизняного споживача інших марок модемів.

Найменш поширеними є модеми фірми Telebit марки Trailblazer та відомий протокол PEP (Packet Ensemble Protocol — протокол множини пакетів). Такі модеми багато коштують, але мають сильний і стійкий сигнал, спроможний ігнорувати навіть оборонні фільтри, що ставляться на АТС для запобігання безкоштовного користування модемами. Практично всі високошвидкісні модеми сумісні з менш швидкими стандартами.

2.2.3. Протоколи коригування помилок

Коли модем приймає дані, він фільтрує корисну інформацію від шумів у лінії. Для цього існують спеціальні протоколи коригування помилок. Найбільш відомий з них — MNP10.

MNP (Microsoft Network Protocols — мережеві протоколи Microsoft) — серія найбільш поширених апаратних протоколів, вперше реалізована на модемах фірми

Microsoft. Ці протоколи забезпечують автоматичну корекцію помилок і компресію даних.

Крім згаданого вище протоколу MNP10, існують протоколи MNP1, MNP2, MNP3, MNP4, MNP5 та MNP7. У теперішній час найбільш поширеним є протокол MNP5, тому що протоколи MNP10 та MNP7 встановлюють на спеціальних модемах, що працюють по виділених лініях. Після того як модем відокремив корисну інформацію від шумів у лінії, він фільтрує власне дані від службової інформації.

Розглянемо особливості протоколів MNP:

- **MNP1** — протокол, що використовує асинхронний напівдуплексний метод передачі даних. Це самий простий із протоколів MNP.
- **MNP2** — протокол, що використовує асинхронний дуплексний метод передачі даних.
- **MNP3** — протокол, що використовує синхронний дуплексний метод передачі даних між модемами (інтерфейс модем–комп'ютер залишається асинхронним). При асинхронній передачі використовується десять бітів на байт (вісім бітів даних, стартовий біт та зупиняючий біт), а при синхронній — лише вісім. Саме в цьому криється можливість прискорення обміну даними на 20%.
- **MNP4** — протокол, що використовує синхронний метод передачі, забезпечує оптимізацію фази даних, яка покращує недоліки протоколів MNP2 і MNP3. Крім того, при зміні числа помилок на лінії відповідно змінюється і розмір блоків переданих даних. При збільшенні числа помилок розмір блоків зменшується, збільшуючи можливість успішного проходження окремих блоків. Ефективність цього методу складає більше 20% у порівнянні з простою передачею даних.
- **MNP5** — додатково до методів MNP4, в протоколі MNP5 часто використовується простий метод ущільнення інформації. Символи кодуються ланцюжками, що часто зустрічаються в переданому блоці. Додатково кодуються довгі ланцюжки однакових символів. Звичайно при цьому текстові файли ущільнюються до 35% своєї вихідної довжини. Разом із 20% MNP4 це дає підвищення ефективності до 50%. Зауважимо, що при передачі вже ущільнених файлів (а в більшості випадків це так і є) додаткового збільшення ефективності за рахунок ущільнення даних модемом не буде.
- **MNP6** — додатково до методів протоколу MNP5, протокол MNP6 автоматично переключається між дуплексним та напівдуплексним методами передачі в залежності від типу інформації. Протокол MNP6 також забезпечує сумісність із протоколом V.29.
- **MNP7** — у порівнянні з ранніми протоколами використовує більш ефективний метод ущільнення даних.
- **MNP9** – використовує протокол V.32 і відповідний метод роботи, що забезпечує сумісність із низькошвидкісними модемами.
- **MNP10** — призначений для забезпечення зв'язку на низькошвидкісних лініях. Це досягається за допомогою наступних методів:
  - багатократного повторення спроби встановити зв'язок;
  - зміни розміру пакетів відповідно до зміни рівня перешкод на лінії;
  - динамічної зміни швидкості передачі відповідно до рівня перешкод лінії.

Всі протоколи MNP сумісні між собою знизу догори. При встановленні зв'язку відбувається встановлення найвищого можливого рівня MNP-протоколу. Якщо ж один із модемів, що зв'язуються, не підтримує протокол MNP, то MNP-модем працює без MNP-протоколу.

2.2.4. Режими MNP-модемів

MNP-модем забезпечує наступні режими передачі даних:

- **Стандартний режим**. Забезпечує буферизацію даних, що дозволяє працювати з різноманітними швидкостями передачі даних між комп'ютером і модемом, а також між двома модемами. В результаті підвищення ефективності передачі даних можна встановити швидкість обміну між комп'ютером та модемом вище, ніж модемом та модемом. У стандартному режимі роботи модем не виконує апаратного коригування помилок.
- **Режим прямої передачі**. Цей режим відповідає звичайному модему, що не підтримує MNP-протокол. Буферизація даних не відбувається, апаратне корегування помилок не виконується.
- **Режим із коригуванням помилок та буферизацією**. Це стандартний режим роботи при зв'язку двох MNP-модемів. Якщо віддалений модем не підтримує протокол MNP, зв'язок не встановлюється.
- **Режим із коригуванням помилок і автоматичним налаштуванням**. Цей режим використовується, коли заздалегідь невідомо, чи підтримує віддалений модем протокол MNP. На початку сеансу зв'язку, після визначення режиму віддаленого модему встановлюється один з трьох інших режимів.

2.2.5. Внутрішні та зовнішні модеми

Модеми бувають внутрішніми та зовнішніми (існують так само спеціальні типи модемів у вигляді PC-карт (PCMCIA), але вони призначені для комп'ютерів типу "ноутбук"). Внутрішні модеми виконані у вигляді плати розширення, що вставляється в спеціальний слот розширення на материнській платі комп'ютера. Зовнішній модем, на відміну від внутрішнього, виконаний у вигляді окремого пристрою, тобто в окремому корпусі і з власним блоком живлення.

Внутрішній модем

Переваги:

- Всі внутрішні модеми без винятку (на відміну від зовнішніх) мають вбудовану мікросхему FIFO (First Input First Output — першим прийшов, першим вийшов). Ця мікросхема забезпечує буферизацію даних. Звичайний модем при проходженні байта даних через порт щоразу робить спробу перервати роботу комп'ютера. Комп'ютер за допомогою спеціальних ліній IRQ (Interrupt Request — запит на переривання) на деякий час припиняє роботу модему, а потім знову відновлює її. Це уповільнює роботу комп'ютера в цілому. Мікросхема FIFO дозволяє використовувати припинення в декілька разів рідше. Це має велике значення при роботі в багатозадачних середовищах.

- При використанні внутрішнього модему зменшується кількість проводів. Так само внутрішній модем не займає місце на робочому столі.
- Внутрішні модеми є послідовним портом комп'ютера і не займають існуючих портів комп'ютера.
- Внутрішні модеми завжди дешевші за зовнішні.

Недоліки:

- Займають слот розширення на материнській платі комп'ютера. Це дуже незручно на мультимедійних машинах, на яких встановлена велика кількість додаткових плат, а також на комп'ютерах, що працюють серверами в мережах.
- Відсутні індикаторні лампочки, що дозволяють стежити за процесами в модемі.
- Якщо модем "завис", то відновити працездатність комп'ютера можна лише натиснувши кнопку перевантаження комп'ютера "RESET".

Зовнішні модеми

Переваги:

- Не займають слот розширення, і при необхідності їх можна легко відключити та перенести на інший комп'ютер.
- На передній панелі є індикатори, що допомагають зрозуміти, яку операцію модем виконує в даний час.
- При "зависанні" модему не потрібно перезавантажувати комп'ютер.

Недоліки:

- Необхідна мультикарта з вмонтованою мікросхемою FIFO. Без FIFO модем звісно буде працювати, але при цьому зменшується швидкість передачі даних.
- Зовнішній модем займає місце на робочому столі, і для нього необхідні додаткові дроти.
- Займає послідовний порт комп'ютера.
- Зовнішній модем завжди дорожче аналогічного внутрішнього.

Призначення індикаторних лампочок

1. **MR** (Modem Ready — модем готовий).
   Показує, що модем включений і готовий до роботи.
2. **TR** (Terminal Ready — термінал готовий).
   Цей індикатор горить, коли модем виявляє сигнал DTR (Data Terminal Ready — термінал даних готовий), переданий комунікаційною програмою.
3. **HS** (High Speed — висока швидкість).
   Цей індикатор починає світитись, коли модем працює з максимальною можливою для нього швидкістю.

4. **CD** (Carrier Detect — виявлення несучої).
   Світиться, коли модем виявляє несучу. Він повинний світитись під час з'єднання модемів і протягом усього сеансу зв'язку, поки один із модемів не "покладе трубку".
5. **AA** (Auto Answer — автовідповідь).
   Показує, що модем включений в режимі автовідповіді, тобто буде "сам" відповідати на всі вхідні дзвінки. Якщо модем виявляє вхідний дзвоник, то цей індикатор мерехтить.
6. **OH** (Off Hook — трубку піднято).
   Індикатор еквівалентний знятій трубці телефону. Він світиться, коли модем займає лінію.
7. **RD** (Receive Data — прийом даних).
   Мерехтить при прийомі комп'ютером даних.
8. **SD** (Send Data — відправлення даних).
   Мерехтить, коли комп'ютер посилає дані.

2.3. Класифікація програмного забезпечення мереж

В загальному випадку програмне забезпечення будь-якої мережі умовно поділяють на чотири групи:

- мережеві операційні системи (ОС);
- мережеві утиліти;
- комунікаційні пакети;
- спеціалізовані програмно-апаратні компоненти нижнього рівня мережі (наприклад, NetBIOS — мережева ОС введення-виведення).

Мережеві ОС забезпечують користувача засобами колективного доступу до периферійних пристроїв комп'ютера, здійснюють пересилку даних по мережі і контроль за роботою мережі: хто використовує мережу та здійснює доступ до даних та пристроїв, які ресурси є в мережі; здійснює сервісні послуги тощо.

Комунікаційні пакети підтримують зв'язок типу "точка-точка" або "ПК↔ПК" (тобто зв'язок двох абонентів) через лінії зв'язку для передачі файлів по стандартним протоколам.

Мережеві ОС можна класифікувати за такими основними ознаками:

- кількість процесорів, що підтримуються ОС — однопроцесорні, багатопроцесорні (наприклад, ОС Sun Solaris підтримує від 9 до 128 процесорів);
- ступінь централізації механізму планування процесів: з витискуючою багатозадачністю (Windows NT/2000/2003, UNIX); з невитискуючою багатозадачністю (Novell NetWare) ;
- спосіб обробки завдань — системи пакетної обробки, системи квантування часу (UNIX, VMS, Windows NT/2000/2003); системи реального часу (QNX).

Часто специфіку мережевої ОС визначає алгоритм розподілу між декількома процесами, наприклад, при механізмі планування з витискуючою багатозначністю рішення про перемикання ЦП з одного процесу на інший приймає ОС, а при невитискуючій багатозадачності поточне завдання (процес) виконується до тих пір, поки во-

но саме не "віддасть" керування мережевій ОС, і тільки після цього ОС вибирає з черги наступне завдання.

Інколи мережеві ОС класифікують наступним чином :

- з відкритим кодом — Linux, Free BSD;
- UNIX та UNIX-подібні ОС — HP-UX, AIX, Solaris 9, Caldera, UNIX Ware тощо;
- Windows-подібні ОС — Windows NT/2000/2003 ;
- NetWare 6;
- OS /400 тощо.

2.3.1. Загальний огляд мережевих ОС

Мережева операційна система необхідна для управління потоками повідомлень між робочими станціями та серверами. Вона може дозволити будь-якій робочій станції працювати з розподіленим мережевим диском чи принтером, який фізично не під'єднаний до цієї станції.

У деяких комп'ютерних мережах є виділений автономний комп'ютер, котрий виконує функції тільки файлового сервера. Такі системи називаються локальними обчислювальними мережами (ЛОМ) з файловим сервером. В інших, малих ЛОМ робоча станція може одночасно виконувати і функції файлового сервера. Це — ***одноранговi*** ЛОМ.

Компоненти мережевої операційної системи на кожній робочій станції та файловому сервері взаємодіють один з іншим за допомогою мови, що називається ***протоколом***. Одним із спільних протоколів є протокол фірми IBM NetBIOS (Network Basic Input Output System — мережева операційна система введення-виведення). Іншим розповсюдженим протоколом є IPX (Internet-work Packet Exchange — міжмережевий обмін пакетами) фірми Novell.

Перелік деяких мережевих операційних систем із зазначенням їх виробника наведено в табл. 2.2.

Таблиця 2.2

Операційна система	Виробник
Apple Talk	Apple
LANtastic	Artisoft
NetWare	Novell
NetWare Lite	Novell
Personal NetWare	Novell
NFS	Sun Microsystems
OS/2 LAN Manager	Microsoft
OS/2 LAN Server	IBM
Windows NT Advanced Server	Microsoft
POWERfusion	Performance Technology
POWERLan	Performance Technology
Vines	Ba
Windows 2000/2003	Microsoft

Далі будуть розглянуті деякі з цих операційних систем.

2.3.2. ОС NetWare фірми Novell

Novell була однією з перших компаній, що почали створювати ЛОМ. Вона виготовляла як апаратні засоби, так і програмні, однак останнім часом компанія Novell сконцентрувала зусилля на програмних засобах ЛОМ.

Далі наведені деякі характеристики програмних продуктів NetWare:

- В середовищі NetWare може працювати більша кількість програм, ніж в будь-якій іншій ЛОМ.
- ОС NetWare здатна підтримувати робочі станції, що керуються DOS, Windows, OS/2, UNIX, Windows NT/2000/2003, Mac System 7 та іншими ОС.
- NetWare може працювати з більшою кількістю різнотипних мережевих адаптерів, ніж будь-яка інша операційна система. Для досягнення поставлених цілей можна вибрати апаратні засоби від безлічі різних постачальників. З ОС NetWare можна використовувати ARCnet, EtherNet, Token Ring чи практично будь-який інший тип мережевого адаптера.
- ЛОМ NetWare може досягати величезних розмірів.
- ЛОМ NetWare надійно працює.
- Засобів захисту даних, що надає NetWare, більш, ніж достатньо, для більшості ЛОМ.
- NetWare допускає використання більше, ніж 200 типів мережевих адаптерів; більше ніж 100 типів дискових підсистем для зберігання даних, приладів дублювання даних та файлових серверів.

Фірма Novell має контракти про підтримку ОС NetWare з найбільшими та потужними з незалежних організацій, таких як Bell Atlantic, DEC, Hewlett Packard, Intel, Prime, Unisys та Xerox.

Структура та версії ОС NetWare

Файловий сервер в NetWare є звичайним ПК, мережева ОС якого здійснює керування роботою ЛОМ. Функція управління включає координацію робочих станцій, регулювання процесу розділення файлів та принтера в ЛОМ. Мережеві файли всіх робочих станцій зберігаються на жорсткому диску файлового сервера, а не на дисках робочих станцій.

Існує три основні версії ОС NetWare. Версія 2.2 може працювати на комп'ютері з МП 80286 і вище, що використовується в якості файлового сервера. При покупці ОС необхідно придбати ліцензію на число користувачів (5, 10, 50, 100). Версії ОС NetWare 3.12 та 4.0 орієнтовані на 32-розрядні шинні архітектури та процесори 80386, 80486 чи Pentium. Є також варіанти мережевої ОС NetWare, призначені для роботи під управлінням багатозадачних операційних систем OS/2 і UNIX. Версію 3.12 ОС NetWare можна придбати для 20, 100 чи 250 користувачів, а версія 4.0 має можливість підтримки до 1000 користувачів.

Всі версії ОС добре сумісні між собою, тому в одній і тій же комп'ютерній мережі можна мати файлові сервери з різними версіями ОС NetWare.

Деякі команди ОС NetWare

Для кращого розуміння функціональних особливостей ОС NetWare розглянемо її основні команди.

NPRINT — передача текстового файлу на принтер.

LOGIN — команда приєднання серверу (для від'єднання — **LOGOUT**)

WHOAMI — ідентифікація користувача (інформація про поточний сеанс).

USERLIST — видача прізвищ користувачів, приєднаних до ЛОМ в даний момент.

SEND — посилка повідомлення якому-небудь користувачу.

В NetWare розрізняють три типи накопичувачів: локальні, мережеві та пошукові. Локальні накопичувачі фізично приєднані до робочих станцій. Мережеві накопичувачі — це накопичувачі на жорстких дисках файлового сервера. Аналогічно тому, як в DOS, використовується засіб PATH для визначення списку накопичувачів і директорій, в яких за замовченням розшукуються додаткові програми, в ОС NetWare використовується поняття пошукового накопичувача.

MAP — перегляд поточного статусу накопичувачів (без параметрів), та їх переназначення (з параметрами).

ОС NetWare дозволяє маніпулювати файлами та директоріями різноманітними способами. Можна копіювати, знищувати, перейменовувати, записувати, роздруковувати та розділяти файли в ЛОМ. Є також визначена система прав доступу до файлів та директорій.

Як файли, так і директорії на сервері в ЛОМ під управлінням ОС NetWare мають атрибути. Ці атрибути можуть відміняти права, надані користувачам в ЛОМ.

RIGHTS — перелік прав поточного користувача для даної директорії.

NCOPY — копіювання файлів.

NDIR — перелік файлів даної директорії. На відміну від DOS-команди DIR, виводить додаткову інформацію для кожного файлу та директорії.

CAPTURE — перенаправлення друку на розподілений мережевий принтер.

Сервер та файлова система з ОС NetWare

Існують деякі відміни сервера ЛОМ NetWare від звичайного ПК. Для накопичувача на жорсткому диску цього комп'ютера застосована спеціальна структура форматування. Неможливо отримати доступ до жорсткого диску такого сервера, якщо ОС завантажена з дискети. Але для користувача ЛОМ, який працює під управлінням ОС і отримав доступ до серверу зі свого терміналу, жорсткий диск сервера уявляється просто як додатковий до вже існуючих.

Формат запису даних на жорсткому диску, який використаний в ОС NetWare, включає більшу кількість інформації про файли та директорії, ніж це було можливо в DOS. Файли в ОС NetWare, крім атрибутів "лише для читання", "скритий" та "архівний" можуть додатково мати атрибут "неподільний" та "подільний" (вказує на можливість поділу файлу в ЛОМ між багатьма користувачами одночасно). Крім цього, ОС NetWare додає до інформації про файл наступні елементи: початкову дату створення, ім'я створювача файлу, дату останнього доступу до файлу, дату останньої модифікації файлу, дату та час останньої архівації файлу.

Захист даних в ОС NetWare

Система захисту даних в ЛОМ NetWare включає в себе наступні засоби:

- захист від несанкціонованого приєднання до ЛОМ шляхом присвоєння імен та паролів користувачам, а також граничного доступу до ЛОМ користувачів з деякими іменами у визначений час доби;
- система довірених прав (*trustee rights*) дозволяє контролювати, до яких файлів та директорій може мати доступ користувач, а також які операції він може з ними здійснювати;
- система атрибутів для директорій чи файлів, які визначають можливість їх копіювання, перегляду, запису та розподілу в ЛОМ.

Для кожної директорії існує ***маска максимальних прав***, яка зберігає максимальні привілеї, котрі в ній може мати користувач. Нижче перелічені права, які можуть бути вказані в цій масці:

- право читання з відкритих файлів;
- право запису у відкриті файли;
- право відкривати файли;
- право створювати нові файли;
- право знищувати файли;
- право створювати, перейменовувати чи знищувати піддиректорії, та встановлювати довірені права над директоріями всередині директорії та її піддиректоріях;
- право виконувати пошук файлів у директорії;
- право модифікувати атрибуті файлу.

Відмовостійкість системи NetWare

Відмовостійкість є на сьогодні однією з найбільш важливих характеристик, і розробники NetWare приділили значну увагу цьому питанню. У версіях 2.2, 3.12 та 4.0 ОС NetWare застосована технологія SFT (System Fault Tolerant — система захисту у випадках відказу приладів). Ця технологія забезпечує безперебійну роботу файлового серверу при різного роду відмовах апаратних засобів. У всіх версіях NetWare є засоби мінімізації витрат даних у випадку фізичних пошкоджень поверхні накопичувачів. Система SFT пішла далі в цьому відношенні, запропонувавши методи дзеркального відображення дисків та дублювання дисків.

У системі NetWare є можливість контролю сигналів джерела постійного струму UPS (Uninterruptible Power Supply). В разі виявленні збою в подачі електроенергії ОС повідомляє про це користувачів і оповіщає їх, який проміжок часу є в їхньому розпорядженні для завершення своєї роботи. Після завершення цього проміжку часу ОС автоматично закриє всі файли в системі та вимикне себе.

Нарешті, технологія SFT пропонує систему TTS (трасування обробки запиту). Прикладні програми, що використовують цю систему, інтерпретують послідовність дій з базами даних як одну операцію: або всі дії виконані успішно, або жодної з них.

Порівняльні характеристики версій ОС Advanced NetWare 2.x

Ця версія була випущена ще у 1986 році. Однією з видатних особливостей ОС NetWare 2.0 була здатність з'єднувати до чотирьох різних мереж з одним файловим сервером.

У 1987 році вийшла NetWare SFT, яка відрізнялась від попередньої версії підвищеною відмовостійкістю та зберіганням даних.

Версія 2.15 NetWare і NetWare для Macintosh дебютували у 1988 році. Істотним недоліком цих версій був занадто великий час інсталяції (вона включала в себе час тестування жорсткого диску та могла продовжуватись добу чи навіть дві).

32-розрядна мережева ОС NetWare 386 була випущена у вересні 1989 року. В ній була значно покращена система захисту даних, підвищена продуктивність та гнучкість.

У версії 2.2 NetWare фірма Novell зібрала всі найкращі з попередніх версій NetWare. Всі варіанти версії 2.2 мають однакові можливості і однаковий рівень відмовостійкості. Покращений процес інсталяції, є підтримка VAP (Value Added Processes — процеси з додатковими можливостями) — окремих програмних модулів, що дозволяють файловому серверу виконувати деякі додаткові функції.

ОС NetWare 3.12

Ця ОС використовує можливості процесорів 80386, 80486 чи Pentium і характеризується:

- приєднанням до одного серверу — до 250 користувачів;
- об'ємом дискової пам'яті — до 32 терабайт;
- розміром файлу — до 4 Гбайт;
- тим, що один файл може розташовуватись на декількох накопичувачах;
- тим, що одночасно можуть бути відкриті до 100000 файлів.

ОС NetWare 3.12 має покращену систему захисту даних. Також новою є концепція NLM (NetWare Loadable Module — завантажувальні модулі NetWare) — програмних модулів, що завантажуються у файловий сервер в процесі роботи.

В сервері можуть зберігатись файли для робочих станцій з різними ОС: DOS, Windows, Macintosh, OS/2, UNIX, Linux. Для цього на сервері передбачені спеціальні атрибути для імен файлів.

Ще одна особливість цієї версії ЛОМ — у неї включено новий інтерфейс транспортного рівня TLI (Transport Layer Interface), що оснований на інтерфейсі ODI (Open Data-link Interface — відкритий канальний інтерфейс). Цей інтерфейс надає широкий діапазон можливостей для організації зв'язків, включаючи IPX/SPX, NetBIOS, LU 6.2 (APPC), іменовані канали зв'язку для робочих станцій, що працюють під керуванням DOS та OS/2, TCP/IP, інтерфейс Berkley 4.3 Sockets та UNIX System V Stream/TLI.

Порівняння характеристик ОС NetWare 2.2 та NetWare 3.12 наведено в табл. 2.3.

Таблиця 2.3

Характеристика	NetWare 2.2	NetWare 3.12
Тип ОС	16 біт	32 біта
Мінімальний ЦП сервера	80286	80386
Мінімальний об'єм ОЗП на 50 користувачів	2.5 Мбайт	6 Мбайт
Максимальний об'єм ОЗП, що підтримується	12 Мбайт	4 Гбайт
Об'єм на жорсткому диску	5 Мбайт	9 Мбайт
Максимальний об'єм жорсткого диску	2 Гбайт	32 Тбайт
Можливість роботи без виділеного серверу	ТАК	НІ
Програми серверу	VAP	NML
Динамічне розподілення ресурсів	ТАК	НІ
Підтримка клієнтів OS/2	ТАК (не підтримує довгі імена системи HPFS)	ТАК
Підтримка клієнтів Mac	ТАК (емуляція)	За бажанням
Підтримка клієнтів NFS	НІ	За бажанням
Підтримка клієнтів OSI	НІ	За бажанням
Мережеві карти	8, 16 біт	8, 16, 32 біт

ОС NetWare 4.0

Версія 4.0 повністю сумісна з попередніми версіями, і користувач може навіть не помітити різниці.

Найбільш значною особливістю ОС NetWare 4.0 є система NDS (NetWare Directory Service — система обслуговування директорій в ОС NetWare), що являє собою ієрархічно організовану базу даних. Використана також нова система іменованих директорій, що дозволяє користувачам приєднуватись до серверів за одну операцію.

ОС NetWare 4.0 дозволяє одночасну роботу 5, 10, 20, 50, 100, 250, 500 та 1000 користувачів. При цьому доступ можливий до 54000 файлових серверів (раніше ця цифра дорівнювала 8). Новинками у версії 4.0 є:

- система кешування уявного читання;
- компресія даних і компоновка блоків даних, що дозволяє економити простір на жорсткому диску файлового сервера;
- поліпшена система захисту даних та ресурсів ЛОМ.

2.3.3. Мережеві ОС LAN Manager і LAN Server

Хоча ці мережеві операційні системи користуються меншою популярністю, ніж ОС NetWare, вони більше підходять для технології "клієнт-сервер". Більшість спеціалістів вважають, що майбутнє саме за цією технологією.

Обидві мережеві ОС LAN Manager (LM) та LAN Server (LS) працюють на базі OS/2. Для роботи ОС LAN Manager 2.2 потрібна OS/2 версії 1.21 чи більш пізня, у той

час як LAN Server 3.0 потребує OS/2 2.0. Робочі станції можуть працювати під керуванням DOS версії 3.3 або OS/2 версії 1.21.

В разі використанні OS/2 в якості операційної системи для управління файловим сервером ЛОМ з'являється можливість обслуговування запитів робочих станцій в багатозадачному середовищі, основаному на принципі розподілення пам'яті. Кожній задачі чи прикладній програмі виділяються визначені області пам'яті, котрі обслуговуються паралельно. При цьому прикладна програма складається з ***процесів***. Важливою перевагою є простота програмного управління комп'ютерами у середовищі OS/2, навіть якщо вони використовуються в якості файлових серверів.

Сама система OS/2 має наступні позитивні риси:

- нема обмежень пам'яті на рівні 640 Кбайт для прикладних програм;
- OS/2 версії 2.x представляє користувачу одночасний доступ до декількох сеансів DOS, кожний з яких може розпоряджатись об'ємом ОЗП 620 Кбайт;
- допускає можливість роботи у середовищі Microsoft Windows;
- проста інсталяція з використанням графічного інтерфейсу;
- використання віртуальної пам'яті;
- швидкий доступ до диску;
- високопродуктивна файлова система (HPFS — High Performance File System);
- підтримка національних мов (NLS — National Language Support);
- підтримка вдосконаленого механізму управління системою живлення (APM — Advanced Power Management);
- захист цілісності системи;
- швидка 32-хрозрядна архітектура;
- підтримка карт розширення PCMCIA.

Система OS/2 надає у розпорядження програміста ***іменовані канали*** (*named pipes*). Програміст може інтерпретувати ці канали як файли, але насправді іменовані канали містять повідомлення. Вони переміщуються від робочих станцій до файлового серверу. На сервері прикладна програма може виконувати їх обробку.

Інтерфейс командного рядка

Для введення мережевих команд необхідно запустити програму NET з деякими параметрами. Далі наведені найбільш важливі та найчастіше вживані варіанти команди NET з параметрами.

Команди для звичайної робочої станції

- **LOAD** — завантажує різноманітні мережеві протоколи;
- **NET CONTINUE** — продовжує призупинене обслуговування;
- **NET HELP** — виводить підказку для команди;
- **NET NAME** — привласнює ім'я комп'ютеру;
- **NET PAUSE** — припиняє зв'язок з мережею;
- **NET PRINT** — виводить чергу завдань друку або відсилає файл на друк;
- **NET START** — запускає мережу WORKSTATION;
- **NET USE** — виводить на екран розподілені ресурси чи привласнює літерні позначки дискам чи імена новим розподіленим ресурсам;
- **UNLOAD** — вивантажує мережевий протокол.

Додаткові команди для поліпшеної роботи робочої станції:

- **NET ACCESS** — переглядає дозвіл на допуск;
- **NET COPY** — копіює мережеві файли;
- **NET LOGON** — приєднує до мережі;
- **NET LOGOFF** — від'єднує від мережі;
- **NET PASSWORD** — змінює пароль;
- **NET START** — запускає робочу станцію чи визначає, з якими робочими станціями існують з'єднання;
- **NET TIME** — виконує синхронізацію годинників робочих станцій з годинником файлового серверу;
- **NET USE** — виводить на екран розподілені ресурси або присвоює літерні позначки дискам;
- **NET VIEW** — виводить на екран перелік серверів та їх ресурси;
- **NET WHO** — показує перелік користувачів, приєднаних до мережі.

В разі використання накопичувачів, адміністратор визначає, які ресурси файлового серверу можуть бути розподілені. З робочої станції виявити, чи є розподіленим увесь диск, чи тільки його директорія, неможливо.

Однією з суттєвих причин, з яких LAN Manager та LAN Server опинились менш популярними, ніж ОС NetWare, є великий об'єм дискового простору, необхідного для зберігання програмних та конфігураційних файлів LM та LS. Зазвичай, при роботі з NetWare, щоб мати доступ до файлового серверу, треба зберігати на робочій станції від двох до шести невеликих файлів. При роботі з LAN Manager та LAN Server для тих же цілей необхідно від 1 Мбайта до 3 Мбайт дискового простору на робочій станції.

Вимоги до об'єму пам'яті для програмного забезпечення робочих станцій під керуванням LAN Manager та LAN Server також багато вище, ніж для NetWare. Для LM та LS приблизний об'єм пам'яті під драйвери дорівнює 90 Кбайт, а робочі станції під керуванням NetWare потребують лише від 50 Кбайт до 60 Кбайт пам'яті під робочі драйвери.

Гарантія захисту даних

Файлові сервери та робочі станції організовані в областях. ***Область*** — це група серверів та робочих станцій з аналогічними вимогами до захисту даних. У великих мережах під управлінням LAN Manager та LAN Server є можливість упорядкування кількох областей. Області надають простий спосіб для контролю доступу користувачів до мережі та до мережевих ресурсів. Користувач може реєструватися у кількох областях, але приєднуватись до мережі він може лише в одній з цих областей.

У мережі під управлінням LAN Manager та LAN Server захист даних на рівні користувача складається з контролю при приєднанні до мережі та системи дозволів.

Кожний зареєстрований користувач має пароль. Для доступу до мережі у визначеній області користувач вказує своє ім'я та пароль. Мережевий адміністратор може обмежити доступ деяким користувачам у визначений час та день чи з конкретних робочих станцій.

Атрибути, які користувач може привласнити файлам та директоріям, наведені в табл. 2.4.

Таблиця 2.4

Дозвіл	Опис
Зміна атрибутів	Відмічає файли як "тільки для читання" чи "тільки для запису"
Зміна дозволів	Надає чи відміняє доступ іншим співробітникам
Створити	Створює файли та директорії
Знищити	Знищує файли та директорії (при наявності дозволу у користувача)
Виконати	Запускає на виконання файли з розширеннями .exe, .bat чи .com та не виконує читання чи копіювання цих файлів
Прочитати	Дозволяє читання чи копіювання файлів, запуск програм, зміну директорії та використання розширених атрибутів системи OS/2 для файлів
Записати	Дозволяє записати файл

Мережеві операційні системи LAN Manager та LAN Server дають можливість контролювати доступ до клавіатури та екрану файлового серверу. У спеціальному невиділеному режимі роботи файловий сервер дозволяє користувачам проглядати та керувати чергами друку, та не дозволяє змінювати псевдоніми зареєстрованих користувачів чи інші адміністративні дані. Для доступу до екранів з адміністративними даними користувач повинен вказати спеціальний пароль.

2.3.4. Мережева ОС Windows NT Advanced Server

Ще на початку 1993 року корпорація Microsoft випустила мережеву ОС Windows NT Advanced Server (NT), яка розширює характеристики та переваги ОС LAN Manager в кількох напрямках.

ОС Advanced Server є 32-х розрядною операційною системою, та на відміну від LAN Server, NT Advanced Server може працювати і на платформах MIPS R4000 фірми Intel чи Alpha фірми DEC. Передбачена робота з симетричними мультипроцесорними (з кількома центральними процесорами) комп'ютерами. Додаткові обчислювальні потужності на файловому сервері можуть бути використані для програм типу "клієнт-сервер".

В системі Advanced Server надається захист даних рівня С2. Це означає, що мережева ОС використовує захищену процедуру приєднання до ЛОМ, захист пам'яті, облік та контроль доступу (господар розподілених ресурсів має можливість визначити, хто на даний момент користується цими ресурсами). Рівня захисту С2 або вище потребують деякі промислові чи воєнні ЛОМ.

Якщо торкнутись надійності, то система Advanced Server використовує файлову систему, основану на транзакціях, яка дозволяє відмінити цілу серію зв'язаних модифікацій файлів, якщо ця серія не була завершена вдало. Вона також має засоби підтримки RAID п'ятого рівня (Redundant Array of Inexpensive Disks — надлишковий масив недорогих накопичувачів), можливість розпізнавання сигналів від джерела безперебійного живлення та програмне забезпечення для зберігання даних на стріммері.

До системи областей, що існувала в ОС LAN Server та LAN Manager, в ОС NT Advanced Server додане нове цікаве рішення, що називається ***областями довіри*** (*Trusted Domains*). Вона полягає в тому, що з однієї області можна "довірити" свої

файли іншій області, і тоді користувач іншої області зможе отримати до них доступ без додаткового приєднання до мережі в першій області.

Системним адміністраторам в управлінні системою NT Advanced Server допомагає утиліта Performance Monitor. Крім того, ця ОС підтримує протоколи керування мережею SNMP та NetView. Іншими утилітами, що входять до складу Advanced Server, є User Manager, Disk Administrator, Event Viewer та поліпшена панель управління (Control Panel).

Корисним є засіб Browse-Master. Кожний ПК з розподіленими ресурсами періодично повідомляє серверу Browse-Master список цих ресурсів. При натисненні на робочій станції кнопки **Browse** видається перелік доступних ресурсів, отриманих Browse-Master від комп'ютера. Цей метод зменшує трафік в ЛОМ, оскільки тепер робочим станціям і серверам не треба безперервно обмінюватись один з одним інформацією щодо ресурсів.

Система Advanced Server використовує протокол SMB на базі NetBIOS для обміну інформацією про перенаправлення файлів. Крім того, система Advanced Server сумісна з системами LAN Manager, LAN Server, родиною ОС Windows і навіть зі старою PC LAN Program. Також, у системі Advanced Server є засоби підтримки протоколів транспортного рівня таких як TCP/IP та IPX/SPX фірми Novell.

2.3.5. Відмінності між LM, NT та LS

Для налагодження системи LAN Server можна просто модифікувати файли `config.sys` та `ibmlan.ini`. В системі LAN Manager є засоби автоналагодження, які контролюють дії файлового серверу і автоматично здійснюють модифікації у файлах ініціалізації. Щоб ці зміни вступили в дію, необхідно час від часу вимикати та вмикати файловий сервер.

Іншою відмінною рисою системи LAN Manager є запам'ятовування мережевих зв'язків. Користувачі при черговому приєднанні до ЛОМ автоматично будуть мати ті ж зв'язки, що і в останньому сеансі. Цю властивість можна дозволити чи відмінити за допомогою опції `/PERSISTENT =` команди `NET USE`.

Архітектура протоколів за потребою (DPA — Demand Protocol Architecture) є характеристикою, яку корпорація Microsoft запозичила у 3Com. Ця компанія придбала у Microsoft ліцензію на систему LAN Manager та дещо її поліпшила, але компанія 3Com не змогла продовжити значного числа копій цієї поліпшеної версії LM. Коли вона вирішила облишити спроби перепродажу поліпшеної версії програмного забезпечення, корпорація Microsoft викупила ліцензію у 3Com.

В загальних рисах, система DPA дозволяє динамічно завантажувати та вивантажувати набір протоколів. DPA можна використовувати для епізодичного доступу до файлового серверу під управлінням NetWare. В цьому випадку система DPA тимчасово завантажить на робочій станції програмне забезпечення IPX та NETX мережевої ОС NetWare. По закінченні сеансу зв'язку з файловим сервером під управлінням NetWare система DPA вивільнить області пам'яті, що використовувались для розміщення IPX та NETX.

Однак у повсякденній діяльності користувачам мережі з файловими серверами, що працюють під керуванням LAN Manager чи NetWare, знадобиться не лише тимчасовий доступ до обох типів серверів. Тому, хоча система DPA і представляє технічне

рішення, його навряд чи можна вважати достатньо корисним та практичним. У системі LAN Manager є спеціальний засіб, що має назву NetWare Connectivity і дозволяє легко отримувати одночасний доступ як до файлового серверу під управлінням LAN Manager, так і до файлового серверу під управлінням NetWare.

Система LAN Manager також надає засіб для дистанційного адміністрування. Той, хто володіє привілеями адміністратора, може приймати чи виключати користувачів і виконувати інші адміністративні задачі з будь-якої робочої станції, що працює під керуванням OS/2, чи поліпшеної робочої станції під керуванням LAN Manager. Таким чином, для виконання адміністративних функцій немає необхідності обов'язково знаходитись біля файлового серверу.

Система LAN Manager має засіб встановлення паролю для обмеження доступу до розподіленого ресурсу чи пристрою. Цієї властивості немає у системі LAN Server.

Обидві системи, LAN Manager і LAN Server, використовують концепції захисту даних шляхом областей і паролів для доступу, та дещо різними шляхами. Тому якщо є потреба використати LAN Manager та LAN Server в одній і тій же мережі, то знадобиться організувати роздільні області для кожної з цих мережевих ОС. В одній з цих областей всі файлові сервери повинні будуть працювати під управлінням LAN Manager, а в іншій — під управлінням LAN Server.

При цьому, якщо робоча станція в області під контролем LAN Server намагається отримати доступ в область контролю LAN Manager, то попередньо необхідно впевнитись у тому, що вона була приєднана до мережі в області, що керується LAN Server. При роботі з робочими станціями з області LAN Manager такої проблеми не існує. В системі LAN Server для розподілених ресурсів можна використовувати скорочені імена, а в системі LAN Manager такої можливості немає. В цій системі треба використовувати повні імена розподілених ресурсів.

Припустимо, що в системі LAN Server до комп'ютера з ім'ям `PRODUCTION`, приєднаний розподілений принтер із скороченим ім'ям `PRINTER1`. Повне ім'я розподіленого принтера є `\\PRODUCTION\PRINTER1`. Таким чином, робоча станція, що працює під керуванням системи LAN Server, може розподіляти прилад `PRINTER1`, а робоча станція, що працює під керуванням системи LAN Manager, для доступу до цього приладу повинна буде використовувати його повне ім'я `\\PRODUCTION\PRINTER1`.

Системи LAN Manager та LAN Server добре працюють сумісно в ЛОМ Token Ring, та не сумісні ідеально в ЛОМ Ethernet. При роботі в ЛОМ Ethernet, можливо, знадобиться змінити конфігурації обох мережевих ОС. Справа в тому, що система LAN Server підтримує протокол DIX (Digital Intel Xerox) версії 2.0 і протокол IEEE 802.3, а LAN Manager не підтримує протокол DIX. Тому, щоб при роботі в ЛОМ Ethernet робочі станції могли використовувати обидва типи серверів, необхідно налагодити обидві системи, LM й LS, на використання протоколу IEEE 802.3.

Система LAN Manager — це 16-розрядне програмне забезпечення, у той час як Windows NT AS та LAN Server є 32-розрядними. Таким чином, теоретично останні два програмних продукти більш підходять для використання на сучасних комп'ютерах, з 32-розрядними процесорами. І справді, мережева ОС LAN Server має добру продуктивність, а вже ОС Windows NT використовує таку архітектуру операційної системи, яка ізолює мережеве програмне забезпечення від мережевого адаптера занадто багатьма прошарками проміжного програмного забезпечення.

В результаті, згідно даним щодо продуктивності, опублікованих в журналах *PC Week* та *PC Magazine*, ОС Windows NT AS повільніша за мережеві ОС NetWare або LAN Server. Крім того, Windows NT AS займає більше місця на диску та в пам'яті, ніж LAN Manager чи LAN Server.

2.3.6. Мережева ОС LANtastic

За популярністю та об'ємами продажу мережева ОС LANtastic компанії Artisoft на протязі довгого часу була лідером на ринку однорангових ЛОМ. Тому компанія Novell з Personal NetWare та корпорація Microsoft з Windows for Workgroups спробували проникнути в цю сферу ринку, створену фірмою Artisoft. Всі ці компанії пропонують високоякісне програмне забезпечення.

ОС LANtastic має ряд характеристик, що дозволяють їй прекрасно функціонувати, незважаючи на те, що вона є не найшвидшою з мережевих ОС для однорангових ЛОМ. Ця ОС має чудові можливості розподілення принтера. З додатковими апаратними засобами, що представляє компанія Artisoft, можлива навіть організація звукової електронної пошти в ЛОМ. ОС LANtastic потребує невеликого об'єму пам'яті та має засоби для розподілення накопичувачів типу CD-ROM.

Компанія Artisoft пропонує мережеві адаптери Ethernet, що особливо добре взаємодіють з ОС LANtastic. В ЛОМ, що працює під керуванням ОС LANtastic, є можливість використання комп'ютерів Macintosh. Ця система прекрасно сумісна і з Windows.

Технічна підтримка ОС LANtastic включає електронну дошку об'яв, до якої можна отримати доступ за допомогою модему, та телефонну консультацію компанії Artisoft у відділі підтримки користувачів.

Можливості ОС LANtastic

LANtastic є популярною мережевою ОС вже на протязі багатьох років. Версія 4.0, що була випущена ще у липні 1991 року, дає можливість працювати з прикладними програмами Windows. Працюючи в ній, можна керувати мережею, чергами друку, електронною поштою тощо простим натисненням кнопок миші.

Компанія Artisoft почала продаж версії 5.0 ОС LANtastic у березні 1993 року. В цій версії додані засоби організації роботи ОС LANtastic в ЛОМ NetWare на базі файлових серверів та можливості для розподілення в ЛОМ графічних і текстових даних прикладних програм пакету Windows.

У липні 1994 року компанія Artisoft випустила версію 6.0 ОС LANtastic. Нова версія має трохи вищу швидкодію, ніж попередні, і надає орієнтовані на використання у середовищі Windows утиліти для керування ресурсами ЛОМ. Є шлюз до цифрового текстового пейджеру — можна визвати до пейджеру співробітників, відсутніх на даний час за робочими станціями.

У версії 6.0 ОС LANtastic передбачені засоби для роботи з факсами в ЛОМ. Для цього необхідно встановити факс-модем на ПК, який є сервером, завантажити додатковий модуль LANtastic для обслуговування факсимільного апарату, і можна починати приймати і відправляти факси з усієї ЛОМ.

Нові версії ОС LANtastic надають більше засобів для управлінням сервером, включаючи контроль за використанням його ОЗП. Цей засіб дозволяє максимізувати

об'єм пам'яті, котру може використовувати сервер для прискорення обробки запитів файлів.

Попередні версії LANtastic були незалежними від мережевих адаптерів. Починаючи з версії 4.0, якщо необхідно використовувати мережевий адаптер, виготовлений іншою фірмою, необхідно додатково доплатити за програмні драйвери мережевого адаптера.

ОС LANtastic може працювати в ЛОМ, яка нараховує від двох до кількох сотень робочих станцій. При великій кількості робочих станцій продуктивність ЛОМ падає. Для подолання цього ефекту компанія Artisoft пропонує використовувати у таких випадках один чи кілька ПК у якості виділених файлових серверів.

У своєму складі ОС LANtastic містить багато корисних мережевих утиліт, що мають інтерфейс з користувачем через систему меню чи з командного рядка. Є також засоби для організації між користувачами ЛОМ діалогу за допомогою клавіатури, електронна пошта та засоби для виконання адміністративних функцій. Крім того, до складу ОС LANtastic входить резидентна програма `LANPUP`, яка дозволяє для доступу до системи меню мережевих утиліт використовувати "гарячі клавіші".

Система меню в ОС LANtastic

При запуску команди `NET` без параметрів автоматично активується система меню ОС LANtastic. Меню Main Functions (головні функції) в ОС LANtastic має наступні опції:

- мережеві накопичувачі й принтери;
- управління чергами друку;
- поштова служба;
- переговори з іншими користувачами;
- приєднання/вихід з системи;
- керування реєстрацією користувачів;
- огляд дій серверу.

Основні команди мережевої ОС LANtastic

- **`ATTACH`** — виділити всі розподілені диски на сервері;
- **`AUDIT`** — занести контрольну інформацію до log-файлу;
- **`CHANGEPW`** — змінити пароль;
- **`CHAT`** — почати набір повідомлень іншому користувачу;
- **`CLOCK`** — синхронізувати годинник робочої станції з годинником файлового серверу;
- **`COPY`** — копіювати файл з сервера на робочу станцію;
- **`DETACH`** — відмітити перепризначення мережевих накопичувачів;
- **`DIR`** — аналог DOS-команди `DIR`, але показує також мережеву інформацію та атрибути файлів;
- **`DISABLEA`** — відмінити псевдонім;
- **`EXPAND`** — визначити повний шлях до файлу;
- **`HELP`** — видати підказку;

- **INDIRECT** — дозволяє створити ***непрямий*** (*indirect*) *файл*, тобто файл, що має посилання на файл в іншій директорії; отже ця команда дозволяє отримати доступ до файлів в інших директоріях без зміни поточної директорії;
- **LOGIN** — почати мережевий сеанс;
- **LOGOUT** — закінчити роботу в ЛОМ;
- **LPT TIMEOUT** — задати довжину перерви для спулера друку ОС LANtastic, по вичерпанні якого файл вважається закінченим;
- **MAIL** — передати поштове повідомлення;
- **MESSAGE** — дозволити чи заборонити повідомлення про прихід чергового поштового повідомлення;
- **POSTBOX** — повідомити про отримані поштові повідомлення;
- **PRINT** — аналогічна DOS-команді PRINT;
- **QUEUE HALT** — зупинити мережевий спулер друку;
- **QUEUE PAUSE** — тимчасово призупинити мережевий спулер друку;
- **QUEUE RESTART** — поновити роботу спулера друку;
- **QUEUE STATUS** — показати чергу друку;
- **RECEIVE** — показати останнє мережеве повідомлення;
- **RUN** — запустити DOS-програму на вказаному сервері;
- **SEND** — надіслати повідомлення іншому користувачу ЛОМ;
- **SHOW** — повідомити про конфігурацію робочої станції в ЛОМ, до яких серверів вона приєднана, та показати перелік існуючих серверів;
- **SHUTDOWN** — задати зупинку чи перезавантаження файлового серверу;
- **UNUSE** — відмінити перепризначення накопичувачів на жорстких дисках та принтерів;
- **USE** — зробити перепризначення накопичувачів на жорстких дисках та принтерів в ЛОМ.

Продуктивність

За своєю продуктивністю ОС LANtastic займає середнє положення серед мережевих ОС. Для прискорення виконання запитів на обслуговування файлів в ній передбачена система кешування, яку називають LANcache. Ця система може працювати в розширеній, додатковій чи звичайній пам'яті. Користувач може задати об'єм пам'яті, що використовується для цих цілей. За замовченням береться вся розширена чи додаткова пам'ять.

Система LANcache має налагоджувальну тимчасову затримку при запису даних на диск (*delayed write function*). Така тимчасова затримка, коли сервер відповідає робочим станціям, що дані записані на диск, може привести до втрати частини даних при збоях у системі електроживлення файлового серверу.

Надійність роботи

В ОС LANtastic коректно здійснюється розподілення файлів, "захоплення" записів в файлах, взаємозв'язок з використанням NetBIOS та інші мережеві операції. Тому прикладні програми коректно працюють в цій ОС, особливо якщо вони є мережевими. При збоях в мережі електропостачання ОС LANtastic може розпізнавати сигнали джерела безперебійного живлення і автоматично зберігати дані.

Починаючи з версії 6.0, в систему додані засоби для автоматичного встановлення зв'язків робочих станцій з сервером після його перезавантаження.

Система захисту даних

Система захисту даних від несанкціонованого доступу в ОС LANtastic пропонує багато можливостей. Після реєстрації кожного з користувачів ЛОМ у вікні системи Windows можна призначити потрібні йому права та дозволи. Те ж саме можна зробити і за допомогою команди NET_MGR для адміністрування системи захисту даних. В обох випадках для виконання цих функцій необхідно мати привілеї системного адміністратора.

Команди NET_MGR потребують спеціального паролю — це перша лінія захисту в системі безпеки ОС LANtastic. Компанія Artisoft пропонує користувачам періодично змінювати свої паролі, і ОС може автоматично нагадувати про це через визначені програмно проміжки часу.

До числа засобів захисту даних також відносять надання прав доступу до визначених директорій на файловому сервері. За допомогою засобу контролю сервера можна відслідковувати доступ до файлів та встановлювати визначені дні тижня і години, коли користувачі з визначеними псевдонімами можуть отримувати доступ до файлів ЛОМ.

Частина II

Мережа Internet

Розділ 3

Всесвітня мережа Internet

"Internet — так багато в цьому слові
для серця кожного злилося..."

3.1. Основні поняття

Internet — це сукупність комп'ютерних мереж, що "обплутує" усю земну кулю, зв'язує урядові, військові, освітні, комерційні структури, а також окремих громадян. ***Internet*** — це об'єднання транснаціональних комп'ютерних мереж, що працюють за різними протоколами та з'єднують всі можливі типи комп'ютерів, фізично передають дані всіма доступними типами носіїв: по телефонному дроту, оптоволокну, через супутникові канали, радіомодеми тощо.

Зараз в країнах СНД діє ряд інформаційних мереж: Relcom, Freenet, Glasnet, UANet та ін. Однак ці мережі в основному використовують протоколи обміну даними більш низького рівня, ніж ті, на яких працює всесвітня інформаційна мережа Internet, і вони орієнтовані на низькочастотні канали зв'язку, які використовують, головним чином, для передачі електронної пошти.

З 1993 року Міжнародний науковий фонд (МНФ) — ISF (International Science Foundation) фінансує та керує кількома проектами по створенню в країнах колишнього СРСР наукових та освітніх комп'ютерних мереж з доступом до мережі Internet, якою сьогодні користуються більше 150 мільйонів абонентів у більш ніж 100 країнах, і яка об'єднує сотні тисяч окремих мереж. Постачальники інтерактивних послуг, подібні America Online, CompuServe, Prodigy Services, Microsoft Network "дають" замовникам прості у застосуванні інтерфейси та навігаційні інструменти для подорожей по Internet.

Телекомунікаційна програма МНФ включає в себе чотири крупні регіональні проекти. Один з них — це опорна наукова та освітня мережа у Києві з п'ятьма регіональними вузлами, що розташовані у районах зосередження інститутів, ВУЗів і з'єднані між собою радіорелейними каналами із супутниковим каналом в Internet.

Internet сьогодні — це мільйони комп'ютерів по всій земній кулі, набір стандартних протоколів, мережа з мереж. Це можливість спілкування мільйонів людей в різ-

них частинах світу, незалежно від їх суспільного стану і роду занять. Це засіб отримання та публікації будь-якої інформації, укладання договорів і проведення дозвілля.

Мережа Internet з'явилась біля двадцяти років тому внаслідок спроб об'єднати мережу Міністерства оборони США ARPAnet з різноманітними радіо- і супутниковими мережами. Мережа ARPAnet (Advanced Research Project Agency Net — мережа управління перспективними дослідженнями) була експериментальна, створена з метою забезпечення військових досліджень, зокрема, науково-дослідницьких робіт по створенню мереж, стійких до часткових відмов.

Мережа ARPAnet була розроблена і розгорнута у 1969 році компанією Bolt Beranek and Newman (BBN) на замовлення Агентства передових дослідницьких проектів (ARPA) Міністерства оборони США. Ця мережа об'єднала учбові заклади, військові організації та їх підлеглих. Вона була створена з метою допомогти дослідникам в обміні інформацією, а також (що стало однією з головних цілей) зв'язати мережі між собою з урахуванням відпрацювання методів підтримки зв'язку у випадку ядерного нападу. Засновники ARPAnet на початку дозволяли вченим тільки увійти у систему і запустити програму на віддаленому комп'ютері. Незабаром до цих можливостей добавились передача файлів, електронна пошта та списки розсилки, які забезпечили спілкування дослідників, що цікавилися однією і тією ж галуззю науки та техніки.

В моделі ARPAnet між комп'ютером-джерелом і комп'ютером-адресатом завжди існує зв'язок. Мережа була побудована так, щоб необхідність в інформації від комп'ютерів-клієнтів була мінімальною. Для пересилки повідомлень по мережі комп'ютер повинен був просто "покласти" дані в конверт, який називається "пакетом міжмережевого протоколу" (IP — Internet Protocol), і правильно "адресувати" такі пакети. Основний принцип полягав у тому, що кожний комп'ютер в мережі міг спілкуватися в якості вузла з будь-яким іншим комп'ютером.

Під тиском ринку творці Internet в США, Великобританії, Скандинавських країнах почали встановлювати IP-програмне забезпечення на різнотипні комп'ютери, що випускались різноманітними фірмами.

Але з розширенням ARPA розвивалися й інші мережі, і на далі виникла потреба в нових засобах зв'язку. Ще у 1973 році, за десять років до того, як відбулась революція персональних комп'ютерів, агентство ARPA (під своїм новим ім'ям DARPA) почало здійснювати програму Internetting Project. Її метою було визначити, як зв'язати мережі між собою з урахуванням того, що кожна з них використовує різні методи передачі інформації.

Серед найбільш важливих мереж слід відзначити Ethernet та NSFNET. Так, NSFNET, мережа національного наукового фонду (NSF) уряду США, наприкінці 80-х років об'єднувала п'ять суперкомп'ютерних центрів у провідних університетах, що дозволяло використовувати великі інформаційні ресурси для всіляких академічних досліджень. Але комунікаційна проблема не була вирішена остаточно, тому що, з причини бюрократичних та кадрових проблем, можливість використовувати для передачі даних мережу ARPAnet успіху не мала.

Тоді фонд NSF у 1985 році вирішив створити свою власну мережу на базі IP-технології мережі ARPAnet. Для цього були створені регіональні мережі, тобто у кожному регіоні США університет з'єднувався із своїми найближчими сусідами. Такий ланцюжок приєднувався до суперкомп'ютера в одній точці, і таким чином всі супер-

комп'ютерні центри (вузли) були з'єднані разом, а будь-який комп'ютер міг спілкуватись із всіма іншими шляхом передачі повідомлень через своїх сусідів (рис. 3.1).

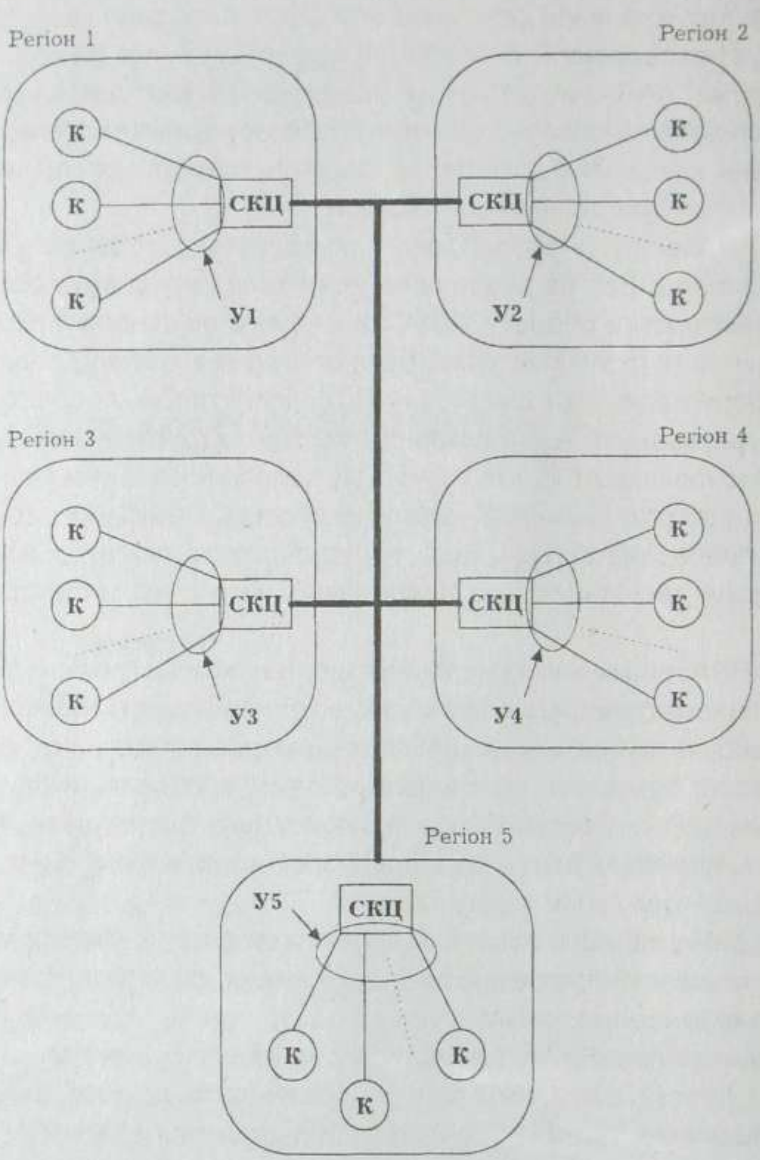


Рис. 3.1. Архітектура Internet у період її заснування NSF у 1985 році: К — комп'ютер; СКЦ — суперкомп'ютерний центр; У — університет

Колективне використання суперкомп'ютерів дозволило підключеним до центрів спільнотам використовувати сумісно велику кількість інших допоміжних ресурсів, що не відносились до цих центрів. З часом інтенсивність обміну даними між університетами зросла до такої межі, що комп'ютери, які керували мережею, почали не витримувати такого навантаження.

В 1987 році компанія Merit Network Inc., що координувала обчислювальну мережу учбових закладів штату Мічиган, отримала контракт на керівництво і розвиток мережі NSF. Застаріла конфігурація була доповнена більш швидкісними телефонними лініями (швидкість передачі зросла в 20 разів відносно початкової швидкості передачі даних — 56000 біт/с, що відповідає можливості передачі близько двох повних ар-

кушів друкованого тексту за секунду), і більш швидкодіючими комп'ютерами. Процес нарощування потужності мережі продовжується і сьогодні, що дозволяє забезпечувати доступ до мережі практично всім бажаючим.

У 1983 році, Агентство зв'язку Міністерства оборони США прийняло рішення про використання протоколів TCP/IP на всіх вузлових машинах ARPAnet. Таким чином, було встановлено стандарт, завдяки якому могла розвиватись мережа Internet. З цього моменту стало можливим добавляти шлюзи і приєднувати до Internet нові мережі, у той час, як початкове ядро залишалось незмінним. Більшість аналітиків вважають, що дійсна дата виникнення Internet — 1983 рік, коли ARPAnet була розділена на мережу MILNET, яка призначалась для використання у військових цілях, і ARPAnet, що орієнтована на продовження досліджень у області мереж.

Між тим, першою окремою мережею, яку агентство DARPA дозволило підключити до ARPAnet ще в 1980 році, стала CSNET, що об'єднала комп'ютери наукових установ декількох штатів. В 1989 році вона злилась із мережею BITNET. Сама ARPAnet припинила своє існування у червні 1990 року, а її функції поступово перейшли до більш розгалуженої структури Internet.

Чи керує хто-небудь мережею Internet, чи вона існує сама по собі? Якщо накреслити найпримітивнішу схему, то в основі Internet буде лежати система магістральних мереж, що називаються ***опорними***. В США найвидатнішою такою мережею є ANSnet (Advanced Network & Services Network), на основі якої надаються послуги NSFNET.

Асоціація Commercial Internet Exchange об'єднує могутню та зростаючу опорну мережу комерційних постачальників послуг мереж. Послуги середнього рівня за своєю природою є регіональними і об'єднують один або декілька штатів в високошвидкісну опорну мережу. На місцевому рівні мережі окремих організацій підключені у регіональні мережі, які в свою чергу надають їм доступ до потоку трафіка в опорній мережі. В Європі в якості прикладу можна назвати мережу EBONE, пан'європейську опорну IP-мережу, а також мережі NORDUnet, EUROPANet та EUnet, що забезпечують з'єднання мереж в усьому світі.

Кожна з мереж відповідає за трафік, який циркулює по ньому і має змогу маршрутизувати його на власний розсуд. Якщо, наприклад, інформацією бажають обмінятися два комп'ютери, що розташовані у одному університеті, то трафік не слід випускати за кордони локальної мережі. Такий же принцип застосований до регіональних мереж. Кожна мережа відповідає за з'єднання з мережею більш високого рівня.

В Internet немає ані президента, ані головного інженера, але вони є у мереж, що входять до складу Internet. Основні напрямки розвитку визначає "Спільнота Internet" (ISOC — Internet Society). ISOC — це організація на громадських засадах, яка має за мету сприяти глобальному інформаційному обміну через Internet. ISOC призначає раду старійшин, яка відповідає за технічне керівництво і орієнтацію Internet. Рада старійшин по архітектурі Internet (IAB — Internet Architecture Board) являє собою групу запрошених осіб, які добровільно виявили бажання прийняти участь в її роботі. Рада IAB регулярно збирається, аби затверджувати стандарти і розподіляти ресурси (наприклад, адреси).

Internet працює завдяки наявності стандартних засобів взаємодії комп'ютерів і прикладних програм. Наявність таких стандартів дозволяє без проблем зв'язувати між собою комп'ютери виробництва різних платформ і фірм. Рада IAB несе відпові-

дальність за ці стандарти, а також слідкує за різного роду номерами, що повинні залишатись унікальними (наприклад, адреса комп'ютера).

Інженерна комісія Internet (IETF — Internet Engineering Task Force) — це ще один громадський орган, де користувачі Internet можуть висловити свої погляди відносно того, як повинна функціонувати мережа. Комісія IETF регулярно збирається для вирішення поточних технічних та організаційних питань і може формувати різні робочі групи з добровольців для вирішення виникаючих важливих проблем.

Можна відзначити ще дві структури: InterNIC — державна служба реєстрації в Internet та WWWC (WWW Consortium) — комітет з питань стандартизації в Internet.

Оплата за послуги Internet здійснюється наступним чином. Будь-яка мережа, що включена в Internet, сама відповідає за своє фінансування і сама може встановлювати власні адміністративні процедури. Чи зростає Internet зі швидкістю 15% в місяць (за одними оцінками) чи 80% в рік (за іншими), це вдається пояснити тільки за умови, коли мережі, що підключаються до Internet залишаються під керівництвом власних адміністраторів мереж.

В зв'язку з тим, що єдиного постачальника мережевих послуг не існує, організації самі фінансують свої мережі, кожна з яких вважається частиною загальної мережі Internet. Також існують приватні опорні мережі, які часто на комерційній основі пропонують доступ до Internet компаніям та окремим особам.

Багато регіональних мереж з самого початку були створені за кошти Національного наукового фонду США з розрахунку на те, що з часом вони перейдуть на самофінансування. Внаслідок виникло безліч мереж, принципи фінансування, управління та "правила поведінки" яких значно різняться. Деякі мережі, наприклад, дозволяють передачу комерційного трафіка, у той час як інші його забороняють. Таким чином, національний науковий фонд сплачує за NSFNET, NASA — за NASA Science Internet і т.ін. Представники мереж збираються і вирішують, як з'єднувати та фінансувати ці взаємні з'єднання. Коледж або корпорація сплачують за підключення до регіональної мережі, яка в свою чергу сплачує за доступ до Internet постачальнику на рівні держави.

Таким чином, Internet — це не мережа, а сукупність мереж, яка є аналогом світової телефонної мережі, що також являє собою об'єднання мереж (наприклад, Pacific Bell, MCI, British Telecom, Telefonos de Mexico та ін). Різноманітні телефонні компанії разом мають змогу обговорити загальні проблеми, але вирішення цих проблем — це справа кожної компанії окремо.

Такий же стан справ існує в Internet. Кожна мережа має свій власний центр керування (ЦКМ). ЦКМ взаємодіють і знають як вирішувати свої проблеми, а кожна організація, що бажає працювати в Internet, повинна налаштувати контакт зі своєю мережею та ЦКМ. Таким чином, однією з властивостей Internet є відсутність централізованого управління мережею.

Завдяки наявності великої кількості конкуруючих між собою компаній-провайдерів, користувач має безпрецедентну можливість вибирати між якістю, кількістю та цінами всього спектру послуг, що надаються. Internet не має недоліків та переваг, що виникають при централізованому керуванні. В Internet не існує єдиного пункту підписки або реєстрації. Замість цього користувач контактує з постачальником послуг, який надає доступ до мережі завдяки місцевому комп'ютеру. Фактично

Internet складається з користувачів, які можуть з'єднуватися між собою завдяки постачальникам послуг.

Наслідки такої децентралізації з точки зору доступності ресурсів мереж дуже вагомі. Те, що користувач зможе знайти у Internet, залежить від рішень тисяч системних адміністраторів всього світу. Не існує якої-небудь однієї компанії, що приймає рішення про загальну будову системи.

На протязі довгого часу за надання мережевих інформаційних послуг у масштабі всієї Internet ніхто не відповідав. Існувала велика кількість центрів мережевої інформації (NIC — Network Information Centre), кожний з яких надавав свою інформацію по мережі клієнтам. Тепер створено центр мережевої інформації InterNIC (NIC Internet), що фінансується Національним науковим центром. NIC Internet — офіційний постачальник інформації по Internet-мережі, послугами якої користуються всі користувачі Internet. З Internet співробітничає безліч американських та закордонних NIC (наприклад, APNIC — азіатсько-тихоокеанський центр, AgNIC — центр сільського господарства та ін.). InterNIC складається з трьох відділень (служб): реєстрації, баз даних та інформації. Кожним із відділень керує окрема компанія зі своїм власним уставом, але в сукупності вони являють собою єдину ефективну структуру.

3.2. Загальні принципи роботи Internet

Загальні принципи роботи Internet базуються на багаторівневому режимі роботи будь-якої сучасної мережі: рівень апаратури та лінії зв'язку (для передачі повідомлень), рівень базового програмного забезпечення (ПЗ) для керування роботою апаратури, рівень стандартного ПЗ для розширення базового ПЗ додатковими можливостями та ін.

Найнижчим рівнем роботи мережі є рівень апаратури та ліній зв'язку. Базу Internet створюють вузлові комп'ютери або просто вузли, що об'єднані каналами зв'язку самого різноманітного типу — від телефонних до супутникових. Як правило, це могутні UNIX-станції, які здатні забезпечувати одночасну роботу у мережі багатьох тисяч користувачів. Такі вузлові центри зараз називають ***Internet-провайдерами***, тобто постачальниками послуг.

За допомогою ліній зв'язку забезпечується доставка даних з одного пункту в інший. Це здійснюється за рахунок важливого принципу комутації пакетів, а не за рахунок комутації каналів (як в телефонній мережі, коли користувачу персонально виділяється деяка частина цієї мережі). Різноманітні ділянки Internet зв'язуються за допомогою системи комп'ютерів, що називаються маршрутизаторами, які з'єднують між собою мережі. Ними можуть бути мережі Ethernet, мережі з маркерним доступом, телефонні лінії (рис. 3.2).

Маршрутизатори приймають рішення про те, куди направляти дані ("пакети"), на основі адреси пакета. За адресу пакета відповідає міжмережевий протокол IP (Internet Protocol), на основі певних правил, які регламентують порядок роботи Internet.

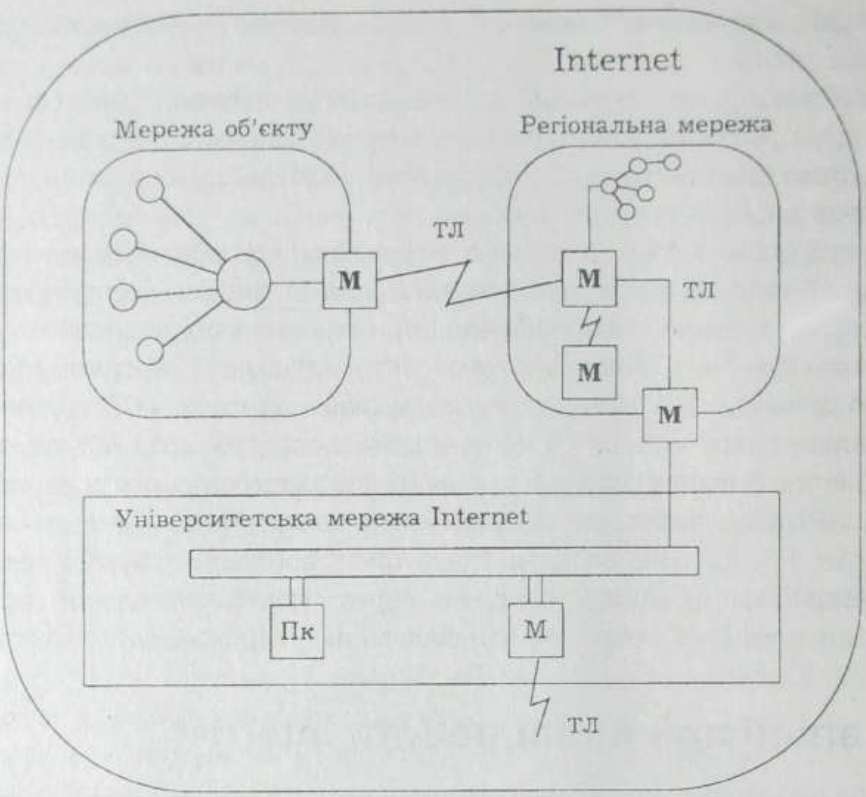


Рис. 3.2. Апаратні засоби Internet:
М — маршрутизатор; ТЛ — телефонна лінія; ПК — персональний комп'ютер

3.2.1. Стандартні протоколи Internet

Протокол — це просто набір домовленостей, який визначає обмін даними між різними програмами. Протоколи задають способи передачі повідомлень та обробки помилок у мережі, а також дозволяють розробляти стандарти, що не зв'язані з конкретною апаратною платформою. Всі параметри — від швидкості передачі до методів адресації при транспортуванні окремих повідомлень — задаються протоколами, що використовуються у даній конкретній мережі. В Internet базовою є родина протоколів TCP/IP (Transmission Control Protocol/Internet Protocol). Протокол IP відповідає за адресацію мережевих вузлів, а TCP забезпечує доставку повідомлень за потрібною адресою. Ці потужні протоколи були запропоновані в 1974 році Робертом Кеном, одним з основних розробників ARPAnet, і вченим у галузі комп'ютерних технологій Вінтоном Серфом. У теперішній час Кен є президентом американської Корпорації національних дослідницьких ініціатив (CNRI), а Серф — президентом Internet Society та віце-президентом CNRI.

➢ Більш докладно протоколи Internet розглядаються в наступному розділі.

Протокол IP

Протокол IP, відповідаючи за адресацію, гарантує, що маршрутизатор знає, що саме робити з даними користувача. За аналогією з поштою можна стверджувати, що

IP виконує функції конверта. Адреса на конверті (Internet-адреса) має дати для мережі відомості, куди саме доставити пакет даних.

На кожному рівні ієрархії Internet мережа, що входить до її складу, сама відповідає за порядок усередині себе. З точки зору адресації це означає, що будь-яка організація, що підключена до мережі, веде базу даних своїх мережевих комп'ютерів. Унікальні номери, які застосовуються для ідентифікації комп'ютерів в Internet, як було сказано, називаються ***IP-адресами***.

IP-адреса складається з чотирьох номерів (кожний розміром не більше 256 в десятковому запису), що відокремлені одна від одної крапками. Наприклад, **192.33.33.22** — це IP-адреса, так само, як і **13.40.11.1**. Крайнє зліва число означає велику мережу, ті, що стоять правіше, вказують на більш дрібні ділянки усередині цієї великої мережі, і так доти, доки ми не попадемо на конкретний комп'ютер, тобто права частина адреси повідомляє мережі, який саме комп'ютер повинен мати зв'язок з мережами (кожний комп'ютер має свою унікальну адресу).

З технічних причин (апаратні обмеження) інформація, що надсилається по IP-мережам, розподіляється на порції, які називають ***пакетами***. В одному пакеті, як правило, надсилається не більше 1500 символів інформації. Це дозволяє кожному сподіватися на своєчасне обслуговування і водночас не дає можливості одному користувачеві монополізувати мережу. Однією з переваг Internet є те, що для роботи на базовому рівні достатньо лише протоколу IP.

Протокол TCP

Протокол керування передачею (TCP) використовується для вирішення таких проблем, як пересилка по мережі інформації, об'єм якої перевищує 1500 символів, за рахунок розподілу інформації на порції; щоб знайти та ліквідувати помилки при пересиланні інформації (втрата або пошкодження пакету); для відновлювання порядку послідовності доставки пакетів і т.ін.

Таким чином, інформація, яка передається користувачу по мережі, TCP розбиває на порції. Кожна порція послідовно нумерується. Для передачі цього порядкового номеру мережі протокол має свій власний "конверт". Дані, що розбиті на порції, розміщуються у конвертах TCP, які, в свою чергу, розміщуються в конверт IP і передаються у мережу (рис. 3.3).

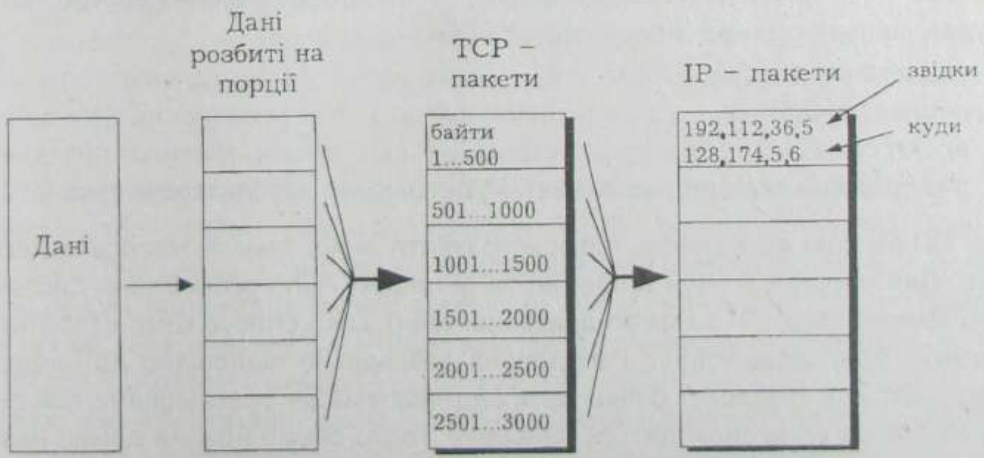


Рис. 3.3. Схема подрібнення і передачі даних в Internet

Протокол TCP, приймає інформацію, ПЗ збирає конверти, вибирає з них дані та розкладає їх у певному порядку. Якщо є втрати будь-якого конверта, програма запрошує дозволу відправника передати їх ще раз. Крім того, TCP при відправленні даних рахує контрольну суму для інформації, що передається, а коли приймає її, — перевіряє знову. Якщо ці суми не збігаються, тобто при передачі виникла помилка, TCP, що приймає інформацію, відкидає цей пакет і запитує повторну передачу.

Слід зауважити, що TCP/IP — не єдиний протокол, який може використовуватись для об'єднання різноманітних мереж. Мережа Internet, в дійсності, перетворилась в багатопротокольну мережу, що інтегрує різні стандарти. Основні серед них — стандарти взаємодії відкритих систем (OSI). Запропоновані Міжнародною організацією по стандартизації (ISO), протоколи OSI отримали велике розповсюдження у Європі, де вплив TCP/IP не був таким великим, як у США. Системи, які засновані на інших протоколах, також підключаються до Internet через шлюзи. Наприклад, BITNET — це мережа, яка використовує для передачі даних свої власні стандарти, але частково вона доступна через шлюзи Internet.

Протокол UDP

Існує ще один стандартний протокол UDP (User Datagram Protocol — протокол дейтаграм користувача), що використовується замість TCP. Цей протокол простіший та дешевший, ніж TCP, тому, що не турбується про втрачені пакети, про розміщення даних у необхідному порядку та про інші тонкощі. UDP використовується у тих програмах, що надсилають лише короткі повідомлення і можуть повторити передачу даних, якщо відповідь затримується. Робота з UDP по передачі аналогічна TCP, тобто формується один пакет UDP, котрий вкладається в IP-пакет і відправляється. Якщо пакет загубився, надсилають ще один запит.

3.2.2. Система доменних імен (DNS)

Зручність користування Internet характеризується багатьма факторами, одним з яких є можливість звертатись до комп'ютерів не за IP-адресою, а за іменем. Для цього необхідно налагодити прикладні програми на конкретну задачу.

Прикладні програми (ПП) — це частина ПЗ, що надбудовується над сервісом TCP та UDP, і які пропонують користувачеві засоби для вирішення конкретної задачі.

Internet має наступні основні категорії ПП:

- віддалений доступ,
- пересилка файлів,
- електронна пошта,
- ряд інших не стандартизованих ПП, що широко використовуються.

Всі ПП Internet дозволяють використовувати імена замість числових адрес комп'ютерів. Для реєстрації імен в Internet та їх пошуку була розроблена система DNS (Domain Name System — система доменних імен). DNS описує комп'ютери та організації, в яких вони влаштовані, і збудована дзеркально відповідно до цифрової IP-адресації. Якщо в IP-адресі більш загальна інформація розташована ліворуч, то в доменних іменах вона знаходиться праворуч, тобто самий правий домен називається ***доменом верхнього рівня***.

Система доменних імен являє собою метод призначення імен шляхом покладення відповідальності за підмножину імен на різні групи користувачів. Кожний рівень цієї системи називається ***доменом***. Імена доменів відділяються один від одного крапками, наприклад, `UX.CSO.UIUC.edu`, `nic.ddn.mil` чи `yoyodyne.com`. В імені може бути будь-яка кількість доменів, але більше п'яти зустрічається рідко. Кожний наступний домен в імені (якщо дивитися зліва направо) більший, ніж попередній. Наприклад, в імені `UX.CSO.UIUC.edu` домени мають наступні значення:

- **`UX`** — ім'я реального комп'ютера з IP-адресою,
- **`CSO`** — ім'я групи (відділу), в якій знаходиться комп'ютер і яка створила ім'я цього комп'ютера як UX;
- **`UIUC`** — відділ університету, куди входить група CSO;
- **`edu`** — учбові заклади національної групи, до складу якої входить UIUC.

Таким чином, домен `edu` включає у себе всі комп'ютери учбових закладів США, домен `UIUC.edu` — всі комп'ютери і т. ін. (див. рис. 1.42). Кожна група має можливість створювати та використовувати всі імена, що знаходяться під її контролем, за допомогою відповідної згоди та правилами, що забезпечують унікальність імені.

При створенні доменної системи були впроваджені наступні первісні імена доменів верхнього рівня:

- **`com`** — комерційні домени, тобто ці адреси належать фірмі або компанії;
- **`edu`** — освітні організації (наприклад, ім'я `wisc.edu`, відповідає Університету штату Вісконсин);
- **`gov`** — домен верхнього рівня для комп'ютерів урядових організацій;
- **`mil`** — відноситься до військового відомства;
- **`net`** — відноситься до організацій, що керують мережами;
- **`org`** — як правило, застосовується для приватних компаній, які не відносяться до згаданих вище категорій (так, Internet Society використовує домен `isoc.org`).

Різні країни мають власні дволітерні домени верхнього рівня. Наприклад, доменне ім'я Німеччини — `.de`, Швейцарії — `.ch`, Італії — `.it`, Канади — `.ca`, України — `.ua`, Росії — `.ru`. США також мають свій домен — `.us`. Наприклад, комп'ютер на території Канади може мати ім'я `hockey.quelph.ca`. Загальна кількість кодів країн — 300, комп'ютерні мережі існують приблизно у 150 з них.

З такою адресною інформацією людині працювати легше, тому що доменні імена мають постійну структуру, спираючись на яку, можна впізнати, до чого вона належить. Користувач набирає ім'я, а маршрутизатори, котрі обробляють мережевий потік даних, підставляють замість нього відповідні цифри IP-адреси.

Щоб звернутись до конкретного користувача за адресою, необхідно після його прізвища (або імені) додати символ "@" (комерційне at), наприклад, `vant@unicyb.kiev.ua`, де `vant` — прізвище (ім'я) користувача; `unicyb` — уточнюючий домен (наприклад, назва організації); `kiev` — місто; `ua` — літерний код країни, в якій живе власник даної адреси (у даному випадку —України).

Ще два приклади: `wvs@hotline.kiev.ua`, `ys@hotline.kiev.ua`.

Якщо існують доменні імена, то може пора відкинути IP-адреси? Виявляється, що ні, оскільки імена дозволяють комп'ютерам в мережі лише отримувати інформацію про адреси. Коли вказується конкретний комп'ютер за допомогою доменного іме-

ні, наприклад vant@unicyb.kiev.ua, то сервер імен, що відповідає за визначену область адрес (доменів), перекладе це ім'я у IP-адресу. Залежно від місця знаходження комп'ютера та географічної відстані від нього, такі запитання можуть пройти через декілька серверів імен, перш ніж буде отриманий кінцевий адрес.

Привабливість цієї системи складається з двох обставин: по-перше, не потрібно обробляти жодного із запитів/відповідей, тому що це робиться автоматично; по-друге, у порівнянні з централізованим списком адрес (до речі, саме так колись працювала мережа Internet), система DNS дозволяє мережі зростати із значно меншими організаційними зусиллями. Завдяки тому, що імен серверів дуже багато, при порушенні одного з них автоматично буде зроблена спроба з'єднатися з місцем призначення іншим шляхом — можливо навіть не найкоротшим.

Для пошуку будь-якої адреси в Internet можливо використовувати:

- NETfind — програма пошуку у вигляді каталогу;
- WHOIS — списки з інформацією про комп'ютери Internet, користувачів та мережі;
- Finger — фахова команда пошуку.

3.3. Служби та послуги в Internet

Мережа Internet створювалась з певною метою: надати експериментальну платформу для розробки комп'ютерних мереж, що змогли б продовжувати функціонувати в умовах різноманітних катастроф. Надмір варіантів маршрутизації, завдяки якому повідомлення користувача буде передано адресату, бере своє походження від наступної тези: аби мережі, пошкоджені на деяких ділянках, зберігали робочий стан, вони повинні мати здатність до "самовідновлення". Ось звідки Internet має "вміння" пересилати інформаційні пакети будь-яким шляхом, який потрібен для забезпечення їх практичної доставки.

Можна сказати, що в Internet існує безліч інформаційних служб та послуг, які за часовим способом отримання інформації класифікуються як:

- ***інтерактивні*** — відповідь на запит здійснюється негайно, тобто користувачі потребують негайної реакції на отриману інформацію (аналог телефону);
- ***прямі*** — відповідь на запит повертається негайно, але від того, хто отримує інформацію, не вимагається негайної реакції (аналог факсу);
- ***відкладеного читання*** — запит і відповідь дуже сильно розподілені у часі; цей вид сервісу найбільш поширений і універсальний, тому що не накладає великих вимог на ресурси комп'ютера та лінії зв'язку (аналог посилки).

Існують наступні основні служби Internet:

- **E-mail** — електронна пошта;
- **USENET** — мережеві новини чи телеконференції;
- **Maillist** — списки розсилки;
- **FTP** — передача файлів, тобто доступ до множини віддалених файлових бібліотек і передача файлів;
- **Archie** —пошукова служба, котра полегшує роботу з серверами anonymous ftp, та організовує пошук файлів на таких серверах;

- **WWW** — мережева інформаційна служба World Wide Web ("всесвітня павутина"), тобто засіб роботи з інформацією на багатьох серверах;
- **Hyper-G** — засіб роботи з інформацією на "своєму" сервері;
- **Gopher** — пошукова служба або розподілена система експорту структурованої інформації;
- **Veronica** — засіб пошуку у Gopher-просторі;
- **WAIS** — пошукова служба, або інформаційна система широкого профілю;
- **IRC (TALK)** — система реалізації переговорів абонентів мережі у реальному часі;
- **MUD** — гра багатьох користувачів;
- **MOO** — об'єктно-орієнтована гра багатьох користувачів;
- **Finger (Whois)** — довідкова служба для отримання інформації про комп'ютери користувачів (локальних чи віддалених);
- **Telnet** — система термінального доступу до віддалених комп'ютерів;
- Шлюзи:
  - **TELEX** — в телекс;
  - **TELETYPE** — в телетайп;
  - **PAGER** — в пейджерну систему;
- Комплекс серверів, котрі реалізують аудіо- та відеоконференції:
  - **Iphone** — комп'ютерний телефон;
  - **RealAudio** — система передачі звуку клієнту з серверу;
  - **CU-SeeMe** — система передачі відеозображення;
  - **Multicast** — система передачі мультимедійної інформації.

Слід, однак, пам'ятати, що для отримання повного набору послуг необхідно конкретний комп'ютер з'єднати з мережею по протоколу TCP/IP, що не завжди можливо.

Розглянемо перелічені служби більш детально.

3.3.1. Електронна пошта (E-mail)

Електронна пошта є найбільш популярною та простою послугою і для багатьох користувачів — найбільш корисною. Більшість користувачів Internet мають справу виключно з електронною поштою і все одно вважають мережу безцінним ресурсом, тому що можна відправляти повідомлення одному або декільком абонентам, пересилати текстові файли, отримувати інформацію через шлюз у BITNET та ін. Завдяки пошті можна співробітничати у різноманітних проектах, пересилати повідомлення своїм колегам і т.ін.

Повідомлення електронної пошти можуть здаватися досить складними. Насправді ж, вони складаються з двох основних компонентів: заголовку з датою, часом, поштовими реквізитами, і власне повідомлення. Крім відправлення повідомлень електронної пошти, можна налагодити зв'язок з великою кількістю академічних та професійних користувачів в будь-якій області знань або діяльності.

На початку існування ARPAnet електронна пошта вважалась незначним додатком до тих можливостей, які надавала мережа. Ніхто не чекав на той величезний потік інформації, що з'явився в період, коли науковці почали робити обмін своїми ідеями з територіально віддаленими колегами. Сьогодні наявність електронної пошти

сприймається як щось звичайне — як для невеличких компаній з локальними мережами, так і для гігантських корпорацій, які об'єднують свої відділення, розкидані по всьому світу. Не менш стрімким було зростання електронної пошти у комерційних мережах, абоненти яких користуються тільки цією послугою.

Електронна пошта забезпечує наступні основні функції:

- **Підпис**. Функція підпису надає можливість автоматично добавляти файл або текстову мітку кожному повідомленню, що відправляється.
- **Адресні книги**. За допомогою адресних книг можна зберігати як індивідуальні адреси Internet, так і групові, тобто "списки розсилання".
- **Додаток**. Додаток оформлюється як частина поштового повідомлення. Найпопулярнішим форматом додатку вважається формат MIME (Multipurpose Internet Mail Extension — багатоцільове поштове розширення Internet). Поштовий пакет, який підтримує MIME, перетворює двійковий файл в текст, відзначаючи маркерами його початок та кінець. Застосовують ще два формати додатку, аналогічні MIME. Формат BinHex популярний в середовищі Macintosh, але погано підтримується системами Windows та Unix. Багато поштових пакетів сумісні також із форматами uencode/uudecode, які поширені в середовищі Unix.
- **Inbox/Outbox**. Звичайно, головне в електронній пошті — це поштові скриньки для листів, що входять (*inbox*), та що виходять (*outbox*). Але вже сьогодні є можливість створювати декілька поштових скриньок або папок (*folders*), як вони називаються у більшості поштових пакетів. Це дозволяє, наприклад, зберігати в одній скриньці всі повідомлення, що пов'язані з роботою, а в іншій — особисту переписку.
- **Фільтрація/маршрутизація**. Завдяки засобам маршрутизації або фільтрації можна задавати правила, по яким програма повинна обробляти вхідні повідомлення. Діапазон дій залежить від окремого пакета, хоч більшість програм, що оснащенні цією функцію, дозволяють сортувати повідомлення по папках.
- **Універсальна поштова скринька** дозволяє збирати в одному місці пошту не тільки з Internet, але також і з інших джерел.

В Internet з'єднання через електронну пошту здійснюється за допомогою спеціального ftpmail-сервера, який самостійно приєднується до віддаленої системи, входить у неї та пересилає потрібні файли на систему користувача у відповідь на набір команд, котрі вказані у його запиті.

3.3.2. Обмін новинами (USENET)

USENET— це сама велика електронна дошка об'яв (BBS — Bulletin Board System). Електронна дошка об'яв завдяки групам новин об'єднує багатьох користувачів Internet, а також інших електронних дошок об'яв в різних дискусійних групах та спеціалізованих конференціях за інтересами. Кожний день сотні повідомлень, підготовані тисячами користувачів, з'являються більш ніж в 7 тисячах груп новин.

USENET користується своєю власною термінологією, наприклад, "передача новин" — замість конференції; "стаття" — замість повідомлення. В основі назви груп

новин лежать чіткі правила. Назва формулюється, починаючи з більшої категорії, і закінчується меншою, зліва направо.

Існують наступні тематичні групи новин:

- **alt** — альтернативні групи;
- **bionet** — біологічна направленість;
- **biz** — підприємницька діяльність;
- **comp** — комп'ютерні групи;
- **k12** — загальноосвітні;
- **misc** — збірники, альманахи;
- **news** — новини і т.п.;
- **rec** — хобі, відпочинок, мистецтво;
- **sci** — наука;
- **soc** — питання соціології;
- **talk** — обмін думками.

3.3.3. Списки розсилки (Maillist)

Maillist — це практично єдина служба, яка не має власного протоколу і програми. Вона працює через електронну пошту. Головна ідея роботи Maillist полягає в тому, що існує багато адрес електронної пошти, які є загальними для багатьох людей-передплатників цього списку розсилки. Коли відправляється лист на цю адресу його отримають всі, хто є передплатниками цього листа через свою поштову скриньку.

3.3.4. FTP

FTP (File Transfer Protocol) — протокол передачі файлів з одного комп'ютера на інший. Доступ до матеріалів, які переведені в розряд загальнодоступних, здійснюється через процедуру під назвою "анонімний FTP" (*anonymous ftp*), що дозволяє користувачу реєструватися на віддалених комп'ютерах і використовувати джерела інформації у тих каталогах, які адміністратори систем відкрили для загального доступу. Використання FTP перетворює пошук програм в Internet в складне завдання. Замість того, щоб запросити каталог всіх файлів із єдиної бібліотеки, як це здійснюється в CompuServe або GEnie, треба аналізувати тисячі комп'ютерних архівів, що містять програми та текстові файли.

У всьому світі існує безліч FTP-серверів (серверів, з яких можна "скачувати" інформацію завдяки FTP). Таким чином, FTP — це метод пересилки файлів між комп'ютерами. Одна з колосальних переваг протоколів сімейства TCP/IP полягає в тому, що вони постачають єдиний набір інструментів комп'ютерам, які функціонують під керуванням різноманітних операційних систем. Так, наприклад, якщо користуватися комп'ютером Apple Macintosh для доступу до вузла постачальника мережевих послуг, де встановлена робоча станція Sun Microsystems під керуванням Unix, а за даними звертатися до віддаленого мейнфрейму IBM з ОС VM, то це не складе великих труднощів.

Протокол FTP дозволяє користувачу легко отримувати доступ до необхідних файлів, використовуючи при цьому нескладні програми та команди. Головна з цих програм — `ftp`, яка призначена для переміщення файлів з одного комп'ютера на ін-

ший незалежно від того, де ці комп'ютери знаходяться, як вони з'єднані між собою та з якою операційною системою працюють.

Для запуску програми `ftp` необхідно набрати команду `% ftp ім'я_віддаленого_комп'ютера`, а потім вказати ім'я користувача та пароль для входу в систему.

3.3.5. Система пошуку файлів Archie

Archie — система пошуку файлів в Internet, яка полегшує роботу з анонімними серверами FTP. Сервери Archie "пам'ятають" списки всіх файлів на багатьох серверах anonymous ftp і по запиту можуть шукати потрібний файл за ім'ям (по ключовим словам).

Реєстрацію на сервері Archie та пошук в базі даних забезпечує служба Telnet. Для цієї ж мети можна користуватися і клієнтом Archie, якщо його було встановлено на вузлі. Мати таку базу, звичайно, корисно, але для пошуку потрібної інформації немає зиску "перекачувати" її на свій комп'ютер, задаючи рядок пошуку. Програма Archie проводить індексацію всіх основних файлів Internet для здійснення пошуку потрібної інформації по назві. Деякі спеціалізовані Archie-комп'ютери працюють практично у всіх системах. Для використання Archie необхідно виконати команду `telnet` та увійти в систему найближчого сервера.

У теперішній час служба Archie індексує близько тисячі серверів та використовує десятки мільйонів файлів. За деякими оцінками, загальний об'єм файлів, що доступні через Internet, сягає 10 Тбайт. Звичайно, дуже багато інформації дублюється, тому що популярні програми зберігаються в архівах на багатьох вузлах мережі.

3.3.6. World Wide Web (WWW)

WWW (World Wide Web), W3 чи Web — це навігаційна система, чи інтерактивний засіб мультимедіа та гіпертексту, за допомогою якої можна у режимі діалогу продивлятися тексти, растрові зображення, анімаційні вставки, замовляти послуги, здійснювати пошук необхідної інформації та багато іншого. Служба WWW заснована на використанні формату мови HTML (HyperText Markup Language — мова гіпертекстової розмітки).

Гіпертекст — це інформація, яка подається в такому вигляді, коли визначені слова у тексті можна в будь-який момент "розкрити" та отримати таким чином про них додаткову інформацію, тобто ці слова є посилками на інші документи, які можуть бути текстом, файлами, малюнками і т.ін.

Спеціальні програми-браузери (Netscape Navigator, NCSA Mosaic, Arena, Microsoft Internet Explorer і ін.) зчитують файли у форматі HTML з серверу та відповідним чином інтерпретують їх для організації діалогу з користувачем. Основний принцип роботи з WWW — "викликав і маєш".

WWW іноді називають підсистемою оперативного доступу до інформації, яка підтримує засоби гіпермедіа. Служба WWW ("всесвітня павутина") була розроблена в 1989 році в Женеві дослідниками лабораторії фізики елементарних частин (European Particle Physics Laboratory) Європейського центру ядерних досліджень (CERN) на чолі з Тімом Бернерсом-Лі. У теперішній час WWW є найбільш динамічною і роз-

винутою службою Internet, що збільшує кількість нових прихильників завдяки дружньому інтерфейсу користувача та гіпертекстовому формату подання інформації.

HTML — це мова створювання гіпертекстів, що дозволяє існувати в одному документі текстовій та графічній інформації. Слова з одного документу можуть бути "прив'язані" до інших документів, що розташовані на іншому кінці земної кулі, або до графічних зображень в форматі GIF або JPEG.

WWW об'єднує більш тисячі серверів, що працюють у режимі реального часу, їх число подвоюється кожні декілька місяців.

Кожна організація або фірма, що бажає розмістити свою інформацію в системі Web, заводить власну адресну сторінку, яка являє собою екранний перелік інформації, що по функціям нагадує зміст книги. Сервер Web став фундаментом інформаційного простору в Internet.

Як правило, користувачі Internet починають свою роботу з простої електронної пошти та дуже швидко "втягуються" в маркетинг своїх послуг по мережі серверів пошуку та доступу до цих файлів за допомогою WWW. Підприємства швидко будують власні екрани (стенди) — так звані "домашні сторінки" (*home pages*), які стають "вітриною" з інформацією про компанію та її послуги.

WWW — це система, що використовує для переходів між джерелами даних гіпертекстові посилки, і дає можливість отримувати доступ до мережевих ресурсів з різних точок входу. Ця служба заснована на протоколі HTTP (Hypertext Transport Protocol — протокол передачі гіпертексту), який служить для передачі складних документів, і мові HTML, що використовує гіпертекстові зв'язки для виявлення об'єктів в середині файлів.

WWW працює наступним чином: будь-яка тема в документах WWW, виділяється блакитним чи іншим кольором і показує на інший документ на ту ж тему. Коли курсор опиняється над одним з таких слів, треба натиснути кнопку миші. В результаті з'явиться новий екран з інформацією, яка пояснює виділене слово, або система відкриє відповідний документ.

Виняткова привабливість WWW полягає у графічному, звуковому та відеооформленні документів. Аби приєднатися до WWW, користувач повинен мати спеціальне програмне забезпечення (браузери), яке розповсюджується по мережі у комплексі з більшістю інших програм та послуг Internet. Однак найбільша складність при роботі з Internet полягає в тому, що для пошуку в ній необхідно знати, де шукати те, що вам потрібно.

Взагалі можна сказати, що WWW — це спроба подати всю інформацію, яка доступна в Internet, плюс всю необхідну користувачу локальну інформацію у вигляді сукупності гіпертекстових документів. При цьому користувач пересувається по мережі, переходячи від одного документа до іншого, завдяки посилкам.

3.3.7. Служба Hyper-G

Hyper-G — це система, аналогічна WWW, але створена пізніше. Вона повторює деякі принципи роботи WWW. Клієнт Hyper-G спілкується не з будь-якими серверами, як у WWW, а тільки зі "своїм" сервером, реєструючись на ньому при підключенні. Серверів Hyper-G небагато і вони можуть працювати і як сервери Gopher, і як WWW.

3.3.8. Системи пошуку Gopher та Veronica

Gopher — це система пошуку інформації за допомогою екранних меню, вона являє собою базу даних, розподілену по комп'ютерам у всьому світі, із структурованою інформацією. Gopher дозволяє знаходити необхідну інформацію за допомогою ієрархії меню.

Служба Gopher дуже схожа на WWW, однак вони мають декілька розбіжностей. По-перше, система WWW побудована на гіпертекстових документах і структурована посилками між сторінками гіпертексту. Система Gopher не така гнучка, представлення інформації в ній базується на індивідуальних ресурсах та серверах. Gopher нічого не знає про те, що знаходиться всередині файлу, при перегляді FTP-ресурсів. Меню системи Gopher надає лише короткий опис; в цій системі відсутнє поняття зв'язку між цікавим матеріалам на одному сервері та матеріалом по тій же самій темі на іншому сервері.

По-друге, WWW краще реалізує задачу подання універсального інтерфейсу для різноманітних видів послуг, завдяки гіпертекстовій подачі та наявності всього двох команд: "слідувати за посилками" та "виконати пошук". По-третє, WWW усуває перешкоду між даними користувача та "загальнодоступними даними". Користувач може встановити Web-сервер, відповідний редактор гіпертексту та інтегрувати в WWW свої персональні нотатки, тобто замість переписування цитат та розміщення картки в картотеці, користувач створює в файлі запису посилку на документ, який цитується.

Для пошуку по ключовим словам користувача меню Gopher використовують систему Veronica. Вона призначена для вибору того чи іншого пункту, де визначений потрібний критерій пошуку. Це значно полегшує пошук по меню Gopher.

3.3.9. Служба WAIS

WAIS (Wide Area Information Service — глобальна інформаційна служба) — це система обробки індексованих баз даних різноманітної тематичної направленості. Вона дозволяє виконувати пошук в індексованому матеріалі (тексті) інформації необхідного складу, тобто — пошук в архівах Internet статей, які мають групи слів, що вказані в запиті користувача.

Зараз в Internet налічується більше тисячі загальнодоступних бібліотек WAIS. Розробка системи WAIS здійснювалась компаніями Thinking Machines, Apple, Dow Jones та Пітом Марвіном з KPMG, але зараз основну роботу по розробці безкоштовного ПЗ для WAIS координує центр CNIDR (Clearinghouse for Networked Information Discovery and Retrieval — Центр координації засобів знаходження та пошуку мережевої інформації).

3.3.10. Служба IRC

IRC (Internet Relay Chat — "балачки в Internet") — це інтерактивна служба для реалізації переговорів через Internet. Сервери IRC синхронізовані між собою, і, підключившись до одного з них, користувач підключається до всієї мережі IRC.

3.3.11. Ігри MUD та MOO

MUD (Multi User Dungeon) — гра для багатьох користувачів в Internet, у більшості випадків — ігрового чи наукового порядку.

MOO (Object-Oriented MUD) — об'єктно-орієнтовна гра для багатьох користувачів, яка служить загальноосвітнім цілям, наприклад, вивченню мови C++.

3.3.12. Програма Finger

Finger — службова програма для отримання будь-якої інформації про користувачів Internet чи інформації про ресурси та версії деяких програм.

3.3.13. Служба Telnet

Служба доступу Telnet дає можливість підключатися до дистанційно віддаленого комп'ютера і працювати з ним в інтерактивному режимі. За допомогою Telnet можна входити у бібліотечні каталоги різних університетів, шукати інформацію про будь-що — від процесів створення далеких галактик до рецептів приготування картопляного супу. Користувач може продивитись рішення Верховного суду або зчитати тексти популярних пісень.

Віддалений комп'ютер може відповідати на команди користувача, а комп'ютер користувача — працювати просто як термінал. В багатьох випадках системи на базі меню, встановлені на віддалених машинах, перетворюють інтерактивний сеанс роботи в інтуїтивний.

Після входу в систему залишається скористатися можливостями, які пропонуються на віддаленому комп'ютері. Наприклад, багато університетів надають інформацію всім бажаючим через загальнодоступні системи CWIS (Campus Wide Information System). З їх допомогою можна довідатись, що відбувається в учбових закладах різних країн світу, знайти завдяки довідникам адреси студентів та викладачів, здійснити пошук по бібліотечним каталогам.

За допомогою набору простих команд Telnet користувач може наказати вузловому комп'ютеру зробити з'єднання з іншим комп'ютером мережі, а потім зареєструватися на віддаленому комп'ютері, вводячи ідентифікатор користувача, ім'я та пароль. Загальнодоступні сервери часто самі підказують вірну послідовність команд для входу в систему, в багатьох випадках ця послідовність описується в довідниках інформаційних ресурсів Internet.

Для використання Telnet необхідно ввести на командному рівні команду `% telnet ім'я_віддаленого_комп'ютера`. Наприклад, ім'я віддаленого комп'ютера може мати вигляд `sonne.uiuc.edu`. Після цього необхідно на запит системи `login` ввести мережеве ім'я та пароль `password` на віддаленому комп'ютері.

Коли встановлено сеанс за протоколом Telnet, починає працювати програма-клієнт, яка виконується на комп'ютері користувача, та програма-сервер, що виконується на комп'ютері, який надає послугу. Мережа, що реалізує засоби протоколів TCP або UDP, являє собою середовище, через яке ці програмні модулі ПП взаємодіють між собою.

Програма-клієнт встановлює мережеве з'єднання з сервером завдяки протоколу TCP, приймає від користувача вхідні дані та перетворює їх до стандартного форма-

ту, надсилає дані серверу, приймає від сервера вихідні дані в стандартному форматі та перетворює їх для відображення на екрані комп'ютера користувача.

Програма-сервер інформує мережеве ПЗ про те, що вона готова до з'єднання, чекає на запити в стандартній формі та обслуговує його, надсилає в стандартній формі результати назад програмі-клієнту, чекає на наступний запит.

Тут існують і бази даних, багато з яких загальнодоступні. Наприклад, БД HPCwire відслідковує розвиток високопродуктивних комп'ютерів і містить інформаційні бюлетені на комп'ютерні теми. Наприклад, довідник "Online X.500 Direkty NUSERNet/PSI" надає інформацію про адреси великих організацій США; "Проект Данте" в Дартмуті надає вченим тексти "Божественної комедії" та коментарі до неї, що накопичилися протягом століть; а "Центр даних Луїса Харріса" містить унікальні матеріали для соціологів — результати опитувань Харріса, починаючи з 60-х років.

Таким чином, бачимо, наскільки широкий спектр джерел доступний завдяки Internet. При цьому, однак, наголос робиться на природничі або точні науки, проте успіх таких починань, як "Проект Данте", гарантує, що участь гуманітаріїв у функціонуванні мережі поступово зростає. Тому Telnet вважається одним з найбільш могутніх інструментів Internet, а його можливість надавати користувачам різноманітних комп'ютерних систем доступ до віддалених баз даних говорить на користь концепції міжмережевої взаємодії, яка існує завдяки протоколу TCP/IP.

3.4. Інструментальні засоби Internet

Розглянемо деякі інструментальні засоби Internet:

- **Listservs** — програма автоматичного розсилання тематичних повідомлень по електронній пошті;
- **Netfind** — програмні засоби для пошуку користувачів Internet;
- X Window System — мережева ПП, яка використовується для стандартного подання графічної інформації та читання інформації з різних пристроїв графічного вводу та клавіатури;
- **NFS** (Network File System) — мережева файлова система для колективного використання дисків та файлів;
- **Timed** — програма встановлення середнього часу на мережі та контролю за годинами з метою встановлення часу скоєння тієї чи іншої події;
- **NTP** (Network Time Protocol) — мережевий протокол синхронізації часу в мережі для роботи по так званому узгодженому всесвітньому часу;
- **Factsline** (Ibex Technogies) — пакет програм доступу до інформації в базах даних WWW для тих, хто займається маркетингом, оформленням замовлень та технічним обслуговуванням, але користується лише телефоном та факсимільним апаратом. Factsline дозволяє клієнтам, користуючись телефонним апаратом тонального набору, замовляти автоматичну відправку факсу;
- **ПЗ передачі повідомлень один одному**:
  - talk — для двосторонніх переговорів;
  - ytalk, chat — для групових дискусій;
- **IRC** (Internet Relay Chat) — сервер групових дискусій;
- **ПЗ для реалізації комп'ютерних ігор**;

- **Krowbot** — програма пошуку інформації у випадку, коли користувач не знає, як сформувати свій запит на пошук (аналог бібліотекаря-референта);
- **GNN** (Global Network Navigator) — інформаційна службова програма ("інформаційний інтерфейс" з Internet). GNN досліджує особливо цікаві обзори ресурсів, вносить доповнення та групує всю цю інформацію з урахуванням зв'язків з іншими послугами Internet;
- **Web-браузери** (програми-навігатори):
  - Mosaic (компанія Spyglass) — сімейство інтерфейсних програм, які створені для надання допомоги абонентам мережі Internet при орієнтуванні в WWW. Створене в 1992 році в Національному Центрі по впровадженню суперкомп'ютерів (National Centre for Applications) при Іллінойскому університеті. Mosaic характеризується оригінальними засобами навігації, що називаються гіперзв'язками, які дозволяють працювати з зображеннями, звуковими даними та відеоінформацією;
  - Netscape Navigator (компанія Netscape) — сучасний Web-браузер з інтерфейсом прикладних програм Navigator Inline Plugin API, який гарантує користувачам включення VRML (мова моделювання віртуальної дійсності) та SGML — стандартної узагальненої маркерної мови;
  - Voyager (компанія Virlus) — браузер, що використовує VRML;
  - MS Internet Explorer (компанія Microsoft) — Web-браузер, який підтримує HTML.
- **Засоби захисту інформації** в Internet реалізовані за рахунок спеціальних програмних засобів брандмауерів, які не дозволяють доступ до інформації несанкціонованим користувачам.
  - У теперішній час для захисту інформації в Internet широко застосовується протокол Netscape Secure Socket Layer (SSL), так званий промисловий стандарт, який використовує технологію шифрування із загальним ключем (використовують фірми Microsoft, Novell, Apple, IBM, SUR, DEC, VISA та ін.). Існує також протокол S-HTTP (Secure HyperText Protocol) компанії EIT.
  - Web STAR for Windows (компанія Quarterdeck) — ПЗ, що забезпечує захист інформації на рівні каталогів WWW та допускає організацію декількох віртуальних серверів на одному невиділеному ПК.
  - Застосовуються також програми Firewall-1 (компанія DEC), Sidewinder (компанія Secure System), SecretAgent (компанія AT&T), Netscape Commerce Server (компанія Netscape), Netscape Communications Server (компанія Netscape) тощо.

3.5. Підключення до Internet

Підключення до Internet здійснюється через постачальника мережевих послуг (так званий "Internet-провайдер"). Підключення може бути здійснено (див. рис. 1.39):

- **По звичайним телефонним лініям** з високопродуктивними модемами: через міжмережевий протокол послідовного каналу SLIP (Serial Line Internet Protocol); або через протокол PPP (Point-to-Point Protocol). Наприклад, за до-

помогою SLIP можна з'єднати домашній комп'ютер з мережею організації або університету, що з'єднані з Internet.

- **За допомогою протоколу UUCP** (UNIX-to-UNIX Copy) — це підклас комутованого доступу для всіх UNIX-систем. Система сервісних програм UUCP пересилає дані по стандартним телефонним лініям. Цей доступ дозволяє користуватися Internet-поштою, конференціями USENET або передавати дані за допомогою протоколу FTP. Протокол UUCP засновано на громіздких методах передачі файлів і вже дещо застарів, тому не дозволяє отримати доступ до таких популярних додатків Internet, як WWW, та програмам, що підтримують природний діалог між абонентами мережі.
- **Через мережу ISDN** (Integrated Services Digital Network) — це цифрова мережа з інтегрованими послугами, яка характеризується використанням цифрової телефонної лінії, що з'єднує комп'ютер користувача з комутатором (центральною станцією) телефонної мережі. Доступ по ISDN може бути або комутованим (не регулярним, а по мірі потреби), або виділеним (постійне підключення).
- **По виділеним лініям зв'язку** за допомогою високошвидкісних каналів типу T1 (швидкість передачі 1,544 Мбіт/с) або каналів з продуктивністю 56 Кбіт/с і вище, що належать компаніям та університетам. Це дає можливість із локальних мереж підприємств (LAN) отримувати доступ до Internet та застосовувати програми-навігатори типу Netscape Navigator або Microsoft Internet Explorer.
- **Комутовані з'єднання через модеми** — це можливість отримання користувачем входу в систему через комп'ютер, для якого вже існує доступ до мережі по виділеному каналу. Це самий простий доступ до Internet, хоча, можливо, не самий зручний і дешевий, тому що комп'ютер користувача не стає частиною Internet, а лише звертається до сервісної машини, яка постійно підключена до мережі. Таким способом користуються як приватні особи, так і невеликі фірми для роботи з засобами FTP, Gopher та перегляду інформації на Web-серверах. Доступ здійснюється через модем до комерційних інформаційних служб, які орієнтовані на використовування в колі родини та в учбових закладах.
- **З використанням інтерпретатора Shell системи UNIX**, що виділяється постачальником мережевих послуг — цей спосіб застосовується частіше за інші. Він дозволяє використовувати протоколи FTP та засоби Gopher, WWW.

Частина Internet, що структурно розташована на території колишнього СРСР, називається Relcom (Reliable Communications, або інша розшифровка — Russian Electronic Communications), та її українська гілка — UANet (UA — код України).

Однак, слід додати, що Relcom та UANet не можна назвати повноцінною гілкою Internet, тому що деякі вузли не використовують протокол TCP/IP. Такі вузли забезпечують роботу користувачів по технології "off-line", на відміну від технології "on-line" (методи, що використовують протокол TCP/IP).

Ці технології (методи) різняться, в основному, кількістю послуг, що надаються. В режимі "on-line" користувачу доступні всі сервери "off-line" плюс додаткові інтерактивні послуги в режимі реального часу.

Розглянемо більш детально сучасні методи доступу до Internet.

3.5.1. Системи доступу по телефонних лініях

Використовують телефонні комутовані та виділені лінії й канали. Вони одержали масове поширення, оскільки забезпечують роботу абонентів загальнодоступної телефонної мережі.

Системи доступу по телефонних лініях, що комутуються

Сучасні системи надають низькошвидкісні канали абонентам, що використовують для з'єднання з провайдером лінії зв'язку міської (міжнародної) телефонної мережі загального користування. До їхнього числа відносяться системи з повним заняттям голосового каналу, що підтримують протоколи V.34 і V.90.

Технічні можливості передачі даних по телефонних лініях, що комутуються, обмежені: швидкість обміну не перевищує 33,6 Кбіт/с (асинхронний режим, протокол V.34+). Системи доступу по протоколу V.90 дозволяють забезпечити швидкість передачі даних у бік абонента до 56 Кбіт/с, для чого сервер доступу (RAS) зв'язується з АТС цифровим каналом E1. Працювати в стандарті V.90 можуть лише абоненти цифрових АТС, інші обслуговуються по протоколу V.34+. В даний час у Києві такі системи досить поширені.

Обладнання обох стандартів широко поширено і випускається більшістю виробників. Забезпечується зустрічна робота різних модемів. Абонентське обладнання доступне масовому споживачу.

Основний недолік подібних систем — порівняно низька швидкість передачі даних, що у даний час досягла своєї межі і не може бути збільшена. Проте, пропускної здатності досить для якісного надання найбільш популярних служб Інтернет: E-mail, Web та ftp. Іншим істотним недоліком є неможливість використання телефонної лінії, зайнятою передачею даних, для голосового обміну.

Лінії, що комутуються, відрізняються високим рівнем перешкод і низкою стійкістю з'єднання, що виявляється в зривах зв'язку абонента із сервером-джерелом. І все ж таки, головною причиною невдоволення споживачів цим видом доступу є його обмежена пропускна здатність, якої на сьогоднішній день найчастіше вже недостатньо.

Системи доступу по виділених телефонних лініях

Для передачі трафіка значного обсягу (підключення локальної мережі офісу, розміщення абонентом власного Web-ресурсу, організація взаємозв'язку філій) застосовують виділені лінії.

Швидкість обміну складає від 16 Кбіт/с і може досягати 8 Мбіт/с і більше і обмежена якістю фізичного з'єднання між провайдером та абонентом. Найбільше широко застосовуваний протокол передачі HDSL, для якого прийнято ряд регламентуючих документів, однак процес стандартизації не довершений, і тому модеми різних виробників, як правило, не працюють спільно.

Використовуються синхронний та асинхронний режими передачі, а з'єднання як симетричні (Pair Gaine), так і асиметричні (Agate). Гранична дальність залежить від

необхідної швидкості, якості лінії зв'язку та інших параметрів і складає 8 км і більше без застосування регенераторів.

Модеми для виділених ліній зараз широко поширені на українському ринку. Основними недоліками організації зв'язку на виділених лініях є високі вимоги до якості фізичного з'єднання, а також значна вартість обладнання.

Обладнання провайдера

Системи абонентського доступу, у залежності від розв'язуваного набору задач, можуть будуватися на основі логічно завершених комплексів (Total Control — 3Com, Ascend — Lucent Technologies), або на основі окремих модулів, при цьому цілком припустимо використання обладнання різних виробників (наприклад, сервер доступу Cisco 2511, модемний пул Motorola Access Way або Total Control 3Com US Robotics).

При виборі обладнання для Internet-провайдера у Києві найбільш важливими критеріями є:

- однотипність усього парку обладнання;
- можливість наступного нарощування числа модемів і швидкості каналу прив'язки;
- підтримка як цифрових, так і аналогових абонентських інтерфейсів;
- можливість обслуговування комутованих та виділених ліній;
- стійкість і надійність функціонування;
- резервування найбільш відповідальних вузлів;
- можливість "гарячої" заміни комплектуючих модулів;
- розвита система моніторингу та дистанційного керування.

Цим критеріям відповідає обладнання декількох виробників (у тому числі 3Com і Cisco), на якому найбільш часто і будуються пункти доступу.

Доступ із збереженням розмовного каналу

З розвитком технологій передачі даних xDSL з'явилося обладнання, здатне забезпечити спільну передачу трафіка даних і мовного (розмовного) каналу.

Обладнання використовує протокол uDSL і випускається декількома виробниками (наприклад, Schmid Telecommunication — NTU-128). Спектр каналу передачі даних зміщений у високочастотну область, за рахунок чого збережені функції звичайного телефону (вихідні і вхідні виклики, голосовий канал). Дальність дії абонентського модему складає до 7 км і залежить від якості лінії зв'язку.

Для роботи абонентів по протоколу uDSL на АТС установлюється блок лінійного закінчення, у якому із сумарного сигналу виділяється спектр каналу передачі даних і маршрутизується до Internet-провайдера. Голосовий канал підводиться до стандартного обладнання АТС.

Безсумнівною перевагою подібних систем у наших умовах є збереження функцій телефону при зникненні електроживлення в абонента, хоча передача даних при цьому неможлива. Недоліком існуючих реалізацій є неможливість одночасної передачі даних декількома користувачами в одному (багатопарному) телефонному кабелі зв'язку з АТС.

Таким чином, застосування технології uDSL для систем масового доступу поки що обмежено. Проте, окремі реалізації таких систем уже є в Києві (послуга Lucky Line).

3.5.2. Радіо-Ethernet

Системи радіо-Ethernet застосовуються в тих випадках, коли провідні (кабельні) рішення не задовольняють вимогам замовника, наприклад, якщо неможливо забезпечити необхідну швидкість передачі через низьку якість наявної лінії зв'язку. Дальність дії систем обмежена межами прямої видимості.

До групи радіо-Ethernet відносять обладнання трьох типів, кожний з яких дозволяє вирішувати різні і цілком визначені задачі:

- Підключення офісних комп'ютерів у локальну мережу організації (обладнання Cisco Aironet, пункт доступу AP4800, абонентські радіокарти PCI4800 або PC4800; обладнання Lucent Technologies: мережевий міст WaveAccess Link WavePoint-II, абонентські пристрої WaveLAN card). Дальність дії — до 180 м на відкритій місцевості, у будинках — 10...40 м.
- Організація радіовставки в наземний канал (Cisco Aironet BR 500, BreezeLink 121). Дальність дії може досягати 30...35 км, швидкість передачі — 64...2048 Кбіт/с.
- Доступ користувачів до мережі оператора системи радіо-Ethernet через його базову станцію (обладнання Lucent Technologies: WaveAccess Link WP-II). У місті віддалення абонента від базової станції може досягати 10...15 км. Максимальна швидкість передачі, як правило, не перевищує 2 Мбіт/с (Cisco Aironet серія 340 —11 Мбіт/с, 25 км; SpeedLAN — 11 Мбіт/с, 45 км). Гранична кількість абонентів складає 10...20, при цьому ще зберігається задовільна пропускна здатність клієнтського з'єднання.

Обладнання радіо-Ethernet усіх трьох груп працює в діапазоні 2,4 Ггц, застосовуються ШПС–сигнали як із прямою модуляцією несучої (DSSS), так і ППРЧ (FHSS). Стандарт радіо-Ethernet (IEEE 802.11x) остаточно не прийнятий, тому використання обладнання різних виробників для спільної роботи, як правило, неможливо.

На українському ринку обладнання радіо-Ethernet поставляється як операторами систем радіодоступу (GU, Interstrada, IRES), так і сторонніми постачальниками.

3.5.3. Комбінована схема

У ряді випадків, коли необхідно підключити невелику групу (6–10) компактно розміщених користувачів і альтернативні рішення не влаштовують або відсутні, застосовується комбінована схема. В абонентів установлюють сервер, до якого витою парою (Ethernet) підключають користувачів. Сервер з'єднують з опорною мережею Internet-провайдера виділеним каналом зі швидкістю передачі даних 64...128 Кбіт/с, що надалі може бути розширений.

Необхідне обладнання: модеми для виділених ліній, сервер (ОС Linux, FreeBSD), концентратор, пристрій безперебійного живлення (UPS). У комп'ютерах абонентів установлюються мережеві карти Ethernet 10Base.

Перевагою комбінованої схеми є простота розгортання і висока стабільність роботи системи, при цьому абоненти одержують високошвидкісне з'єднання. Недоліком є необхідність оренди приміщення серверної і забезпечення схоронності кабельної абонентської проводки, що можливо тільки при взаємодії з організаціями, у чиєму розпорядженні знаходяться будинки.

Потреба в забезпеченні якісного доступу і відсутність прийнятних за ціною рішень на основі традиційних технологій призвели до появи на ринку нових систем, що використовують інші канали зв'язку для передачі даних абонентам.

3.5.4. Передача по мережах кабельного телебачення

Мережі кабельного телебачення за ступенем охоплення потенційних споживачів не уступають телефонним — телевізійний кабель уже проведений у кожну квартиру, а пропускна здатність одного каналу кабельної мережі на два-три порядки вище, ніж у традиційних системах передачі даних.

Обладнання доступу (міст) установлюється поруч з передавальною телевізійною станцією. Передача даних до абонента виконується в стандартному 6 МГц телевізійному каналі (у діапазоні 50–862 МГц), для побудови зворотного каналу в структурі кабельного телебачення виділяється 0,2–6,0 МГц у смузі частот 5–48 МГц, а кабельна мережа модернізується до двонаправленої. Швидкість передачі в прямому каналі складає від 10 Мбіт/с (QPSK) до 50 Мбіт/с (256 QAM), у зворотному каналі — від 7 Мбіт/с (QPSK) до 36 Мбіт/с (64 QAM).

Широкосмужні системи Internet-доступу використовують сигнали з кодовим поділом і працюють у нових мережах кабельного телебачення.

Серед виробників обладнання Internet-доступу по мережах кабельного телебачення немає єдиного уявлення про те, яким чином передавати запити користувача. Тому сьогодні випускається головне обладнання для:

- односпрямованих систем, де в зворотному каналі використовується телефонна лінія;
- двонаправлених, зі зворотним каналом у телевізійному кабелі;
- комбінованих.

У перших кабельний модем задіяний тільки для прийому даних по мережі кабельного телебачення, а для вихідного зв'язку служить звичайний телефонний модем. До числа виробників таких комплексів належать 3Com, Cisco, General Instrument, Hybrid Networks, і New Media Communication.

Ці системи задовольняють, насамперед, потреби користувачів у високошвидкісному прийомі інформації. Застосування телефонного каналу в якості зворотного дає можливість кабельним операторам, що мають односпрямовану мережу, швидко зайняти ринок домашніх високошвидкісних комунікацій. Однак у цьому випадку не вдається забезпечити основні переваги: сталість з'єднання і звільнення абонентської телефонної лінії.

Проте вважається, що такі рішення зручні для розгортання інфраструктури повносервісних Internet-провайдерів, тому пропонуються як односпрямовані кабельні служби, так і традиційний телефонні, і частково — двонаправлений кабельний сервіс там, де телемережа підтримує зворотний канал. Головна мета цієї стратегії — захопити локальний ринок домашніх Internet-послуг і отримати можливість згодом запро-

понувати користувачам перехід від телефонного дозвону до одно- або двонаправлених сучасних служб.

У 1999 році прийнята версія стандарту DOCSIS 1.1, що підтримана багатьма великими виробниками. Обладнання різних вендорів, що відповідає стандартові, здатне повноцінно працювати спільно. Кабельні модеми, що одержали сертифікат DOCSIS 1.1, з'явилися в продажі в третьому кварталі 1999 року. До 2001 року сертифікат відповідності стандартові DOCSIS 1.1 отримали дванадцять виробників головного й абонентського обладнання: 3Com, Arris (Nortel/Antec), Askey, Cisco, Com21, General Instrument, Philips Broadband Networks, Samsung, Sony, Terayon, Thomson Consumer Electronics(RCA), Toshiba.

Прийняття стандарту дозволяє операторові експлуатувати спільно обладнання різних виробників, наприклад, головне обладнання Cisco 7246 і абонентські модеми E-Tech.

3.5.5. Супутникові системи доступу

Системи Internet-доступу із супутниковими каналами відносяться до національних, оскільки охоплюють територію, порівнянну з розмірами держави. Вони призначені, насамперед, для забезпечення роботи абонентів на периферії — там, де традиційні канали зв'язку прийнятної якості відсутні.

Першими в цій області стала компанія Hughes Network Systems із системою DirecPC, потім на ринок вийшли ComBox, New Media Communication і ряд інших компаній.

Структура систем передачі даних Internet із супутниковим зворотним каналом у різних виробників багато в чому подібна і включає:

- головне обладнання:
  - апаратно-програмний комплекс мережевого операційного центру NOC;
  - супутниковий ретранслятор;
- абонентське обладнання:
  - карта супутникового модему з програмним забезпеченням клієнта;
  - супутникова прийомна система (антена з конвертором);
  - звичайний телефонний модем.

Найбільш важливими характеристиками, що визначають показники системи, є:

- вид модуляції сигналу — QPSK;
- швидкість групового потоку даних — від 2 до 53 Мбіт/с;
- формат передачі інформації — DVB /MPEG-2.

Системи надають широкий спектр послуг, до числа яких відносяться:

- Високошвидкісний прийом даних користувачами Internet. Для звичайного абонента це, насамперед, означає можливість швидко одержувати дані в необхідному обсязі.
- Пакетне розсилання інформації заздалегідь обраній групі абонентів одночасно.
- Доставка інформації телеконференцій для Internet-провайдерів. Здатність системи передавати інформацію в "віщальному" режимі також дозволяє застосовувати її для дистанційного навчання.

- Прийом новин, сигналів цифрового відео й аудіо в реальному масштабі часу. Надалі цей режим дозволить розширити набір пропонованих послуг, наприклад, шляхом трансляції відеопрограм "по запиту".
- Доступ багатьох користувачів до корпоративних мереж. Штатна можливість підтримки локальних (корпоративних) мереж дозволяє забезпечити спільний доступ до телефонного і супутникового модемів.

Потенційними споживачами послуг є:

- індивідуальні користувачі Internet;
- корпоративні користувачі (побудова власних Intranet-мереж);
- провайдери Internet- послуг;
- інформаційні агентства, телерадіокомпанії, редакції газет і журналів, рекламні агентства.

Усі системи працюють у Ku-діапазоні (11 ГГц) і забезпечують стійкий прийом даних ($P_{ош}$ не гірше 10^{-6}) на дзеркальні антени ∅ 0,9...1,8 м по всій території України.

Основний недолік систем із супутниковим каналом прийому полягає в принципі організації передачі потоків даних: усі запити абонентів маршрутизуються через мережевий операційний центр (NOC — Network Operations Center). Тому місце розташування NOC істотно впливає на вартість послуг. За оцінками ряду американських Internet-провайдерів, більше 65% їх сумарного трафіка має локальне призначення — адресується абонентам того ж провайдера, і лише частина запитів, що залишилася, направляється в "зовнішній світ". Також обмежує застосування подібних систем значна затримка поширення сигналів через геостаціонарний супутник.

3.5.6. Перспективні системи

Системи передачі даних по мережах електроживлення

Мережі електроживлення є в кожній квартирі і тому становлять інтерес як транспортна основа для систем передачі даних. Промислові зразки абонентського обладнання для комп'ютерних локальних мереж виробництва Intelogic PassPort і Tut Systems HomeRun з'явилися в середині 1998 року і являють собою адаптери, що підключаються до розеток мережі електроживлення 220 В. Швидкість передачі даних складає від 0,35 до 1,3 Мбіт/с. Розміри комп'ютерної мережі обмежені межами гальванічно зв'язаної мережі електроживлення (до найближчого силового трансформатора). В якості дисципліни доступу до ресурсів використаний протокол Ethernet.

Поширення в Україні подібні мережі поки що не отримали: в умовах високої щільності заселення (багатоквартирні будинки) пропускної і навантажувальної здатності виявилося явно недостатньо. Проте, розробки в цій області продовжуються, а можливість використовувати для передачі дані мережі електроживлення залишається як і раніше привабливою.

Нещодавно німецька фірма Veba представила проект створення комп'ютерної мережі в межах будинку, на базі ліній електропередачі "ONELINE All services". Система проходить тестування в декількох сотнях житлових будинків у Німеччині. Будинковий пристрій передачі даних ONELINE Box розташовується поруч з лічильником електроенергії і через мережу електроживлення зв'язується з подмемо ONELINE

Inhouse Modem, що розташований безпосередньо в квартирі абонента. Модем може бути підключений у будь-яку розетку в квартирі, забезпечує передачу даних зі швидкістю до 10 Мбіт/с і має інтерфейс Ethernet для з'єднання з комп'ютером. До ONELINE Box може бути підключене до 30 телефонних ліній і канал передачі даних Internet.

Системи бездротового Internet -доступу

Проблеми організації високошвидкісних магістральних каналів зв'язку, пошук транспортних систем, здатних забезпечити швидке охоплення великої кількості абонентів, необхідність зниження експлуатаційних витрат змушують звернути увагу на системи бездротового Internet -доступу (MMDS).

Широкосмужні бездротові системи відносяться до міських і дозволяють забезпечити високошвидкісну (до 2 Мбіт/с на абонента) двосторонню передачу даних. Швидкість групового потоку в одному стандартному телеканалі (6 МГц) — до 30 Мбіт/с, вид модуляції — QAM (у зворотному каналі QPSK). Радіус зони покриття до 35–40 км, у залежності від типу системи і діапазону її робочих частот (табл. 3.1).

Таблиця 3.1

Тип системи бездротового Internet-доступу	Опис
MMDS (multichannel multipoint distribution service)	Більшість телепередавачів системи аналогові; для формування субканалів потрібна установка цифрових передавачів
MDS (multipoint distribution service)	Старі системи аналогового телебачення
WCS (wireless communications services)	Нові системи телебачення
ITFS (instructional television fixed service)	Застосовуються в системах навчання й Internet-доступу
LMDS (local multipoint distribution service)	Нові системи телебачення
ISM (instructional scientific and medical)	Використовують неліцензовані частотні смуги для побудови радіомереж і формування зворотного каналу в двонаправлених системах

Робочі частоти бездротового Internet-доступу наведені в табл. 3.2.

Таблиця 3.2

Тип	Частота, МГц	Примітки
MMDS	2500–2686	Забезпечує передачу до 31 телеканалу 6 МГц; у тому числі освітніх. Віддалення абонентів — до 35–40 км на відстані прямої видимості
MDS1	2150–2156	Єдиний канал 6 МГц; подібний MMDS.
ITFS	2500–2690	Канали 6 МГц у смузі MMDS
LPTV	54–72, 78–88 174–216, 470–806	Малопотужне віщання (до 50 КВт ефективної випромінюваної потужності, з урахуванням посилення антени); канали 6 МГц. Ліцензується як експериментальна. Забезпечує стійкий прийом на відстані прямої видимості

Таблиця 3.2. Закінчення

Тип	Частота, МГц	Примітки
LMDS	27500–28350 31000–31300	Мала дальність, 3–5 км, канали 20 МГц. Поширення радіохвиль залежить від погодних умов (дощ, сніг)
ISM	2,400–2,483.5	Мала дальність, але може працювати на відстанях 15–20 км. Застосовується для формування зворотного каналу в двонаправлених системах

Архітектура

Систему MMDS утворюють наступні елементи (рис. 3.4):

- базова (передавальна) станція із всеспрямованою або секторною антеною;
- малопотужні ретранслятори;
- абонентські (приймальні) пункти.

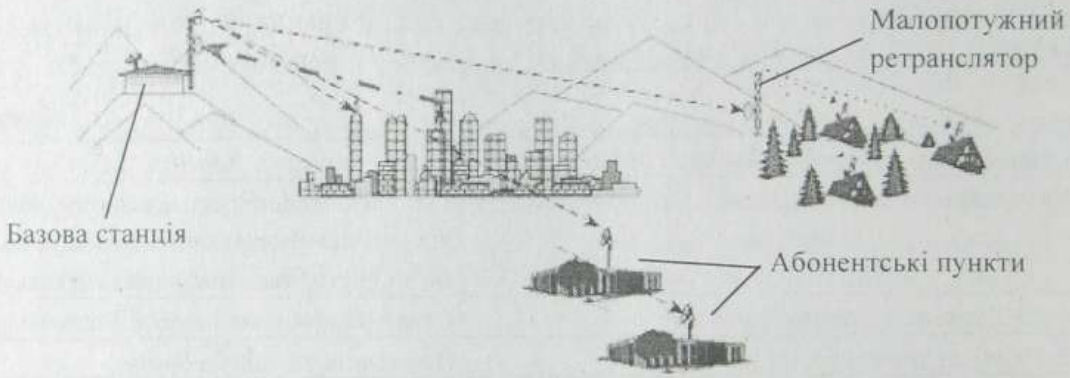


Рис. 3.4. Архітектура MMDS

Системи передачі MMDS/WCS для Internet-доступу включають канальні передавачі, комбайнери, підсистему автоматичного резервування й обладнання мережного керування. Системи передачі підтримують протокол SNMP для мережевого конфігурування, керування і відновлення в аварійних ситуаціях.

Абонентське обладнання включає конвертор і кабельний модем з інтерфейсами Ethernet 10Base і RS-232. Зворотний трафік передається через стандартний телефонний модем. Абонентський кабельний модем дозволяє обслуговувати до 20 абонентських ПК (через додатковий Ethernet-комутатор).

Можлива організація змішаного радіодоступу в Internet — з ефірними і телефонним зворотними каналами одночасно. При дуплексному радіодоступі абонентський модем формує QPSK сигнал, що дозволяє використовувати малопотужні прийомо-передавачі на відстанях до 30 км від головної станції.

Підтримка в одній системі як телефонних, так і ефірних зворотних каналів дає операторові можливість, почавши з телефонного зворотного каналу, перейти надалі до двонаправленої роботи з радіо.

Найчастіше для зворотного каналу використовується діапазон 2150–2162 МГц, хоча виробляється також обладнання і для інших діапазонів: 2305–2320 або 2345–2360 МГц, у тому числі 2500–2686 МГц, однак такі рішення виявляються більш дорогими.

Двосторонній доступ в Internet через MMDS є привабливим вибором для Internet-провайдерів у країнах, що розвиваються. Це рішення дозволяє обійти як проблему неякісних місцевих телефонних ліній, так і погодинних тарифів, що діють у місцевих телефонних операторів.

В США системи MMDS використовуються для трансляції розважальних відеопрограм і дистанційного навчання. Кілька систем розгорнуто на території СНД, наприклад, у Києві (MITPIC). Система TOCOM® виробництва General Instrument використовується з 1992 року у мережах MMDS "Космос-ТВ" у Москві, з 1998 року — "Тіло-Плюс" у Санки-Петербурзі, "СХИЛ-7" у Череповці. ("Космос-ТВ" належить американської корпорації "ITI/Метромедія", що розгорнула аналогічні системи в 17 містах СНД.)

У Новгороді використовується подібна система виробництва Eastern Electronics Corp. У Владивостоці встановлена готельна система Guestlink (готель "Versailles"). В 1999 році державне унітарне підприємство "Радіотелемережі" приступило до створення в Казані телевізійної системи MMDS, що дозволяє приймати більш 20 телевізійних каналів.

В даний час у Росії отримано кілька ліцензій на побудову мереж LMDS. Компанія "Рустелеком" (у деяких джерелах вона фігурує як "Севіар") провела тестування і демонстрацію служби по наданню комбінованого доступу в Internet (до користувача — по системі LMDS, назад — по телефонній лінії).

Перешкоди, що стримують побудову мереж передачі даних і доступу в Internet на базі систем LMDS:

- завантаженість цього діапазону іншими радіозасобами;
- заборони на одержання ліцензії на користуваннями частотами в діапазонах так бездротового кабельного телебачення (MMDS, LMDS і MVDS);
- високі в порівнянні з мережами радіо-Ethernet первісні витрати на розгортання системи.

3.6. Адресація в Internet

Кожен комп'ютер у мережі TCP/IP має адреси трьох рівнів:

- Локальна адреса вузла, обумовлена технологією, за допомогою якої побудована окрема мережа, до якої відноситься даний вузол. Для вузлів, що входять у локальні мережі — це так звана ***MAC-адреса*** мережевого адаптеру чи порту маршрутизотора, наприклад, **11-AO-17-3D-BC-01**. Ці адреси призначаються виробниками устаткування і є унікальними, тому що керуються централізовано. Для всіх існуючих технологій локальних мереж MAC-адреса має формат 6 байтів: старші 3 байти — ідентифікатор фірми виробника; молодші 3 байти призначаються унікальним чином самим виробником. Для вузлів, що входять у глобальні мережі, локальна адреса призначається адміністратором глобальної мережі.
- ***IP-адреса***, що складається з 4 байт, наприклад, **109.26.17.100**. Ця адреса використовується на мережевому рівні й призначається адміністратором під час конфігурування комп'ютерів і маршрутизоторів. IP-адреса складається з двох частин: номера мережі і номера вузла. Номер мережі може бути обраний ад-

міністратором довільно або призначений за рекомендацією спеціального підрозділу Internet (NIC), якщо мережа повинна працювати як складова частина Internet. Звичайно провайдери послуг Internet одержують діапазони адрес у підрозділів NIC, а потім розподіляють їх між своїми абонентами.

Номер вузла в протоколі IP призначається незалежно від локальної адреси вузла. Поділ IP-адреси на поле номера мережі і номери вузла — гнучкий і межа між цими полями може встановлюватися дуже довільно. Вузол може входити в кілька IP-мереж. У цьому випадку він повинен мати кілька IP-адрес, по числу мережевих зв'язків. У такий спосіб IP-адреса характеризує не окремий комп'ютер чи маршрутизотор, а одне мережеве з'єднання.

- ***Символьний ідентифікатор-ім'я***, наприклад, `serv1.ibm.com`. Ця адреса призначається адміністратором і складається з декількох частин, наприклад, імені машини, імені організації, імені домену. Така адреса, яку називають також DNS-ім'ям, використовується на прикладному рівні, наприклад, у протоколах FTP чи Telnet.

3.6.1. IP-адреса

Структура IP-адреси вже розглядалась вище в цьому ж розділі. Розглянемо більш детально технічні нюанси її побудови та використання.

Основні класи IP-адрес

IP-адреса має довжину 4 байти і, звичайно, записується у вигляді чотирьох чисел, що представляють значення кожного байта в десятковій формі і відділені одне від одного крапками, наприклад: **128.10.2.30** — традиційна десяткова форма представлення адреси. Двійкова форма представлення цієї ж адреси:

10000000 00001010 00000010 00011110.

Адреса складається з двох логічних частин: номеру мережі і номеру вузла в мережі. Яка частина адреси відноситься до номера мережі, а яка до номера вузла, визначається значеннями перших бітів адреси:

- Якщо адреса починається з **0**, то мережу відносять до класу **A**, і номер мережі займає один байт. Інші 3 байти інтерпретуються як номер вузла в мережі. Мережі класу A мають номери в діапазоні від 1 до 126. Номер 0 не використовується, а номер 127 зарезервований для спеціальних цілей, про що буде сказано нижче. У мережах класу A кількість вузлів повинна бути більше 216, але не перевищувати 224.
- Якщо перші два біти адреси дорівнюють **10**, то мережа відноситься до класу **B** і є мережею середніх розмірів з числом вузлів 28–216. У мережах класу **B** під адресу мережі і під адресу вузла виділено по 16 бітів, тобто по 2 байти.
- Якщо адреса починається з послідовності **110**, то це — мережа класу **C** з числом вузлів не більше 28. Під адресу мережі виділено 24 біти, а під адресу вузла — 8 бітів.
- Якщо адреса починається з послідовності **1110**, то вона є адресою класу **D** і позначає особливу, групову адресу — multicast. Якщо в пакеті в якості адреси

призначення зазначена адреса класу D, то такий пакет повинні одержати усі вузли, яким присвоєна дана адреса.
- Якщо адреса починається з послідовності **11110**, то це — адреса класу E, вона зарезервована для майбутніх застосувань.

Угоди про спеціальні адреси broadcast, multicast та loopback

У протоколі IP існує кілька угод про особливу інтерпретацію IP-адрес:
- якщо IP-адреса складається лише з двійкових нулів, то вона позначає адресу того вузла, що з генерував цей пакет;
- якщо в полі номера мережі стоять нулі, то за замовчуванням вважається, що цей вузол належить тій же самій мережі, що й вузол, який відправив пакет;
- якщо всі двійкові розряди IP-адреси містять одиниці, то пакет з такою адресою призначення повинен розсилатися усім вузлам, що знаходяться в тій же мережі, що і джерело цього пакета. Таке розсилання називається ***обмеженим широкомовним повідомленням*** (*limited broadcast*);
- якщо в полі адреси призначення стоять суцільні одиниці, то пакет, що має таку адресу розсилається усім вузлам мережі з заданим номером. Таке розсилання називається ***широкомовним повідомленням*** (*broadcast*);
- адреса **127.0.0.1** зарезервована для організації зворотного зв'язку при тестуванні роботи програмного забезпечення вузла без реального відправлення пакета по мережі. Ця адреса має назву "***зворотна петля***" (*loopback*).

Уже згадувана форма групової IP-адреси — multicast — означає, що даний пакет повинен бути доставлений відразу декільком вузлам, що утворюють групу з номером, зазначеним у полі адреси. Вузли самі ідентифікують себе, тобто визначають, до якої із груп вони відносяться. Той самий вузол може входити в кілька груп. Такі повідомлення на відміну від широкомовних називають мультимовними. Групова адреса не поділяється на поля номеру мережі й вузла і обробляється маршрутизатором особливим чином.

У протоколі IP відсутнє поняття широкомовності в тому змісті, у якому воно використовується в протоколах канального рівня локальних мереж, коли дані повинні бути доставлені абсолютно всім вузлам. Як обмежена широкомовна IP-адреса, так і широкомовна IP-адреса мають межі поширення в мережі — вони обмежені або мережею, до якої належить вузол-джерело пакета, або мережею, номер якої зазначений в адресі призначення. Тому поділ мережі за допомогою маршрутизаторів на частини локалізує широкомовний шторм межами однієї зі складових частин просто тому, що немає способу адресувати пакет одночасно усім вузлам усіх мереж складеної мережі.

3.6.2. Відображення фізичних адрес на IP-адреси: протоколи ARP і RARP

У протоколі IP адреса вузла, тобто адреса комп'ютера чи порту маршрутизотора, призначається довільно адміністратором мережі і прямо не зв'язана з його локальною адресою, як це зроблено, наприклад, у протоколі IPX. Підхід, використовуваний у IP, зручно використовувати у великих мережах і через його незалежність від

формату локальної адреси, і через стабільність, тому що в протилежному випадку, при зміні на комп'ютері мережевого адаптера цю зміну повинні б були враховувати всі адресати всесвітньої мережі Internet (у тому випадку, звичайно, якщо мережа підключена до Internet).

Локальна адреса використовується в протоколі IP лише в межах локальної мережі при обміні даними між маршрутизотором і вузлом цієї мережі. Маршрутизотор, отримавши пакет для вузла однієї з мереж, безпосередньо підключених до його портів, повинен для передачі пакета сформувати кадр відповідно до вимог прийнятої в цій мережі технології і вказати в ньому локальну адресу вузла, наприклад його MAC-адресу. У пакеті, що прийшов, ця адреса не зазначена, тому перед маршрутизотором постає задача пошуку його по відомій IP-адресі, що зазначена в пакеті як адреса призначення. З аналогічною задачею зіштовхується і кінцевий вузол, коли він хоче відправити пакет у віддалену мережу через маршрутизотор, підключений до тієї ж локальної мережі, що і даний вузол.

Для визначення локальної адреси по IP-адресі використовується протокол дозволу адреси ARP (Address Resolution Protocol). Протокол ARP працює по-різному, в залежності від того, який протокол канального рівня застосовується в даній мережі: протокол локальної мережі (Ethernet, Token Ring, FDDI) з можливістю широкомовного доступу одночасно до усіх вузлів мережі; чи протокол глобальної мережі (X.25, frame relay), що як правило не підтримує широкомовний доступ.

Існує також протокол, що вирішує зворотну задачу — знаходження IP-адреси по відомій локальній адресі. Він називається реверсивний ARP — RARP (Reverse Address Resolution Protocol) і використовується при включенні бездискових станцій, що не знають у початковий момент своєї IP-адреси, але знають адресу свого мережевого адаптера.

У локальних мережах протокол ARP використовує широкомовні кадри протоколу канального рівня для пошуку в мережі вузла із заданою IP-адресою.

Вузол, якому потрібно виконати відображення IP-адреси на локальну адресу, формує ARP-запит, вкладає його в кадр протоколу канального рівня, вказуючи в ньому відому IP-адресу, і розсилає запит широкомовно. Усі вузли локальної мережі одержують ARP-запит і порівнюють зазначену в ньому IP-адресу з власною. У випадку їхнього збігу вузол формує ARP-відповідь, в якій вказує свою IP-адресу і свою локальну адресу і відправляє його вже цілеспрямовано, тому що в ARP-запиті відправник указує свою локальну адресу. ARP-запити і відповіді використовують той самий формат пакета. Оскільки локальні адреси можуть у різних типах мереж мати різну довжину, то формат пакета протоколу ARP залежить від типу мережі.

У полі типу мережі для мереж Ethernet вказується значення **1**. Поле типу протоколу дозволяє використовувати пакети ARP не лише для протоколу IP, але і для інших мережевих протоколів. Для IP значення цього поля дорівнює **080016**.

Довжина локальної адреси для протоколу Ethernet дорівнює 6 байтам, а довжина IP-адреси — 4 байтам. У полі операції для ARP запитів вказується значення **1** для протоколу ARP і **2** для протоколу RARP.

Вузол, що відправляє ARP-запит, заповнює в пакеті всі поля, крім поля шуканої локальної адреси (для RARP-запиту не вказується шукана IP-адреса). Значення цього поля заповнюється вузлом, що розпізнав свою IP-адресу.

У глобальних мережах адміністратору мережі найчастіше доводиться вручну формувати ARP-таблиці, у яких він задає, наприклад, відповідність IP-адреси адресі вузла мережі X.25, що має зміст локальної адреси. Останнім часом намітилася тенденція автоматизації роботи протоколу ARP і в глобальних мережах. Для цієї мети серед усіх маршрутизоторів, підключених до якої-небудь глобальної мережі, виділяється спеціальний маршрутизотор, що веде ARP-таблицю для всіх інших вузлів і маршрутизоторів цієї мережі.

При такому централізованому підході для усіх вузлів і маршрутизоторів вручну потрібно задати лише IP-адреси і локальну адресу виділеного маршрутизотора. Потім кожен вузол і маршрутизотор реєструє свої адреси у виділеному маршрутизоторі, а при необхідності установлення відповідності між IP-адресою і локальною адресою вузол звертається до виділеного маршрутизотора з запитом і автоматично одержує відповідь без участі адміністратора.

3.6.3. Відображення символьних адрес на IP-адреси

Вище в цьому ж розділі вже згадувалась служба доменних імен. По суті, DNS — це розподілена база даних, що підтримує ієрархічну систему імен для ідентифікації вузлів у Internet. Ця служба призначена для автоматичного пошуку IP-адреси по відомому символьному імені вузла. Специфікація DNS визначається стандартами RFC 1034 і 1035. DNS вимагає статичної конфігурації своїх таблиць, що відображають імена комп'ютерів у IP-адресу.

Протокол DNS є службовим протоколом прикладного рівня. Цей протокол несиметричний: у ньому визначені DNS-сервери і DNS-клієнти. DNS-сервери зберігають частину розподіленої бази даних про відповідність символьних імен та IP-адрес. Ця база даних розподілена по адміністративним доменам Internet. Клієнти сервера DNS знають IP-адресу сервера DNS свого адміністративного домену і по протоколу IP передають запит, у якому повідомляють відоме символьне ім'я і просять повернути відповідну йому IP-адресу.

Якщо дані про запитану відповідність зберігаються в базі даного DNS-сервера, то він одразу посилає відповідь клієнту, якщо ж ні — він надсилає запит DNS-серверу іншого домену, що може сам обробити запит, або передати його іншому DNS-серверу.

Усі DNS-сервери з'єднані ієрархічно, відповідно до ієрархії доменів Internet. Клієнт опитує ці сервери імен, поки не знайде потрібні відображення. Цей процес прискорюється через те, що сервери імен постійно кешують інформацію, надану по запитах. Клієнтські комп'ютери можуть використовувати у своїй роботі IP-адреси декількох DNS-серверів, для підвищення надійності своєї роботи.

База даних DNS має структуру дерева, яку називають ***доменним простором імен***, в якому кожен домен (вузол дерева) має ім'я і може містити піддомени. Ім'я домену ідентифікує його положення в цій базі даних стосовно батьківського домену, причому крапки в імені відокремлюють частини, що відповідають вузлам домену.

Домен (*domain*) — це фрагмент, що описує в текстовій формі частину адреси, подібно тому як це робиться при оформленні конвертів звичайних листів, але, на відміну від них, у доменній адресі (так само як і в інших використовуваних текстових

адресах) не допускається, використання пробілів. У конкретних адресах може бути представлене різне число доменів.

➢ Приклади доменних імен розглядались вище в пункті "3.2.2. Система доменних імен (DNS)".

3.6.4. Протокол DHCP

Як уже було сказано, IP-адреси можуть призначатися адміністратором мережі вручну. Це представляє для адміністратора складну процедуру. Ситуація ускладнюється ще тим, що багато користувачів не мають достатніх знань для того, щоб конфігурувати свої комп'ютери для роботи в Internet і тому повинні покладатися на адміністраторів.

Протокол DHCP (Dynamic Host Configuration Protocol — протокол динамічного конфігурування хосту) був розроблений для того, щоб звільнити адміністратора від цих проблем. Основним призначенням DHCP є динамічне призначення IP-адрес. Однак, крім динамічного, DHCP може підтримувати і більш прості способи ручного й автоматичного статичного призначення адрес.

У ручній процедурі призначення адрес активну участь приймає адміністратор, що надає DHCP-серверу інформацію про відповідність IP-адрес фізичним адресам чи іншим ідентифікаторам клієнтів. Ці адреси повідомляються клієнтам у відповідь на їхні запити до DHCP-сервера.

При автоматичному статичному способі DHCP-сервер присвоює IP-адресу (і, можливо, інші параметри конфігурації клієнта) з пула наявних IP-адрес без втручання оператора. Межі пула призначуваних адрес задає адміністратор при конфігуруванні DHCP-сервера. Між ідентифікатором клієнта і його IP-адресою, як і раніше, при ручному призначенні, існує постійна відповідність. Вона встановлюється в момент первинного призначення сервером DHCP IP-адреси клієнту. При всіх наступних запитах сервер повертає ту ж саму IP-адресу.

При динамічному розподілі адрес DHCP-сервер видає адресу клієнту на обмежений час, що дає можливість згодом повторно використовувати IP-адреси іншими комп'ютерами. Динамічний поділ адрес дозволяє будувати IP-мережу, кількість вузлів у якій набагато перевищує кількість наявних у розпорядженні адміністратора IP-адрес.

Протокол DHCP забезпечує надійний і простий спосіб конфігурування мережі TCP/IP, гарантуючи відсутність конфліктів адрес за рахунок централізованого керування їхнім розподілом. Адміністратор керує процесом призначення адрес за допомогою параметра "тривалості оренди" (*lease duration*), що визначає, як довго комп'ютер може використовувати призначену IP-адресу, перед тим як знову запросити його від сервера DHCP в оренду.

Прикладом роботи протоколу DHCP може служити ситуація, коли комп'ютер, що є клієнтом DHCP, видаляється з підмережі. При цьому призначена йому IP-адреса автоматично звільняється. Коли комп'ютер підключається до іншої підмережі, то йому автоматично призначається нова адреса. Ані користувач, ані мережевий адміністратор не втручаються в цей процес. Ця властивість дуже важлива для мобільних користувачів.

Протокол DHCP використовує модель "клієнт-сервер". Під час запуску системи комп'ютер-клієнт DHCP, що знаходиться в стані "ініціалізація", посилає повідомлення "discover" (дослідити), що широкомовно поширюється по локальній мережі і передається всім DHCP-серверам приватної мережі. Кожен DHCP-сервер, що одержав це повідомлення, відповідає на нього повідомленням "offer" (пропозиція), що містить IP-адресу і конфігураційну інформацію.

Комп'ютер-клієнт DHCP переходить у стан "вибір" і збирає конфігураційні пропозиції від DHCP-серверів. Потім він вибирає одну з цих пропозицій, переходить у стан "запит" і відправляє повідомлення "request" (запит) тому DHCP-серверу, чия пропозиція була обрана.

Обраний DHCP-сервер посилає повідомлення "acknowledgment" (підтвердження), що містить ту ж IP-адресу, що уже була відправлена раніше на стадії дослідження, а також параметр оренди для цієї адреси. Крім того, DHCP-сервер посилає параметри мережевої конфігурації. Після того, як клієнт одержить це підтвердження, він переходить у стан "зв'язок", знаходячись в якому він може брати участь у роботі мережі TCP/IP.

Комп'ютери-клієнти, що мають локальні диски, зберігають отриману адресу для використання при наступних запусках системи. При закінченні терміну оренди адреси комп'ютер намагається обновити параметри оренди в DHCP-сервера, а якщо ця IP-адреса не може бути виділена знову, то йому повертається інша IP-адреса.

У протоколі DHCP описується кілька типів повідомлень, що використовуються для виявлення і вибору DHCP-серверів, для запитів інформації про конфігурацію, для продовження і припинення ліцензії на IP-адресу. Усі ці операції спрямовані на те, щоб звільнити адміністратора мережі від складних рутинних операцій по конфігуруванню мережі.

Однак використання DHCP несе в собі і деякі проблеми. По-перше, це проблема узгодження інформаційної адресної бази в службах DHCP і DNS. Як відомо, DNS служить для перетворення символьних імен у IP-адреси. Якщо IP-адреси будуть динамічно змінюватися сервером DHCP, то ці зміни необхідно також динамічно вносити в базу даних сервера DNS. Хоча протокол динамічної взаємодії між службами DNS і DHCP уже реалізований деякими компаніями (так звана служба Dynamic DNS), стандарт на нього поки що не прийнятий.

По-друге, нестабільність IP-адрес ускладнює процес керування мережею. Системи керування, засновані на протоколі SNMP, розроблені з розрахунком на статичність IP-адрес. Аналогічні проблеми виникають і при конфігуруванні фільтрів маршрутизаторів, що оперують з IP-адресами.

Нарешті, централізація процедури призначення адрес знижує надійність системи. При відмові DHCP-серверу всі його клієнти виявляються не в змозі одержати IP-адресу й іншу інформацію про конфігурацію. Наслідки такої відмови можуть бути зменшені шляхом використання в мережі декількох серверів DHCP, кожен з який має свій пул IP-адрес.

3.6.5. Адресація в електронній пошті

Оскільки в Internet забезпечується доступ не тільки до комп'ютерів, але і до різноманітних програм електронної пошти, що є комп'ютерним аналогом звичайної пошти, то передбачена адресація особистостей, що беруть участь в листуванні.

Поштова адреса користувача має наступну структуру:

`ім'я_користувача@адреса_комп'ютера.`

Оскільки вже розглянута адресація комп'ютерів безумовно повинна забезпечувати унікальність адреси кожного комп'ютера, з ім'ям користувача усе ще простіше. У загальному випадку, формат поштової адреси має на увазі, що один комп'ютер може бути використаний і декількома користувачами, імена яких (у такій групі) повинні, природно, розрізнятися. Широке поширення на практиці одержало також використання декількох (альтернативних) імен для одного користувача.

Оскільки електронна пошта (не обов'язково з адресацією описаного формату) використовується не тільки в Internet, але й в інших комп'ютерних мережах, відзначимо, що в Internet вона відрізняється розширеними можливостями і підвищеною оперативністю, що перетворює її фактично в експрес-пошту.

3.6.6. Адресація документів у Web-технології

У Web-технології документом прийнято називати не зовсім те, що прийнято вважати таким у звичайному житті, а вміст того чи іншого файлу, незалежно від характеру інформації, розміщеної в ньому. Насправді ж у файлі може зберігатися гіпертекстовий документ, якась частина документу, наприклад, ілюстрація і програма, що навіть виконується, (чи якась її частина).

В даному випадку адресою документа є так званий ***URL*** (Uniform Resource Locator — уніфікований покажчик інформаційного ресурсу), що включає весь комплекс відомостей, необхідних для його пошуку і правильної інтерпретації тим чи іншим Web-браузером. Найбільш популярними серед браузерів на сьогодні є Netscape Navigator і Microsoft Internet Explorer.

Розглянемо, наприклад, наступну адресу:

`http://www.relcom.ru/Internet/Literature/index.html`.

Розглянемо призначення складових частин цього URL:

- **`http`** — визначає протокол, тобто спосіб передачі документа. У даному випадку, документ повинен бути переданий як гіпертекстовий. Можливі варіанти: `ftp`, `gopher` і деякі інші.
- **`www.relcom.ru`** — адреса сервера. Домен `www` не обов'язково повинен бути явним покажчиком типу сервера — це справа смаку. Адреса сервера може бути вказана і в числовій формі.
- **`/Internet/Literature/`** — каталог чи шлях до шуканого файлу у файловому архіві сервера.
- **`index.html`** — ім'я файлу, що включає розширення `html`. Це вказує, що документ підготовлений мовою HTML. Для одного з файлів кожного каталогу (звичайно з ім'ям `index.html`) така вказівка може бути і зовсім опущеною, оскільки його ім'я визначається за замовчуванням.

Ніякого загального тематичного каталогу для документів усіх серверів не існує, оскільки в умовах їхнього постійного поновлення це просто неможливо. Далеко не всі сервери до того ж вузькоорієнтовані тематично. Час від часу, щоправда, особливо завзяті мережеві літератори видають так називані "Жовті сторінки" зі спробою дати хоча б коротку характеристику всіх інформаційних серверів — світових чи регіональних. Ці дуже корисні книги встигають трохи застаріти вже до моменту їхнього виходу у світ. Зважаючи на це, пошук документів залишається проблемою, що намагаються вирішувати за допомогою різних пошукових програм, чого у реальній практиці не завжди вдається досягти з очікуваною ефективністю.

3.6.7. Адресація і мережева інтеграція в Internet

Представлені вище основи доменної адресації були розроблені задовго до фактичного створення і зміцнення Internet, і така адресація вже широко використовувалась і використовується й у мережах, що не підтримують IP-технологію. Історично існує кілька варіантів такого підключення до Internet, що забезпечують користувачам практично повний доступ до ресурсів мережі, але з істотною втратою якості за рахунок, насамперед, відсутності можливості (прямого) використання засобів Web-технології. Робота виходить технологічно істотно більш складною і, як правило, повільною, але зате обмежується дуже невеликими вимогами до конфігурації комп'ютерів користувачів і, нерідко, — навіть до характеристик ліній зв'язку, що використовують для їхнього підключення до Internet.

Поштові мережі

Ряд комп'ютерних мереж відноситься до так званих поштових мереж чи систем, що базуються практично на наданні лише однієї послуги: електронної пошти. Такі мережі допускають успішну роботу навіть на комп'ютерах IBM PC XT з дуже скромними характеристиками і, у порівнянні з IP-мережами, відрізняються невибагливістю до якості ліній зв'язку. Останнє пов'язане з тим, що, з одного боку, передбачається, що вся основна робота користувача проводиться на їхніх комп'ютерах автономно, тобто без яких-небудь контактів з мережею, і лише епізодично і на досить короткий час проводиться сеанс зв'язку для обміну накопиченими (обома сторонами) листами.

З іншого боку, у таких мережах передача листів здійснюється послідовно від вузла до вузла, і тому не вимагає встановлення прямого зв'язку на всьому маршруті, що використовується в телефонії. В результаті робота в таких мережах виявляється, як правило, істотно більш дешевою, але зате і досить "тихохідною": в залежності від ситуації передача листа в кінцевий пункт може зайняти годину і більше.

Для підключення найбільших з поштових мереж, їхні назви зареєстровані в Internet у якості доменів верхнього рівня: `bitnet`, `fidonet`, `uucp`.

Може скластися уявлення, що поштові мережі підходять тільки для підтримки спілкування між людьми і не можуть використовуватися для доступу до таких ресурсів, як комп'ютери Internet. Однак це не так. Досить широко поширені спеціальні поштові сервери (NEWSmail-, FTPmail-, навіть WWWmail-сервери й ін.), що імітують "людське поводження", тобто вміють одержувати листа і формувати відповіді на них.

Такі сервери здатні правильно інтерпретувати лише листа, написаного на хоча і не складній, але все ж таки формальній мові, що нагадує процес програмування. Крім відносно низької швидкості, це — і трудомістка процедура.

Серед поштових мереж трохи виділяються UUCP-орієнтовані мережі, оскільки більшість з них використовує доменну адресацію в стандарті Internet. З останніми не виникає ніяких проблем, оскільки в Internet забезпечується підтримка технології таких поштових мереж, включаючи транспорт їхніх листів через IP-мережі і навіть забезпечення можливості листування між користувачем такої поштової мережі і користувачем IP-пошти. Проблеми, що виникають, скоріше пов'язані з уживанням декількох кодувань букв у листах українською чи російською мовами.

А як же бути з поштовими мережами, адресація в яких відрізняється від прийнятої в Internet? Для таких мереж використовуються шлюзи. Шлюз — це комп'ютер, здатний одночасно працювати в двох різних мережах і здійснювати необхідне узгодження при обмінах інформацією між ними. У даному випадку мова йде насамперед про шлюзи, здатні перетворити адресу однієї мережі в адресу іншої при спробі листа перейти відповідну міжмережеву межу.

Наприклад, існують шлюзи, що вдало імітують групову роботу користувачів на одному комп'ютері в рамках Internet. Це відбувається в такий спосіб. Скажімо, приходить лист із Internet. Шлюз насамперед виділяє з поштової адреси одержувача його ім'я, через непотрібність відкидаючи все інше. Далі відбувається саме цікаве. Шлюз "дістає" зі своєї пам'яті перелік імен користувачів місцевої мережі і їхніх адрес у ній і перевіряє, чи є в цьому переліку ім'я, витягнуте з листа, що прийшов. Якщо такого не знайшлось, шлюз відсилає листа, що прийшов, по зворотній адресі з позначкою типу "Такого адресату в нас немає". Якщо ж таке ім'я знайшлось, по переліку визначається відповідна місцева адреса, по якій лист відправляється в місцеву мережу. Шлюз забезпечує роботу досить великого числа користувачів місцевої мережі. Описана особливість шлюзів використовується для організації взаємодії і з некомп'ютерними системами, наприклад, з факс-мережею.

Текстові термінали

У рамках Internet існує програма `telnet`, що забезпечує зв'язок комп'ютера користувача з кожним (правда, розрахованим на такого роду контакт) віддаленим комп'ютером. Комп'ютер при встановленні такого зв'язку фактично стає терміналом в оболонці віддаленого комп'ютера (як правило — це Unix, хоча можливі й інші варіанти).

Якщо користувач знайомий з оболонкою, у яку він потрапляє, то він продовжує роботу так (точніше, майже так), якби ця оболонка була б установлена на його власному комп'ютері. Ще задовго до появи сучасних технологічних засобів чимало цікавих інформаційних джерел було створено з розрахунку саме на такий спосіб зв'язку. У ряді випадків джерело, доповнювалося спеціальними засобами підтримки інтерфейсу з користувачем (наприклад, для роботи з великими бібліотечними системами). Сама ж програма `telnet` здатна працювати тільки на комп'ютері IP-мережі.

Комп'ютери багатьох великих вузлів в Internet дозволяють встановлювати з ними зв'язок по телефонній лінії будь-якому комп'ютеру, що не має підключення в яку-

небудь IP-мережу. У першому сеансі звичайно ставиться питання про реєстрацію відповідного користувача на вузлі, що є досить простою процедурою.

Особливість стилю роботи в режимі віддаленого термінала з комп'ютера, не включеного в IP-мережу, полягає в тому, що такий комп'ютер вважається виродженим у термінальний пристрій, що має тільки екран і клавіатуру: процесор і пам'ять як би відсутні і їхню роль виконує вузловий комп'ютер.

Термінальні пристрої, що мають зв'язок з досить могутніми комп'ютерами по виділених лініях зв'язку, інколи називають мережевими станціями, їх активно використовують для застосування в Internet, для чого використовуються термінали з графічними можливостями, що забезпечують повноцінний доступ до Internet.

На будь-якому комп'ютері, що має IP-підключення, теж доступна робота по протоколу `telnet`. І хоча така робота — все ж таки перехід у деяке "ретро", вона в цьому випадку помітно спрощується. Крім того, необхідність використання `telnet` зводиться до організації доступу лише до інформаційних джерел, недоступних іншими засобами.

Звичайно в емуляції віддаленого терміналу доступна й електронна пошта зі стандартною адресацією Internet, що цілком природно, оскільки цю послугу підтримує вузлова машина. Оскільки в Internet такий комп'ютер виступає тільки від свого імені (числової IP-адреси), адресація терміналів є його внутрішньою справою і зводиться, до присвоєння їм послідовних номерів, кожному з який ставиться у відповідність призначена при реєстрації пара параметрів: `Login` — ім'я користувача, `Password` — пароль, що використовуються для контролю при кожному вході в сеанс зв'язку.

3.7. Електронна пошта

Електронна пошта — це комплекс сервісних програм, які служить для передачі та прийому даних у мережах. Ця служба відрізняється від інших служб тим, що комп'ютери, які передають повідомлення, не обов'язково повинні безпосередньо взаємодіяти з комп'ютерами, які приймають повідомлення. Електронна пошта відома як служба "з проміжним накопиченням". Пошта передається від одного комп'ютера до іншого, поки, нарешті, не попаде до пункту призначення.

3.7.1. У яких випадках корисна електронна пошта?

Як і будь-яка інша система сервісних програм, електронна пошта має свої сильні та слабкі сторони. Для того, щоб визначити у яких випадках електронна пошта дійсно корисна, визначимо, чим вона відрізняється від інших середовищ передачі даних.

Електронна пошта дуже схожа і на телефон, і на традиційну пошту. Порівняння цих методів передачі даних наведено в табл. 3.3.

Таблиця 3.3

	Телефон	Електронна пошта	Звичайна пошта
Швидкість	Висока	Помірна	Низька
Синхронізація	Потрібна	Непотрібна	Непотрібна

Таблиця 3.3. Закінчення

	Телефон	Електронна пошта	Звичайна пошта
Формальність	Змінюється	Помірна	Змінюється
Відповідальність	Низька	Помірна	Висока
Конференц-зв'язок	В невеликій групі	"Кожен з усіма"	Тільки односторонній
Захищеність	Середня	Низька	Висока

Проаналізуємо, як швидко в кожному з цих середовищ доставляється повідомлення. Телефонний зв'язок — це миттєва доставка. Час, необхідний для доставки електронної пошти, складає від декількох секунд до одного дня; звичайна пошта приходить в кращому випадку на наступну добу. Ціна, яку необхідно платити за швидкий зв'язок по телефону, полягає в тому, що той, хто визиває і той, кого визивають мають бути синхронізовані, тобто вони повинні знаходитись біля телефону одночасно. Електронна пошта і традиційна пошта — асинхронні системи.

Час доставки електронної пошти складається з двох часових проміжків: часу необхідного мережі для доставки повідомлення в ваш поштовий комп'ютер, і часу від надходження до прочитання одержувачем. Перша частина залежить від способу зв'язку поштового комп'ютера з мережею; її можна змінити тільки за допомогою додаткових грошових витрат. Друга частина знаходиться повністю під контролем користувача.

Результат спілкування в значній мірі залежить від виконання різного роду умов. Розмовляючи по телефону, ви легко знаходите необхідний тон: до одного співрозмовника ви звертаєтесь достатньо офіційно, а до другого по-товариські. Теж саме характерне і для звичайної пошти. Більшість користувачів прагнуть в своїй електронній переписці до неформального спілкування.

Спілкування в письмовій формі передбачає більш високу відповідальність, ніж усне. Якщо в розмові по телефону ви сказали щось таке, про що жалкували б в майбутньому, то в майбутньому ви можете заявити, що нічого такого не говорили або, що ваш співбесідник вас невірно зрозумів. Якщо ж ви будете працювати в електронній пошті, то виявиться, що хтось записав копію такого повідомлення в файл і при потребі може скористатись ним. Єдиний фактор, який може знизити ступінь відповідальності в електронній пошті, полягає в тому, що особа відправника не завжди може бути визначена.

Тепер розглянемо можливості спілкування груп людей. Телефон — чудовий засіб, але тільки для небагаточисельних груп абонентів. Конференц-зв'язок дозволяє спілкуватися один з одним, але в міру зростання чисельності групи планування і настроювання такого зв'язку стають занадто складними задачами. Поштова система, при якій повідомлення виходять з однієї точки і розсилаються всій групі, пов'язана з проблемами, бо доставка кореспонденції від кожного члена групи всім іншим членам — далеко не просте завдання. Електронна пошта дозволяє вільно формувати великі групи, і будь-який член групи може в довільний час здійснювати зв'язок з іншими її членами. Це робить електронну пошту корисною як для розповсюдження інформації, так і для опитування групи.

І нарешті, захищеність електронної пошти, як правило, нижче, в порівнянні з іншими засобами передачі даних. Якщо дотримуватися безпеки, то лист, що відправлений звичайною поштою, може залишатись в закритому ящику поштового відділення до тих пір, доки не потрапить до адресата. Якщо в процесі доставки його хтонебудь відкривав, то пошкодження, як правило, робить цей факт очевидним. Для підслуховування телефонних розмов необхідний доступ до засобів зв'язку на одному з кінців лінії, а це не для всіх можливо. Електронна пошта ж йде по досить непередбачуваному маршруту, через різні комп'ютери, захищеність яких може бути і недостатньо високою. Крім того, існують випадки, коли через помилку повідомлення неможливо доставити, і комп'ютер, не знаючи, що робити, доставляє його адміністратору пошти. А потрапляння вашої пошти в чужі руки небажане. Існують спеціальні програми підвищення захисту, які кодують повідомлення, але вони використовуються не багатьма.

3.7.2. Поради по веденню електронної кореспонденції

Наведемо деякі корисні поради для роботи с електронною поштою:

- ніколи не довіряйте електронній пошті те, що ви не хотіли б зробити загальновідомим;
- не відправляйте образливих повідомлень і таких, що містять погрози;
- відправники часто розглядають електронну пошту як товариську бесіду, в той час як одержувачі часто сприймають повідомлення як діловий лист;
- будьте обережні з висловлюваннями, що допускають різне тлумачення;
- якщо ви хочете, щоб ваше повідомлення відображалося на терміналі практично любого типу, встановлюйте стандартну довжину рядка (менше 60 символів);
- використовуйте обидва регістри літер;
- не потрібно використовувати різні екзотичні можливості термінала (різний шрифт, курсив і т. ін.).

3.7.3. Про те, як працює електронна пошта

Якщо повідомлення адресовано вами вірно, то мережа доставить його по призначенню. Повідомлення можуть переміщуватись між Internet і іншими мережами електронної пошти, але для того, щоб потрапити в іншу мережу і "пройти" по ній, повідомлення повинно мати більш складну адресу.

Пункти з'єднання між мережами електронної пошти — це комп'ютери, які називаються "прикладними шлюзами" (прикладними, бо їх "знань" про прикладні програми електронної пошти, що знаходяться по обох сторонах, достатньо, щоб переформатувати повідомлення у відповідності з вимогами мереж призначення). Для пересилки пошти через шлюз часто доводиться давати адресу, що містить інформацію про те, як добратися до шлюзу, і про те, як доставити пошту на інший бік.

Перед тим як відправити листа засобами електронної пошті, необхідно дати йому заголовок. Заголовок включає в себе позиції **To:** (кому), **From:** (від кого), **Subject:** (тема), що стоять на початку повідомлення. В електронні пошті заголовок повідомлення по дорозі вбирає в себе інформацію про те, як подорожувало повідомлення, щоб у випадку його повернення можна було визначити маршрут.

➢ Правила адресації в Internet розглядались в попередньому пункті цього ж розділу.

3.7.4. Накопичення адрес електронної пошти

Після того, як ви почнете користуватися електронною поштою, ви зіштовхнетеся з проблемою пошуку адрес електронної пошти. Національного реєстру таких адрес не існує. Є декілька спеціалізованих серверів, до яких можна звертатися з метою пошуку невідомої адреси. Ці сервери відомі як сервери "білих сторінок". Але найбільш простий і ефективний спосіб отримання необхідної адреси — взяти її із інформації, яка надіслана безпосередньо вам.

Цей метод отримання адрес електронної пошти має переваги. Він гарантує, що ви отримаєте поточну адресу електронної пошти, яка регулярно перевіряється. Адреса ж, знайдена у вказівнику, може бути застарілою, або такою, що більше не існує.

Іноді при спробі вилучити адресу електронної пошти з пошти, що надійшла, ви можете побачити наступну форму: `John Bigboote<johnb@yoyodyne.com>`. Формат цієї адреси більш складний: `коментарі<адреса_електронної_пошти>`.

Якщо ви отримуєте повідомлення по списку розсилки, і адміністратор списку включив у нього коментарі, то, подивившись у поле **To:**, ви легко зможете побачити, кому, крім вас, воно було направлене, навіть якщо адреси електронної пошти розпізнати неможливо.

3.7.5. Програма електронної пошти ОС UNIX

Для запуску програми `mail` виконайте наступну команду:

```
%mail рядок_адрес
```

`Рядок_адрес` — необов'язковий параметр. Якщо він присутній, то команда посилає повідомлення особам, адреса яких вказана у рядку. Якщо рядок адрес пропущений, то відбувається перехід до командного режиму, в якому ви можете, наприклад, читати повідомлення, що надходять.

Читання пошти, що надходить

Для читання, отриманої пошти ввійдемо в командний режим. Якщо пошти на ваше ім'я немає, то програма повідомить:

```
% mail
No mail for krol
```

(Для користувача `krol` пошти немає).

Якщо в поштовій скринці вас очікують повідомлення, то команда `mail` видає перелік заголовків 20 перших повідомлень:

```
%mail
"/usr/spool/mail/krol": 5 messages 1 new
1 LISTSERV@bitni Fri Nov 8 16:02 128/6172
"File: "LISTSERV FILEI"
2 LISTSERV@bitni Fri Nov 8 16:08 164/9834
"File: "BITNODE FILELIS"
U 3 daemon@pit-man Sat Nov 9 09:26 72/2817
```

```
"Reply from mserv re: s"
U 4 akida Sat Dec 28 05:53 12/298 "Overthruster found"
>N 5 buckaroo Thu Jan 2 19:15 11/305 "Aliens in Grovers Mill"
&
```

Кожне повідомлення має статус і номер. Статус відмічається буквою (або відсутністю букви) на початку кожного рядка. Букви можуть бути наступними:

- **N** — нове повідомлення, отримане після вашого останнього входження в командний режим для читання повідомлення;
- **P** — повідомлення, яке ви прочитали і вирішили залишити в своїй корзині для паперів, що надходять на даному сеансі роботи з програмою;
- **U** — непрочитане повідомлення. Нові повідомлення перетворюються в непрочитані, якщо ви виходите з програми, не прочитавши їх;
- Без букви — повідомлення було прочитане і збережене на попередньому сеансі роботи з програмою.

Номер повідомлення використовується в різних командах для посилання на дане конкретне повідомлення. На одне з повідомлень вказує знак ">" ("більше"). Це поточне повідомлення. Якщо ви дасте команду, не вказуючи номеру повідомлення, то вона буде виконана для поточного повідомлення.

Знак "&" після списку повідомлень — це запрошення до введення команди програми `mail`. Для читання електронної пошти необхідно знати всього чотири команди. Вони, як правило, мають вигляд однієї літери, але є декілька команд, імена яких подовжені для запобігання співпадання.

Команда p

Для читання повідомлень використовується команда виводу `p` (від англ. "print"), яка має наступний формат:

```
& p повідомлення
```

Параметр `повідомлення` — необов'язковий. Якщо його випустити, то на екран буде виведене поточне повідомлення. Цей параметр може приймати одну з наступних форм:

- **& p 3** — вивести повідомлення номер 3;
- **& p 3-5** — вивести повідомлення з номерами з 3 по 5;
- **& p $** — вивести останнє повідомлення;
- **& p 3-$** — вивести повідомлення з номера 3 до останнього.

Команда виводу також є командою, що виконується по замовчуванню. Тому наступні команди ідентичні:

```
&3
&p3
```

Таким чином, результатом натиснення клавіши <Enter> без команди буде виведення на екран поточного повідомлення.

Команда f

Команда швидкого виводу `f` (від англ. "fast print") видає "зміст" кошику вхідної кореспонденції у вигляді меню, подібно до того, яке ви отримуєте при запуску про-

грами mail. У вихідному списку наведені 20 нових повідомлень, починаючи з самого старого. Якщо у вас більше повідомлень, то можна пересуватись в меню шляхом введення додаткових команд f:

- & f 1-20 — вивести адресу відправника і тему з 1 по 20;
- & f 21-& — вивести адресу відправника і тему з 21 по останню.

Команда z

Список повідомлень можна "прокручувати" за допомогою команди z, яка видає на екран наступну порцію списку. Для пересування по списку в зворотному напрямку використовується команда z-.

Інші команди

Якщо нічого особливого ви не робите, то при виході з програми всі повідомлення, які ви прочитали в цьому сеансі, переміщуються із кошика для кореспонденції, що надходить, в файл mbox. Альтернативою слугують наступні команди:

- d повідомлення — видалити вказані повідомлення (чи поточне повідомлення, якщо повідомлення не вказані). Ця команда видаляє номери вказаних повідомлень із меню, а в кінці сеансу роботи з електронною поштою знищує і самі повідомлення.
- pre повідомлення — зберегти вказані повідомлення (чи поточне повідомлення, якщо повідомлення не вказані). Це означає: зберігати дане повідомлення в кошику для повідомлень, що надходять, звідки їх можна читати в наступних сеансах.
- q — вийти із програми.

Відправка повідомлення

Відправити повідомлення можна двома способами:

- ввести команду запуску програми mail із переліком адрес;
- ввести команду mail в програмі mail (тобто, після запрошення &). Обидві команди мають однаковий синтаксис: % mail список_адрес.

Список адрес може складатись з однієї чи декількох адрес, розділених комами.

Після цього програма mail запитає тему повідомлення. Введіть в якості теми який-небудь вислів, що має зміст. Цей вислів і ваше ім'я — це все, на основі чого одержувач вирішує, який пріоритет присвоїти цьому повідомленню. Після вводу теми починайте набирати текст повідомлення. Кінцем повідомлення для програми mail служить крапка на першій позиції останнього рядка. Після завершення введення повідомлення воно відправляється, а ви повертаєтесь до того, що робили до подачі цієї команди.

Набирати повідомлення таким чином дуже незручно, особливо якщо повідомлення довге. У цьому випадку важко виправляти помилки. Тому для складання повідомлень можна використовувати який-небудь редактор, наприклад, vi. Він викликається командою ~v, причому "~" (тильда) повинна знаходитись на першій позиції екрану. Відредагуйте повідомлення за допомогою команд цього редактора, потім запишіть його і вийдіть з редактора. Поверніться до програми mail, поставте у першому стовпчику останнього рядка крапку для завершення повідомлення і його передачі.

Якщо редактор `vi` вам не подобається, то вийдіть з програми `mail` і виконайте наступну UNIX-команду:

```
%setenv EDITOR мій_улюблений_редактор
```

де `мій_улюблений_редактор` — ім'я редактора, яким би ви хотіли скористуватись.

3.7.6. Можливості систем електронної пошти

Загальні можливості

Робота з псевдонімами

Дуже корисна можливість присвоювати користувачам псевдоніми. Якщо вам не подобається набирати повні Internet-адреси, то можна, наприклад, адресу `krol@uxl.cso.uiuc.edu` замінити скороченням `edk`. Якщо потім використовувати `edk` в якості імені одержувача, то ваша система підставить замість нього повну адресу.

Папки

Папки дозволяють зберігати повідомлення в систематизованому вигляді. Наприклад, у вас може бути папка для кожного з проектів, у яких ви приймаєте участь, і одна папка з назвою "Особисте". По мірі надходження пошти можна розміщувати повідомлення у відповідні папки для подальшої їх обробки.

Пересилка пошти, що надійшла

У програмі ОС UNIX термін "пересилка" має два різних значення. По-перше, пересилка означає автоматичну пересилку вхідної пошти на конкретне ім'я користувача на одному комп'ютері, на інший комп'ютер. Це особливо корисно, якщо у вас є доступ до декількох різних комп'ютерів. Для того, щоб не потрібно було постійно перевіряти пошту на декількох комп'ютерах, можна зробити так, що вся пошта, що надходить на всі ваші імена, пересилалась на ту систему, в якій, як правило, ви її читаєте.

По-друге, "пересилка" означає пересилку отриманого вами повідомлення всім, кого воно може зацікавити.

Списки розсилки

В електронній пошті відправити повідомлення групі людей так же легко, як і окремій людині. Засіб, який дозволяє це зробити, називається ***списком розсилки***. Він дає можливість позначити групу одержувачів псевдонімом. Наприклад, псевдонім `staff` можна визначити як "всі співробітники". При пересиланні пошти на ім'я `staff` вона доставляється всім членам цієї групи.

Відповідь

Ця функція дозволяє повідомити вашій програмі електронної пошти про те, що ви хочете відповісти людині, яка надіслала вам повідомлення. Вона дозволяє не вводити адресу електронної пошти. Програма електронної пошти, як правило, копіює поле **From:** (чи **Reply-To**) з вихідного повідомлення при створенні рядка **To:** нового

повідомлення; а для створення нового рядка **Subject:** програма просто копіює оригінал і добавляє на початок "Re:", вказуючи на те, що це — відповідь на повідомлення.

Розширені можливості

Відправлення копій

Всі програми електронної пошти дозволяють розміщувати в полі **To:** заголовку повідомлення декілька адрес. Але часто буває необхідним розрізняти тих користувачів, кому це повідомлення адресоване в першу чергу, і тих, хто отримує їх "до відома".

"Сліпі" копії

Розсипка "сліпих" копій проводиться по списку адресатів, як і розсипка звичайних копій. Але рядок заголовку, в якому перераховані одержувачі з вхідної кореспонденції, автоматично видаляється. В результаті, жоден адресат не знає, хто ще отримав надіслані таким чином повідомлення.

Файли підпису

Файл підпису — це спосіб введення додаткової інформації у повідомлення вихідної електронної пошти. Він часто використовується для представлення інформації про те, хто ви такий і як з вами зв'язатися.

Нестандартні можливості

Приєднання документів

Документи, що приєднуються, обслуговує засіб, що називається "мультимедіапоштою". Це розширення дозволяє відправляти у складі повідомлення разом з іншими приєднаними документами (наприклад, з двійковими даними) оцифровані звук і відеозображення. У мережі Internet існує новий стандартний метод реалізації цього розширення — MIME.

Повідомлення про отримання

Полягає в автоматичному відправленні повідомлення після того, як відправлена вами пошта вміщена в електронний поштовий ящик одержувача.

Повідомлення про прочитання

Полягає в автоматичному відправленні вам повідомлення після того, як відправлена вами пошта виведена одержувачем на екран його дисплею. Воно не означає, що адресат прочитав чи зрозумів ваше повідомлення. Повідомлення про прочитання допоможе вам викрити одержувача, який стверджує, що він прочитав повідомлення тільки сьогодні вранці, коли ви знаєте, що повідомлення надійшло на екран його комп'ютера значно раніше.

Відміна повідомлення

Цей засіб дозволяє забирати повідомлення назад після того, як воно відіслане. Така можливість зручна, якщо вам часто трапляється відсилати повідомлення, про

які пізніше ви починаєте жалкувати. На протязі деякого обмеженого інтервалу часу ви можете відмінити повідомлення. Тривалість цього інтервалу змінюється в залежності від пункту призначення повідомлення і системи підключення програм електронної пошти.

Якщо повідомлення відправляється іншому користувачу в межах цієї ж системи електронної пошти, то його, як правило, можна відмінити, поки воно не прочитане. Якщо воно адресоване в кошик для вхідної кореспонденції в іншій системі електронної пошти, то його можна відізвати лише до того моменту, коли воно залишить межі системи відправника.

3.7.7. Розширення MIME

MIME (Multi-purpose Internet Mail Extensions) — це специфікація процедур автоматичного відправлення нетекстових об'єктів у повідомленнях електронної пошти.

Хоча MIME являє собою специфікацію, яка служить головним чином для приєднання файлів до повідомлень електронної пошти, для більшості користувачів MIME пов'язана з мультимедіа. MIME-програми, як правило, знають як обробляти різнотипні файли, що приєднуються, зокрема, відображення, звукозаписи і фільми. (Здатність відтворювати все це залежить від наявності у вашому комп'ютері як відповідних апаратних, так і програмних засобів).

Так, можна відправити повідомлення з описом нової іграшки з дистанційним управлінням, яку ви випускаєте на ринок, приєднавши до цього повідомлення відеофайл, в якому показано, як іграшка ганяє собаку по кімнаті. Одержувач зможе прочитати це повідомлення і продивитись відеофільм, не роблячи ніяких додаткових дій. (Звичайно, сам факт отримання гігантського файлу може не викликати великого задоволення у одержувача, особливо якщо він платить за час з'єднання).

В самих екзотичних варіантах деякі MIME-повідомлення можуть навіть містити програмне забезпечення, яке буде виконуватись у вашій системі. Наприклад, можна отримати якесь повідомлення і після того, як ви його прочитаєте, отримане ПЗ автоматично створює новий екран, на який виведеться зразок листа і напис "Якщо ви хочете відправити цей лист по електронній пошті, то зробіть тут помітку". Лист складається, підписується і відправляється автоматично.

Хоча дозволяти комусь сторонньому виконувати програми в своїй системі досить небезпечно, але тут реалізований дуже розумний інженерний підхід: ці програмні засоби написані на спеціальній мові, яка зрозуміла вашій MIME-програмі електронної пошти, і їм не дозволяється зробити нічого такого, що могло б пошкодити вашій системі.

Ще одна особливість MIME: ця система допомагає запобігти відправку файлу, якщо цього не хоче одержувач. Замість того, щоб розміщувати файл у повідомленні електронної пошти у вигляді закодованої вставки, в повідомлення вводиться директива, що повідомляє програмі електронної пошти одержувача про те, як автоматично отримати цей файл по протоколу FTP. Цей засіб дає можливість вирішити, хочете ви отримати даний файл чи ні. Якщо ні, то це дозволить зекономити місце на диску і зменшити трафік у мережі.

Наприклад, ви відправляєте по електронній пошті проект нового керівництва користувача всім співробітникам вашої компанії. Насправді цей документ не відправ-

ляється; він знаходиться в архіві FTP, і будь-який з адресатів може запросити його у зручний для себе час. Причому бажаючі отримати дане керівництво користувача можуть зробити це, навіть не знаючи, як працювати з програмою FTP — їх MIME-програма електронної пошти сама виконає всі операції FTP-пересилки. Ті ж, кому цей документ непотрібний, можуть просто ігнорувати його наявність.

3.7.8. Pine — реалізація MIME

Існує декілька програмних пакетів, які можуть обробляти повідомлення MIME. Одним з таких пакетів є `metamail` (безкоштовні розширення стандартної програми електронної пошти ОС BCD). Другий пакет — `Z-Mail` (популярна комерційна програма електронної пошти). Ще одним продуктом в області електронної пошти, що розповсюджується безкоштовно, є `pine`. Простота цієї програми і її орієнтація на меню дозволяє використовувати її в більшості випадків досить ефективно.

Запускається `pine` наступною командою:

```
% pine
```

Після цього на екран виводиться перелік основних команд:

```
PINE 3.05 MAIN MENU Folder; inbox 1 Message
? HELP - Get help using Pine
```

(Допомога у використанні Pine)

```
C Compose  — Compose and send a message
```

(Скласти і відправити повідомлення)

```
 I MAIL INDEX - Read mail in current folder
```

(Прочитати пошту в поточній папці)

```
F FOLDERS - Open a different mail folder
```

(Відкрити іншу папку пошти)

```
A ADDRESSES - Update yuor adress book
```

(Поновити книгу адрес)

```
O OTHER - Use other function
```

(Використати інші функції)

```
Q QUIT - Exit the Pine mail program
```

(Вийти з програми)

Всі команди реалізуються однократним натисненням клавіш. Для передачі програми, наприклад, `overthruster`, користувачу `johnb@yoyodyne.com` треба натиснути `C`. Після цього MIME видає шаблон вводу даних для електронної пошти:

```
PINE 3.05 COMPOSE MESSAGE Folderinbox 1 Message
To: johnb@yoyodyne.com
Cc:
Attchmnt: l./cso/staff/krol/overthruster (185 KB)
Subject: Program ———Message Text———
Here is the program you wanted.
```

Курсор стоїть у полі `To:`. Набираємо адресу пункту призначення, а потім при допомозі табулятора переміщуємося на інші поля. Дійшовши до поля `Attachment:`, вводимо ім'я файлу, що буде відправлятись (у даному випадку це `overthmster`). Після вводу імені файлу програма сама дописує наступну частину шляху, розмір

файлу і всю іншу інформацію. Після цього вводиться тема і текст повідомлення. Далі натискається комбінація клавіш <Ctrl+X>, і повідомлення відправляється.

3.7.9. Обробка пошти, що повернулась

Якщо електронна пошта не може бути доставлена, то відправнику, як правило, видається повідомлення із зазначенням причини.

Найчастіше електронна пошта повертається по одній із наступних причин:

- система не може знайти комп'ютер одержувача;
- одержувач на вказаному комп'ютері невідомий;
- пошта може знайти комп'ютер, але все одно не можна доставити повідомлення.

Невідомі комп'ютери

Якщо програма електронної пошти прислала повідомлення, тема якого має вигляд "Subject: Returned mail: Host unknown", значить причиною повернення є невідомий комп'ютер-адресат. Якщо виникла подібна ситуація, то:

- по-перше, перевірте адресу: чи вірно написане ім'я;
- по-друге, перевірте повноту адреси. Якщо вказане неповне ім'я, то більшість систем автоматично додають доменний суфікс, припускаючи, що він такий же, як і у них. Наприклад, якщо ви написали `yoyodyne` замість `yoyodyne.com`, то на комп'ютері `ux1.cso.uiuc.edu` ця адреса перетвориться у `yoyodyne.cso.uiuc.edu`. Тому будьте уважні при відправці повідомлень і не забувайте вказувати повну доменну адресу.

Невідомі одержувачі

Якщо програма електронної пошти прислала повідомлення, тема якого має вигляд "Subject: Returned mail: User unknown", значить причиною повернення є невідомий комп'ютеру вказаний адресат. У цьому випадку слід перевірити правильність набраного імені користувача в адресі. Але може трапитись і так, що ім'я користувача набране вірно, а ім'я комп'ютера невірно, але воно допустиме. Наприклад, якщо ви помилково заадресували повідомлення `johnb@ux2`, а не `john@ux1`, то можете отримати повідомлення "Користувач невідомий", бо комп'ютер `ux2` існує, а користувача `johnb` на ньому немає.

Пошту неможливо доставити

Інколи можна зустріти повідомлення "...Service unavailable". Це повідомлення означає, що, не зважаючи на те, що комп'ютер знайдений і зв'язок з ним встановлений, в даний момент він пошту не приймає. В цьому випадку слід трошки зачекати і спробувати надіслати пошту знову дещо пізніше. Причина в тому, що деякі системи налаштовані так, що вони не приймають пошту у вихідні дні чи на протязі інших, спеціально відведених проміжках часу.

Може трапитись також така ситуація, коли проблема залишається невиявленою декілька днів: комп'ютер відомий, але не є доступним. В таких випадках комп'ютер-відправник може намагатися відправити пошту на протязі двох-трьох днів (а то й бі-

льше) і лише потім повідомить вам про невдачу. В цьому випадку тема поверненого повідомлення буде мати вигляд "Subject: Returned mail: Cannot send message for 2 days".

Причин тут може бути декілька:

- несправність в мережі, що робить контакт з віддаленою системою неможливим;
- віддалена система може бути "недієздатною" внаслідок серйозних апаратних проблем;
- конфігурація віддаленої системи може бути порушеною (нерідко трапляються випадки, коли хтось змінює свою конфігурацію і забуває повідомити про це адміністратора мережі).

3.7.10. Адміністратор e-mail

Для кожного комп'ютера, що обмінюється поштою, визначена поштова скринька з іменем `postmaster`. Пошта, яка адресується на цю скриньку, повинна читатися адміністратором електронної пошти. Якщо у вас виникли проблеми з яким-небудь комп'ютером, то можна відправити запит на адресу:

`postmaster@ім'я_комп'ютера`

При відправленні повідомлення в `postmaster` ви можете розраховувати на:

- допомогу в пошуку адреси електронної пошти одержувача, який, як ви знаєте, користується цим комп'ютером;
- допомогу у пошуку відповідного шлюзу для відправлення електронної пошти у зовнішні мережі;
- співпрацю у боротьбі з чиїмось недозволеними діями на цьому комп'ютері.

3.8. Інформаційні ресурси в Internet

3.8.1. Засоби пошуку

У загальному випадку засоби пошуку в Internet розподіляють на основні та додаткові. До основних засобів можна віднести:

- пошукові машини (*search engines*) — універсальні та спеціалізовані;
- пошукові каталоги (*index sites*);
- портали або мультипортали (*search engines* + *index sites*);
- довідникові системи ("жовті" та "білі" сторінки);
- багатопошукові системи (*meta reference sites*) — Metasearch tools і Multisearch tools;
- PPCS (Pay-Per-Click Systems) та каталоги пошукових систем.

До додаткових засобів пошуку відносять: Web-board, USNet, Chat, ICQ, IRC, FTP, Live Journal, тематичні колекції посилань, енциклопедії тощо.

Для вирішення стандартних пошукових задач мережа має базові засоби. Ці основні засоби пошуку працюють автоматично, тому відповідають на запит досить швидко та можуть обробляти багато запитів від одного абонента.

Додаткові способи пошуку використовують можливості, що надаються іншими службами мережі, її персоналом, а також її абонентами, і можуть якось сприяти пошуку. Ці способи є додатковими, тому що вони:

- або не призначені для масового використання;
- або не є універсальними (накопичують адреси у недостатньому об'ємі чи у вузькому напрямку);
- або не є стандартними чи обов'язковими для того, хто їх пропонує.

Характерним прикладом такого додаткового способу пошуку є публікація запиту у відповідній телеконференції — є деяка ймовірність, що хто-небудь відгукнеться, але можуть і промовчати.

Якщо хтось має бажання, щоб його змогли знайти за допомогою пошукових служб Internet, то йому слід не забувати залишати інформацію про себе на пошукових серверах, що пропонують таку можливість. Більшість white- та yellow- систем пропонують "зареєструватись", тобто заповнити форму своїми координатами. Найчастіше реєстрація не є обов'язковою, хоча бувають і виключення. Досить часто реєстрація дозволяє користуватися додатковими можливостями пошуку. Нижче буде наведений список серверів, на яких можна залишити "слід".

White- та Yellow-пошук

Формальні відмінності

Для подальшого викладення будемо розрізняти white-пошук та yellow-пошук. Під ***white-пошуком*** ("білим" пошуком) розуміють пошук адреси одного конкретного адресата за його досить визначеним ім'ям (людини — за прізвищем, організації — за назвою). "Досить визначене ім'я" означає, що об'єкт пошуку наперед відомий: є впевненість в його існуванні та ім'я, що пропонується для пошуку, достатньо унікальне. Тому є деяка впевненість в тому, що дані будуть знайдені на першому ж кроці пошуку.

"Білі сторінки" в звичайних телефонних довідниках — це просто перелік людей або організацій за алфавітом. Якщо перший крок пошуку невдалий, то виникає припущення, що у вихідних даних помилка, і можна спробувати yellow-пошук.

Yellow-пошук ("жовтий" пошук) важливий, перш за все, сам по собі, а не як продовження white-пошуку. Він дозволяє шукати адресатів не тільки і не стільки за їх власним ім'ям, а й за іншими ознаками, що охоплені тією чи іншою класифікацією. Формально кажучи, ***yellow-пошук*** — це пошук адреси одного чи декількох адресатів за недостатньо визначеним чи зовсім не визначеним ім'ям.

"Недостатньо визначене ім'я" означає неповне, неоднозначне чи в чомусь невірне ім'я, яке не дозволяє знайти адресата засобами white-пошуку. "Невизначене ім'я" означає або просто відсутність будь-якого конкретного імені, або таке його значення, що дозволяє лише віднести адресата до якоїсь досить широкої групи (наприклад, задача знайти адресу "деякої лікарні в США").

Стисло сформулювати різницю між цими двома видами пошуку можна так:

- white-пошук — адрес конкретний, наперед відомий. Пошук адресної інформації за одною, але досить важною ознакою — за власним ім'ям адресата. Ця ознака — первинна, персоніфікуюча. У людини або організації можуть зміни-

тися поштова адреса, номер телефону, але власне ім'я залишається незмінним.

- Yellow-пошук — адреса адресата або групи адресатів наперед невідома. Спершу розшукується власне ім'я (імена) за деякими додатковими ознаками (за родом діяльності, за географічними ознаками), а після цього за вже знайденим ім'ям можна знайти адресну інформацію.

У більшості випадків системи "жовтих сторінок" включають у себе і "білі сторінки" — у знайденого адресату можна побачити його телефон та поштову адресу тощо. Крім того, деякі "жовті сторінки" дозволяють шукати безпосередньо в алфавітному списку своїх абонентів (white-пошук). З іншої сторони, "білі сторінки" також містять елементи yellow-пошуку — крім власного імені вони частіш за все дозволяють вказати назву міста, країни та інші дані, що звужують пошук. Можливо, саме тому багато "on-line" телефонних довідників, що виконують фактично white-пошук, хоча називають себе "жовтими сторінками".

Yellow-пошук за ієрархічним класифікатором та за ключовими словами

Перед занесенням до свого каталогу, система "жовтих сторінок" пропонує адресату заповнити картку, де окрім полів з адресною інформацією йому потрібно заповнити поля, що відповідають тим ознакам, за якими класифікований каталог (наприклад, поле "вид діяльності").

При цьому можливі два підходи:

- адресат може вибрати готове значення чинника із списку, що пропонується (наприклад, з класифікатора галузей);
- адресат може написати цю ознаку довільно, своїми словами.

У першому випадку пошук буде проводитися за тим же самим класифікатором, у другому — за "ключовими словами", які будуть порівнюватися (необов'язково до повного співпадіння) з тими, якими охарактеризував себе адресат. Пошук за ієрархічним класифікатором проходить до точного співпадіння зразка з яким-небудь значенням, що міститься там. Пошук по ключовим словам може проводитися до різного ступеня співпадіння.

Найчастіше умова співпадіння буває "м'якою" — достатньо для того, щоб запропонований зразок виявився фрагментом у тексті, що порівнюється, однак іноді пошукові системи дозволяють посилювати умову. Найчастіше пошук за ознакою "галузь промисловості" здійснюється в ієрархічному класифікаторі (або в стандартному, що використовується статистичними службами, або в "саморобному", що менший за розмірами, але більш зручний), а пошук за "продукцією, що виготовляється" проходить за ключовими словами, хоча реально для цього використовується теж ієрархічний класифікатор (у більшості випадків — стандартний кодифікатор продукції та товарів).

На відміну від "жовтих сторінок" організацій, які можуть бути зібрані без опиту самих адресатів (за галузевими списками або за списками статистичних органів), "жовті сторінки" приватних осіб малочислені та не є вичерпними. Внесення до такої бази даних є суто добровільною справою, при заповненні картки пропонуються на вибір поля типу "місце роботи", "професія", "хоббі та інтереси", "місце навчання" і т.ін.

Зворотній white- та yellow-пошук

Деякі телефонні "білі сторінки" мають можливість зворотного пошуку — знаходження адресата за вказаним телефонним номером. "Жовті сторінки" також мають схожу можливість: для адресата, знайденого за ключовими словами або за ієрархією значень деякої ознаки, вказується і значення ознак з інших ієрархій (знайшовши, наприклад, виробника програмного забезпечення, можна поряд побачити, що це — компанія-розробчик ПЗ).

Пошук адрес електронної пошти

Задачі, засоби та способи пошуку адрес e-mail

Основні задачі:

- white-пошук адреси конкретної людини;
- white-пошук адреси конкретної організації;
- white-пошук адреси у конкретній організації;
- white-пошук адреси вузла мережі;
- зворотній white-пошук абонента за адресою;
- yellow-пошук адреси організації;
- окремий випадок: пошук e-mail адрес учбових закладів та їх студентів.

Для пошуку адрес e-mail людей, організацій та вузлів мережі спеціально призначені довідкова служба Netfind (доступна в режимі "on-line") та автоматичні адресні сервери типу Whois (доступні і в "on-line" і по e-mail). Свої бази адрес вони заповнюють автоматично, неперервно та в масовому порядку прямо з баз адрес великих вузлів мережі (або шукають прямо в них), які містять досить повний набір записів про своїх абонентів.

Основні засоби пошуку:

- автоматична довідкова служба Netfind;
- автоматичні адресні сервери типу Whois;
- автоматичні адресні сервери інших типів;
- довідникові системи "Жовті сторінки", що містять і адреси e-mail.

Додаткові способи пошуку:

- спеціальні запити до автоматичних серверів деякої масової служби (телеконференції чи списку розсилки) або одержання деяким іншим шляхом списку її абонентів;
- запит до абонентів деякої масової служби;
- запит до персоналу вузла мережі;
- запит до автоматичного адресного серверу або до персоналу деякої організації, що надає публічний доступ до адрес своїх кореспондентів (клієнтів, партнерів і т.ін.).

Довідкова служба Netfind

Netfind за одним запитом виконує white-пошук у багатьох адресних базах даних та в інших довідкових службах. Для використання цього сервісу на своєму комп'ютері

потрібно мати клієнтську програму Netfind, при її відсутності можна використовувати telnet (logname "netfind") на будь-який з Netfind-серверів у світі:

```
archie.au                    mudhoney.micro.umn.edu
bruno.cs.colorado.edu        netfind.oc.com
dino.conicit.ve              netfind.vslib.cz
ds.internic.net              nic.nm.kr
lincoln.technet.sg           nic.uakom.sk
macs.ee.mcgill.ca            redmont.cis.uab.edu
malloco.ing.puc.cl           monolith.cc.ic.ac.uk
```

Крім того, Netfind доступна і через Web-сервери (URL-адреси):

```
http://www.internic.net
http://www.nova.edu/Inter-Links
```

Whois- сервери

Whois-сервери виконують white-пошук у базах адрес великих вузлів або цілих мереж. Робота з Whois-сервером здійснюється за визначеними правилами (whois-протокол), тому бажано користуватися клієнтською програмою (whois-клієнт). В UNIX така програма так і зветься: `whois`. Їй можна задавати адресу будь-якого з whois-серверів для пошуку на ньому.

Whois-сервери доступні також і через telnet:

```
http://www.internic.net
telnet://nic.ddn.mil
mailto:service@nic.ddn.mil
telnet://rs.internic.net
mailto:mailserv@internic.net
telnet://whois.internic.net
```

"Головний" whois-сервер Internet, що містить адреси людей та вузлів:

```
telnet://whois.ripe.net
```

Whois-пошук адрес e-mail, телефонні довідники:

```
http://aalaxy.einet.net/hytelnet/DIROOO.html
ftp://sipb.mit.edu/pub/whois/whois-servers.list
ftp://rtfm.mit.edu/pub/whois/whois-servers.list
```

Список відомих whois-серверів:

```
http://www.nova.edu/Inter-Links/cgi-bin/whois.pl
```

Зворотній пошук за e-mail адресою (Finger)

Пошукова служба Finger дозволяє знайти конкретну людину за його повною адресою e-mail. Її можна використовувати, знаходячись в операційній системі UNIX, та шукати абонентів на інших UNIX-машинах (в більшості, Internet складається саме з них). Finger може показати реєстраційне ім'я абонента на віддаленій машині (*login name*), деякі важливі дані про себе, якщо він їх вказав (назва організації, де він працює, телефон), а також час останнього читання електронної пошти (так можна визначити, чи користується він ще своєю адресою e-mail). Багато UNIX-машин мають Finger-сервер, що приймає запити не лише про конкретні e-mail адреси, але й дозволяють зробити пошук серед своїх користувачів.

```
http://www.nova.edu/Inter-Links/CQi-bin/finger.pl
```

Довідкова служба X.500

Глобальна довідкова служба X.500 об'єднує локальні довідкові служби, що мають вигляд одного ієрархічного довідника. Для доступу потрібна спеціальна програма (локальний X.500-клієнт), однак, якщо її нема, то можна скористатися будь-яким з загальнодоступних віддалених X.500 клієнтів, доступ до яких можливий через деякі Gopher- та Web-сервери і навіть через e-mail:

http://www.dante.net:8888/

telnet://ashe.cs.tcd.de de — Німеччина

telnet://chico.rediris.es directorio — Іспанія

telnet://dir.ulcc.ac.uk dua — Англія

telnet://elcl.mat.torun.edu.pl de — Польща

telnet://elem4.vub.ac.be dua — Бельгія

telnet://hypatia.umdc.umu.se de — Швеція

telnet://jethro.ucc.su.oz.au fred — Австралія

telnet://jolly.nis.garr.it de — Італія

telnet://login.dkuug.dk ds — Данія

telnet://nic.funet.fi dua — Фінляндія

telnet://nic.switch.ch dua — Швейцарія

telnet://paradise.ulcc.ac.uk dua — Англія (проект Paradise)

telnet://x500.denet.dk de — Данія

telnet://x500.ieunet.ie de — Ірландія

telnet://x500.tu-chemnitz.de x500 — Німеччина

telnet://zoek.nic.surfnet.nl zoek — Нідерланди

mailto:Directorv@uninett.no

Інші автоматичні адресні сервери і служби

telnet://regulus.cs.bucknell.edu 185

mail-to : mail-server® rtfm.rnit.eduHELP

Сервер дозволяє знайти адреси людей, що посилали повідомлення в конференції USENET.

http://okra.ucr.edu/okra/

"OKRA: net.citizen Directory Service" — один з найбільших серверів "білих сторінок" в Internet з 5.3 мільйонами адрес e-mail. Пошук може відбуватись за адресою e-mail, іменем, організацією і за іменем хосту.

http://www.benoit.com/jean/find.html

http://www.curtin.edu.au/search/people.html — пошук людей.

http://sunsite.unc.edu:80/~masha/ — пошук e-mail адрес людей.

http://www.law.flinders.edu.au/othsites.htm — пошук людей і географічних точок.

mailto:whoiswho@kiae.su — адресний сервер вузла kiae. Містить адреси з бази вузла kiae.

mailto:ais@relarn.net.kiae.su — адресний сервер РосНИИРОС.

mailto:intermap@botik.ru — сервер Intermap. Накопичує адреси з листів, що проходять через нього, і оголошень.

mailto:botik@cs.kiev.ua

gopher://rain.psg.com:70/1m/networks/connect/countries/ao — довідки про різні мережі в світі.

Системи "жовтих сторінок"

На вузлах "жовтих сторінок", звичайно, використовуються класифікації за родом діяльності, по продукції, що випускається, послугам, що надаються, за географічною ознакою. Іноді вони доповнені пошуком просто за алфавітом.

http://www.four11.com

Пошуковий сервер — щось середнє між "жовтими сторінками" та "білими сторінками". Робить пошук людей не тільки за власним іменем, але і за країною, містом, штатом, доменом, хобі. На сервері можна залишити інформацію про себе.

http://www.lookup.com
http://s19.bigyellow.com
http://www.whowhere.com
http://superpages.gte.com
http://sunsite.oit.unc.edu/~masha
http://www.yellownet.com
http://okra.ucr.edu/okra
http://www.yellow.com
http://www.ibdi.com
http://comfind.com
http://www.bigbook.com
http://www.bizweb.com
http://www.mcp.com/nrp/wwwyp

Запити до серверів телеконференцій і списків розсилки

У масових службах типу систем телеконференцій або списків розсилки звичайно передбачений вільний доступ до списку їх абонентів.

mailto:news@kiae.su USTAT
mailto:news@demos.su USTAT

Сервери новин, що роздають новини USENET, мають команду USTAT, що дозволяють одержати список абонентів, підписаних на цьому сервері на ту або іншу телеконференцію.

mailto:listserv@bitnic.bitnet

Списки розсилки мережі BITNET дозволяють дізнатися про склад їх передплатників.

Допомога з телеконференцій

comp.mail.maps

У цій телеконференції USENET регулярно публікуються списки UUCP-станцій, що потім зберігаються на багатьох файлових серверах.

comp.mail.misc

Регулярно публікується перелік питань, що задаються найчастіше (FAQ — Frequent Answered Questions) по пошуку адрес.

soc.college

Регулярно публікується FAQ по пошуку адрес у коледжах і університетах.

soc-net-people

Розміщення запитів адрес, регулярна публікація FAQ по пошуку адрес.

relcom.maps та ukr.maps

Телеконференції мережі Relcom, призначені для публікації запитів адрес, а також для розміщення різного роду матеріалів, що сприяють пошуку (наприклад списків вузлів мережі Relcom і списків абонентів деяких вузлів).

Додаткові документи

http://www.cis.ohio-state.edu/hypertext/faq/usenet/finding-addresses/faq.html

Документ, що допомагає в пошуку адрес e-mail людей.

Пошук e-mail і Web-адрес навчальних закладів і студентів

http://www.qusic.queensu.ca/FAQs/email/college.html

Документ, що допомагає в пошуку адрес людей, що закінчують або вже закінчили університет або коледж.

Деякі "білі сторінки" та пошукові системи:

http://www.nova.edu/Inter-links/cgi-bin/coll-email.pl

Пошук адрес e-mail коледжів:

http://www.procd.com — містить посилання на більш ніж 100 коледжів і університетів у світі.

Пошук Web-адрес в Internet

Зауваження по термінології

Для деякої однозначності термінології далі замість терміна ***URL*** використовується термін ***Web-адреса***. При необхідності будуть розрізнятися Web-адреса серверу і Web-адреса документу. В якості синоніма буде використовуватися термін ***посилання на Web-сервер*** або ***посилання на Web-документ***.

Сервер пошуку (*search engine*) — пошуковий Web-сервер, що обробляє запити на пошук Web-адрес документів. У WWW-технології кожний Web-документ при його створенні може бути позначений набором ключових слів на розсуд автора. Пошукові сервери зчитують ці ключові слова, знаходять такі ж у своїх великих словниках і добавляють посилання на даний Web-документ до списку вже наявних посилань для кожного такого слова.

Крім пошуку за ключовими словами, усі великі пошукові сервери мають універсальні ієрархічні класифікатори, що охоплюють усі галузі знань, сфери діяльності, області інтересів, області громадського життя і т. ін.

Деякі пошукові сервери містять усередині себе "жовті" та "білі сторінки" Web-адрес серверів.

"Жовті сторінки" WWW — пошуковий Web-сервер, що відпрацьовує запити на пошук Web-адрес серверів ("домашніх сторінок" організацій). Усі великі системи "жовтих сторінок" WWW дозволяють знаходити не тільки Web-адресу організації, але

разом із ним показують телефон, факс, телекс, звичайну та e-mail адресу. Іноді дається і короткий опис профілю організації.

Сайти-вказівники — Web-сервери, що містять велику кількість посилань на інші Web-сервери або на Web-документи. Можуть бути спеціалізованими або універсальними. Якщо предметна область складна або універсальна, то посилання звичайно не є ієрархічно класифіковані, у простому випадку вони просто упорядковані за алфавітом. Від серверів пошуку і "жовтих сторінок" WWW сайти-вказівники відрізняються обмеженим числом посилань (часто це кругле число: 100 або 1000), їхнім ретельним добором і відсутністю пошуку за ключовими словами.

Сайти-довідники — збірна назва для серверів пошуку, сайтів-вказівників, "жовтих сторінок" WWW і інших схожих джерел Web-адрес.

Завдання, засоби і способи пошуку Web-адрес

Пошук Web-адрес цікавий як самий по собі (пошук Web-документів на задану тему), так і як завдання пошуку організацій у мережі. У свою чергу, знайдений Web-сервер конкретної організації може бути цікавий як самий по собі, так і в якості джерела адресної інформації (телефони і адреси e-mail самої організації і її підрозділів).

Типові пошукові завдання:

- пошук Web-адреси серверу конкретної організації або конкретної людини;
- пошук Web-адреси серверу якоюсь організації за визначеними умовами;
- пошук нових Web-адрес серверів;
- пошук нових Web-адрес документів.

Основні засоби пошуку (сайти-довідники):

- великі універсальні пошукові Web-системи (сервери пошуку);
- пошукові Web-системи по декількох серверах пошуку;
- інші універсальні і спеціалізовані сервери пошуку та сайти-вказівники;
- довідкові системи "жовті сторінки" по Web-адресах серверів;
- каталоги серверів пошуку, сайтів-вказівників, "жовтих сторінок" WWW.

Додаткові способи пошуку:

- персональні і тематичні добірки посилань на Web-сервери;
- інформаційні бюлетені та довідники на Web-серверах, оголошення в телеконференціях, списки розсилки;
- запит до абонентів якої-небудь масової служби.

Слід зазначити, що число навіть основних пошукових засобів (різних сайтів-довідників) сягає декількох сотень, якщо не більше. Нижче перераховані або найбільш відомі з них, або ті, повідомлення про існування або появу котрих просто привернули увагу по тій або іншій причині.

Великі універсальні пошукові Web-системи (сервери пошуку)

Пошукові Web-сервери дозволяють знайти окремі Web-документи (Web-сторінки), що відносяться до заданих тематик або мають задані ключові слова або їх комбінації. На великих пошукових серверах є обидва ці способи пошуку (за ієрархією понять і за ключовими словами). Наповнення пошукових серверів відбувається або

автоматично (вони безупинно оглядають усі Web-сервери у світі), або вручну (з відбором матеріалу).

Пошуковий сервер звичайно має посилання і на інші великі пошукові сервери, але не передає їм автоматично запит на пошук. Найбільш відомі пошукові Web-сервери:

http://yahoo.com

Один із перших і найвідоміших серверів пошуку, що відрізняється дуже розвинутою ієрархічною класифікацією. Наповнюється вручну, має найменшу кількість посилань у порівнянні з іншими севрерами пошуку. Крім Web-адрес, містить посилання на ftp- і gopher-ресурси.

Має ієрархічний класифікатор і пошук по ключових словах, що підтримує операції "і", "або", але тільки одного типу в однім запиті. Знайдені по ключових словах посилання доповнюються ще й вказівкою про їх положення в ієрархічній класифікації серверу. Для скорочення діапазону пошуку передбачена можливість шукати в межах поточної підтеми класифікатора. На першому рівні ієрархії є розділ "References", що містить багато посилань на різного роду сайти-довідники. Має розвинутий сервіс новин.

http://www/lycos.com

Один із найвідоміших і найбільших по кількості посилань серверів пошуку. Має ієрархічний класифікатор і пошук за ключовими словами, що підтримує операції "і", "або", але тільки одного типу в однім запиті.

http://webcrawler.com

Відносно невеликий сервер пошуку, наповняється вручну — приблизно в 10 разів менший за Lycos. Має ієрархічний класифікатор і пошук за ключовими словами, що підтримує операції "і", "або", "не" і їхні комбінації.

http://www.inktomi.com

Новий і, мабуть, найбільший за кількістю посилань сервер пошуку — наповняється автоматично і зберігає всі знайдені посилання (приблизно в 5 разів більший за Lycos). Має ієрархічний класифікатор і пошук за ключовими словами, що підтримує операції "і", "або", але тільки одного типу в однім запиті. Крім власне пошуку документів має "жовті сторінки" по різних категоріях.

http://www.infoseek.com

Середній за кількістю посилань сервер пошуку (по деяких темах ледве більший за Webcrawler, але по деяких — більший за Lycos). Має ієрархічну класифікацію і пошук за ключовими словами, що не підтримує операцій "і", "або".

Пошук може здійснюватись:

- на всьому WWW-просторі;
- лише серед серверів, відібраних експертами даного серверу (у цьому випадку кожне знайдене посилання доповнюється короткою анотацією);
- лише серед серверів компаній (yellow-пошук, кожне знайдене посилання доповнюється коротким описом профілю компанії);
- серед статей USENET;
- серед адрес e-mail;
- серед свіжих новин.

Результат пошуку доповнюється списком відповідних тем з ієрархічної класифікації (загальний список тем для даного запиту, а не конкретні ієрархічні шляхи для кожного знайденого посилання, як у yahoo).

http://www.altavista.com

Великий сервер пошуку (трохи менший за Lycos по числу посилань). Має лише пошук по ключових словах, що підтримує тільки операцію "і". Пошук може проводитися на всьому WWW-просторі або серед статей USENET.

http://www.dejanews.com

Вважається найбільш потужним засобом пошуку новин USENET. Пошук може виконуватися по даті, автору, темі і групі.

http://www.hotbot.com

Середній за кількістю посилань сервер пошуку, із досить стійким зв'язком. Робить пошук за ключовими словами, що підтримує операції "і", "або", але тільки одного типу в однім запиті.

http://www.mckinley.com

Пошукова система Magellan — це система, типу Yahoo. Можливий пошук як за ключовими словами, так і за темами. Можна шукати за ключовими словами в рамках обраної теми. Має сервіс новин.

http://www.exite.com

http://www.opentext.com

http://www.nlightn.com

Пошукові WWW-системи по декількох серверах пошуку

Для проведення "тотального" пошуку відразу на декількох великих серверах є спеціальні засоби — сервери, що транслюють запит на пошук у форми, що підходять для декількох пошукових серверів:

http://www.w3com./fsearch/FrameSearch

http://ds2.internic.net/tools/web-subject.htmlhttp://www.albany.net/~wcross/all1srch.html

http://members.aol.com/markwelch/search.htm

http://www.nln.com

Інші універсальні і спеціалізовані сервери пошуку, великі сайти-вказівники

http://www.cs.colorado.edu/home/mcbryan/wwww.html — WWWWorm.

Пошук по усім URL, адресам, заголовкам документів, адресам документів. Можливість використання функцій "і", "або", але тільки одного типу в запиті.

http://www.miyawaki.pair.com/Bingo!

Має в списку декілька пошукових серверів, однак шукає, тільки на першому із визначеного списку.

http://info.cern.ch/hyDertext/DataSources/bvSubiect/Qverview.html

http://sailfish.perearine.com/WebWorld/welcome.html

http://Qagme.wwa.com/~boba/spiderl.html

http://www.diaitmark.net/wow

http://pubweb.nexor.co.ukpublic/cusi/cusi.html

http://www.quantumsoft.com/formHomologvsearch
http://netcenter.com/netcentr/roadmap/index.html
http://schiller.wustl.edu/DACLOD/daclodDA-CLOD
http://qalaxv.einet.net/galaxy.htmlGalaxvWWW
http://www.stir.ac.uk/jsbin/jsii
http://www.ozemail.com.au/%7Ehuaom/other.html

Добірки посилань на Web-сервери (маленькі сайти-вказівники)

Дуже багато домашніх сторінок містять добірки WWW-посилань за якоюсь спеціальною тематикою, або по декількох темах. Ці добірки не претендують на повноту, тому що є результатом ручного добору, але саме з цієї причини вони можуть бути особливо корисні для початкового ознайомлення з якоюсь темою.

http://www.tricky.com/liz

Сторінка, "розбита" на декілька тематик, у кожній з яких є посилання. Містить теми: бізнес, політика, комп'ютери й Internet, розваги, пошук зниклих дітей і ін.

http://www.tstimpreso.com/hotsheet/

Має розбивку на безліч тем і підтем. Велике число посилань по кожній темі. Посилання як на пошукові системи і сервери новин, так і на кожну з підтем (спорт, ПЗ, ігри, гумор, мистецтво і література, наука, їжа і т.ін.).

http://www.myanmar.com/myanmar/motherof all link.html
http://www.hookup.net/~jmorris/cool/
http://www.tias.com/dealers.html
http://www.island.net/~rednikki
http://www.vnet.net/users/voyager/CybernetVoyager.html
http://homepaaes.enterprise.net/kevinc/addrand.html
http://www.skylink.net/%7Eanytime/cool.html
http://www.dublclick.com/coolLinks.html
http://cent.com/abettina/Links.html
http://btdqs.usqs.gov/index/cooHink.html
http://www.n-vision.com/Ouazimoto/bookmark.htmBookmarks
http://cactus.org/%7Ewoan/hotlist.htmlWoan's
http://totalweb.totalpc.com/cool.html
http://www.craftstore.com/links.htmSeriesACollection
http://www.dancon.com/cool.htmlCoolLinks
http://www.qil.com.au/%7Etedb/fav.html
http://www2.ecst.csuchico.edu/%7Ebeej/hotlist.html
http://felix.scvnet.com/%7Etimm/bkmk1.html

Системи "жовтих сторінок" по Web-адресах серверів

На відмінність від пошукових Web-серверів, системи "жовтих сторінок" із Web-адресами містять посилання не на окремі Web-документи, а на Web-сервери різних організацій (домашні сторінки). Самі ж організації розташовані за родом діяльності, продукцією, що виготовляється, послугами, що надаються, за географічною ознакою, просто за алфавітом ("білі сторінки").

http://www.gnn.com

Одна з перших і найбільш відомих систем "жовтих сторінок" WWW. Містить посилання на тисячі Web-серверів. Має універсальну ієрархічну класифікацію, а також алфавітний перелік тем і серверів ("білі сторінки"). Кожне посилання постачається коментарем приблизно в один абзац.

http://www.vellow.com

Відома система "жовтих сторінок". Кожне посилання доповнене коментарем із назвою організації, поштовою адресою, телефоном, факсом, телексом.

http://www.directory.net

Пошук за парою ключових слів, що підтримує операції "і", "або".

http://www.fourll.com

Пошуковий сервер — щось середнє між "жовтими" і "білими сторінками". Досить гнучка система пошуку адреси e-mail людини за його іменем, прізвищем, доменом, країною, штатом, містом, хобі. На сервері можна залишити інформацію про себе.

http://www.whowhere.com

Швидкий і зручний засіб пошуку людей і організацій. Є можливість обробки невірно введених або неповних імен і ініціалів.

http://okr3.ucr.edu/okra/

Пошук адреси e-mail людини за ключовими словами (ім'я, прізвище, посада і т. ін.). Всі ці ключові слова йдуть через зв'язування "або".

http://www.lookup.com

http://ben.net/yellow/index.html

http://sunsite.oit.unc.edu/~masha/

Каталоги серверів пошуку, сайтів-вказівників, "жовтих сторінок" WWW.

http://www.ont3p.com/sub serv.html

http://www.ncsa.uiuc.edu/SDG/Software/Mosaic/MetaIndex.html

http://cuiwww.uniae.ch/meta-index.html

http://www.rit.edu/~djs4601/searcher.htm

http://www.paael.org/search/

http://www.virtualynx.com/whereisit

http://ourworld.compuserve.com/homepages/frankvad/tools.htm

http://www.infohiway.com/way/search/index.html

http://www.pegasus.oz.au/%7Ehomebuilder/search/search-enaines/

Інформаційні бюллетені, керівництва, списки розсилки, телеконференції

Електронні інформаційні бюлетні (*newsletters*), що розташовані на Web-сервері, інформують про різні новини на визначену тему. Багато інформаційних бюллетенів доступні і через список розсилки.

Учбово-довідникові матеріали звичайно містять в якості прикладів корисні добірки посилань.

Інформаційні бюллетені та керівництва, присвячені новинам у Web-світі:

http://www.elvis.ru/koi8/internet/journals

Путівник The Internet Press (російською мовою).

http://ourworld.compuserve.com/homepages/frankvad/Internet.htm

http://www.ibic.com/Diaest

http://www.hw.ac.uk/libWWW/irn/irn.html

Деякі з інформаційних бюллетенів доступні і у вигляді списків розсилки для передплатників електронної пошти. Підписатись на будь-який список можна як традиційним способом, пославши лист із запитом на підписку за адресою списку, так і заповнивши передплатну форму на відповідному Web-сервері.

```
listserv@lists.internic.net
NET-HAPPENINGS-REQUEST@lists.internic.net
infomaq@larant.mipt.msk.su
http://rs.internic.net/scout/report
ftp://rs.internic.net/scout/
majordomo@colossus.net
http://csbh.mhv.net/~bobrankin/tourbus
weeklv@webcom.com
http://www.webcom.com/weekly/wb/weekly.html
http://www.webcom.com/weekly/weekly-mail.html
http://www.webcom.com/weekly/wb/subscribe.html
listserv@library.berkeley.edu
http://sunsite.berkeley.edu/Web4Lib/archive.html
```

Пошук Web-адрес навчальних закладів

Деякі "білі сторінки" та пошукові сервери Web-адрес університетів і колледжів:

```
http://www.mit.edu:8001/people/cdemello/univ.html
http://isl-qarnet.uah.edu/Universities/universities.html
http://www.clas.ufl.edu/CLAS/american-universities.html
http://www.usmall.com/college/search/index.html
http://www.mcli.dist.maricopa.edu/cc/
http://www.colleaenet.com/
http://watservl.uwaterloo.ca/~credmond/univ.html
http://src.doc.ic.ac.uk/uk-academic.html
http://www.procd.com
```

Містить більше, ніж 100 посилань на домашні сторінки і телефонні довідники коледжів і університетів по всьому світі.

3.8.2. Пошукові ресурси України

Пошукові ресурси України детально подані в довіднику компанії Lucky.net за 2003/2004 роки "Ресурси українського Internet". Вони класифікуються за наступними темами:

- держава і політика;
- бізнес та економіка;
- засоби масової інформації;
- комп'ютери та програмне забезпечення;
- Internet і зв'язок;
- наука і освіта;
- культура, мистецтво і релігія;
- медицина і екологія;
- досуг і розваги.

Враховуючи те, що кількість пошукових ресурсів на сьогодні складає сотні тисяч, то немає сенсу перелічувати їх. Читач сам вирішить, які ресурси йому будуть корисні, а які ні, коли звернеться до вказаного вище довідника, що публікується щорічно.

3.9. Web-сервер Apache

Враховуючи популярність, важливість та розповсюдження сервера Apache, розглянемо деякі його характеристики.

3.9.1. Основні відомості про Web-сервер Apache

Перш за все, Apache є HTTP-сервером для системи UNIX. Як відомо, все програмне забезпечення під UNIX має спільні особливості, що дуже відрізняють його від інших програмних продуктів. По-перше, програмні продукти для UNIX пропонуються й розповсюджуються виробниками у вигляді вихідних текстів програми (звичайно написаних на мові C), які мають бути скомпільовані безпосередньо на машині, куди встановлюється відповідне програмне забезпечення. По-друге, більшість з них може бути встановлена тільки людиною, що є адміністратором даної машини або має на ній достатні права та привілеї.

Чому саме UNIX? Ця система вирізняється підвищеною надійністю і якнайкраще підходить для роботи з Internet. Вона була розроблена програмістами і для програмістів, і хоча UNIX досить складна в адмініструванні, вона користується надзвичайною популярністю і авторитетом серед спеціалістів. Правами на Apache володіє група розробників програмного забезпечення Національного Центру Суперкомп'ютерних Прикладних Програм Іллінойського Університету (the Apache Group).

Цей Web-сервер використовується багатьма Web-адміністраторами UNIX через його зручність та гнучкість, і є чи не найбільш популярним HTTP-сервером у світі. У наступних розділах мова піде безпосередньо про роботу з Apache: від компіляції до стадій використання.

3.9.2. Встановлення Apache на власному комп'ютері

Якщо вирішено створити власний Web-сервер, і вибрана саме платформа UNIX та Apache, перш за все потрібно отримати вихідні тексти цього продукту за адресою `http://www.apache.org`.

На Web-сайті `http://www.lexa.ru` можна дістати російські версії Apache. Отже, треба отримати архів, розпакувати його та скомпілювати Apache. Компіляція звичайно складається з двох етапів: передкомпіляційне конфігурування зі створенням make-файлу і безпосередньо компіляція. У комплекті вихідних файлів та документації по Apache звичайно є й конфігураційний сценарій (програма, що інтерпретується безпосередньо системою, а тому компіляції не потребує). Це файл під назвою `configure`. Тому для того, аби задати Apache певну конфігурацію, треба, знаходячись в директорії з вихідними файлами, ввести команду

```
=> ./configure [-options]
```

де [-options] — опції, які можуть бути вказані для конфігурування. Весь набір цих опцій можна продивитись за допомогою команди

```
=> ./configure –help
```

Для того, аби додати модуль, попередньо потрібно в каталог Apache_home_folder/src/modules/ ще не скомпільованого Apache проінсталювати всі потрібні модулі, отримавши в результаті файли Apache_home_fblder/src/modules/*.a. Потім при запуску конфігураційного сценарію потрібно вказати наступні опції:

```
=>./configure -activate-module=src/modules/your_module.a
```

Наприклад, це може виглядати так:

```
=> ./configure –activate-module=src/modules/libphp3.a -
activate-module=src/libperl.a
```

Після того, як Apache попередньо сконфігуровано, потрібно зпустити власне компіляцію. Це виконується командою

```
=> make
```

Після досить довгої процедури компілювання отримуємо вже майже готовий до запуску продукт у тому самому каталозі, де знаходилися вихідні тексти. Але його може бути перенесено в інший каталог, для чого потрібно переписати туди каталоги conf/, logs/, icons/ та основний його бінарний файл, який запускає власне HTTP-сервер — httpd. Але перед тим, як його запускати, Apache має бути остаточно сконфігуровано.

3.9.3. Конфігурування та запуск Apache

Наступним кроком має бути редагування конфігураційних файлів, в яких знаходяться різноманітні директиви. Від цих директив залежить, як саме буде працювати сервер.

Усього основних конфігураційних файлів три: srm.conf, access.conf та httpd.conf. Вони знаходяться в каталозі Apache_home_folder/conf/ і потребують якнайретельнішого налаштування. Також в цьому каталозі повинен бути файл mime.types, який редагувати звичайно не треба. Для тих, хто вперше зустрічається з Apache, там же існують файли-приклади: srm.conf-dist, access.conf-dist і httpd.conf-dist. Для того, аби ними скористатися, потрібно спочатку виконати команду

```
=> mv *.conf-dist *.conf
```

Потім потрібно буде послідовно відредагувати ці файли. У кожному з них є детальні пояснення до кожної директиви.

Файл httpd.conf

Перш за все потрібно відредагувати httpd.conf — основний файл, в якому знаходиться вся найважливіша інформація про сервер: порт, на якому він працює, користувач, що є його власником та має право його запускати і багато інших. Деякі його директиви розглянемо детальніше.

Директива **ServerRoot** — визначає кореневий каталог HTTP-сервера, відносно якого будуть налаштовані всі шляхи, задані не абсолютним, а відносним шляхом,

підключатися конфігураційні файли тощо. Можна сказати, що це — каталог, в якому "живе" сервер. В ньому звичайно мають знаходитись каталоги `conf/` та `logs/`.

Наприклад:

```
ServerRoot /usr/local/apache
```

Директива **`Port`** — задає порт, на якому сервер буде працювати і має наступний синтаксис:

```
Port <port number>
```

Наприклад, щоб вказати стандартний HTTP-порт 80, можна написати в `httpd.conf` наступний рядок: `Port 80`

При запуску потрібно врахувати, щоб вказаний порт не використовувався іншим процесом, або ж, якщо Apache запускає не адміністратор, а інший користувач, щоб номер порта був не нижчий за 1024. (Допустимі номери — від 0 до 65535).

Директива **`ServerType`** — може приймати одне з двох значень: `inetd` або `standalone`. Якщо в конфігурації вказати

```
ServerType inetd,
```

то процес `httpd` буде завантажуватися кожний раз, коли стартує стандартний процес `inetd`, і буде автоматично вписаний в `inetd.conf`. Якщо ж вказана опція `standalone`, то `httpd` потребуватиме запуску вручну.

Директива **`PidFile`** (PID — Process ID) — використовується для того, аби задати файл, в якому буде зберігатися номер процесу `httpd`. Цю директиву потрібно використовувати тільки для `standalone`-типу сервера.

Приклад:

```
PidFile/usr/local/httpd/log/httpd.pid
```

або ж

```
PidFile logs/pidFile.httpd
```

Директива **`DocumentRoot`** — вказує місцезнаходження html-файлів. За приклад можна взяти уривок з конфігураційного файлу:

```
DocumentRoot /usr/xvar/pages
```

Директива **`AccessConfig`** — в `httpd.conf` можна задати свої власні файли конфігурації замість `access.conf` та `srm.conf` за допомогою директив `AccessConfig` та `ResourceConfig`.

Приклад:

```
AccessConfig conf/MyAccess.conf
```

Директива **`ErrorDocument`** — відповідає за обробку помилок при зверненні до сервера. Ця директива має наступний синтаксис:

```
ErrorDocument <код помилки> <повідомлення>
```

За повідомлення може стати як текст, поданий у лапках, так і html- або cgi-файл.

Наприклад:

```
ErrorDocument 403 "Sorry, you have no access" ErrorDocument
404 /cgi-bin/bad_url.pl ErrorDocument 401 ttp://host.com/subs.html
```

Файл access.conf

Тепер потрібно відредагувати access.conf, аби встановити хоча б базову систему доступу до нашого сервера. Доступ також може регулюватись файлами з назвою .htaccess безпосередньо в каталогах. Так само, як і httpd.conf, access.conf має безліч директив, за допомогою яких регулюється доступ до сервера та його частин. Розглінемо декілька з них.

Директива **require** — може обмежувати доступ до каталогів, розподіляючи його за групами або користувачами. Синтаксис у неї відповідний:

```
require group <groupname> <groupname> <...>
require user <userID> <userID> <...>
```

Це означає, що тільки перелічені групи та користувачі матимуть доступ до відповідного каталогу, якщо цей каталог вказується в файлі .htaccess, або до всього сервера, якщо його вказано в access.conf.

Директива **AccessFilkeName** — з її допомогою можна вказати власну назву для файлу коригування доступу до певного каталогу. Наприклад,

```
AccessFileName .acc
```

означатиме, що властивості каталогу задає файл з назвою .ace, який знаходиться в цьому каталогу. Тоді для /usr/local/httpd/docs/somedir/ сервер спочатку перевірятиме директиви в /.acc, потім — в usr/.acc, /usr/local/httpd/.acc, /usr/local/httpd/docs/.acc і, нарешті, — в /usr/local/httpd/docs/somedir/.acc, якщо вони не були відключені директивою

```
<Directory/>
AllowOverride None
</Directory>
```

Запуск Web-сервера

Для того, аби активізувати сервер, потрібно запустити бінарний файл httpd з каталогу, куди було проінстальовано Apache. Для нього можуть бути вказані наступні рядкові опції:

- **-f <config-file>** — якщо конфігурація знаходиться не в conf/httpd.conf, а в якомусь іншому файлі;
- **-d <server-root>** — для задання робочого каталогу серверу не з файлу конфігурації, а вручну;
- **-X** — ця опція призначена для запуску процесу в однопроцесному режимі для відлагодження. Тоді він не перейде в фоновий режим і не буде створювати жодного дочірнього процесу. Не рекомендується використовувати цей режим для звичайного Web-сервісу;
- **-v** — вивести версію продукту і перервати роботу;
- **-h** — вивести перелік усіх директив разом з довідкою про них;
- **-l** — вивести перелік всіх модулів, що були зкомпільовані разом з Apache;
- **-?** — вивести всі можливі опції і припинити роботу.

Коли основний процес запущено в звичайному режимі, він створює кілька додаткових процесів. Щоб припинити роботу сервера, не потрібно переривати додаткові процеси, бо вони знову будуть створені основним. Аби зупинити роботу сервера, не-

обхідно надіслати повідомлення TERM основному процесу. Необхідна для цього команда:

```
=> kill -TERM 'cat /usr/local/httpd/logs/httpd.pid'
```

3.9.4. Робота з Apache

Коли сервер вже сконфігуровано, для його функціонування потрібно лише запускати його щоразу, як машина стартує, що може бути легко реалізовано за допомогою inetd. Крім того, потрібно створити ряд основних html-документів, які власне будуть обличчям вашого сервера. Для того, аби змінювати конфігурацію Apache, його потрібно зупиняти і запускати знову після редагування конфігураційних файлів. А от для зміни структури його html-частини це робити не обов'язково.

Головна сторінка серверу знаходиться прямо в каталозі, заданого директивою DocumentRoot і має назву index.html. Взагалі, якщо у будь-якому www-каталозі сервера знаходиться файл з такою назвою, посилання на цей каталог автоматично перенаправляється на цей файл. Наприклад, якщо всі документи лежать у каталозі /r/www/, то URL http://snark.ukma.kiev.ua буде відповідати /r/www/ index.html, а http://snark.ukma.kiev.ua/links/ буде відповідати r/www/links/index.html.

Розділ 4

Протоколи Internet

"Усе розумне має свої межі,
безмежна тільки дурість"
(Народна мудрість)

Коротко протоколи мереж розглядаються в першому розділі книги, а стандартні протоколи Internet, такі як IP, TCP та UDP, — в попередньому розділі. В поточному розділі протоколи Internet розглянуті більш докладно, з описом технічних деталей та структури пакетів даних.

4.1. Організаційні структури Internet

Реалізація проекту стосовно основ побудови Internet з самого початку потребувала прийняття організаційних мір, що координують зусилля розробників. У зв'язку з цим створювались і реорганізовувались різні технічні комітети, які у 1989 році набули сучасної структури у вигляді центрального органу IAB (Internet Activities Board — координаційна рада Internet), що включає два підкомітети: дослідницький — IRTF (Internet Research Task Force) і "законодавчий" — IETF (Internet Engineering Task Force). IETF — це основна структура Internet, що відає питаннями стандартизації, яка приймає стандарти RFC (Request For Comments — запити на коментарі) і є міжнародною організацією, що включає великі секції (по напрямках), усередині яких у свою чергу формуються робочі групи (по задачах). У підкомітету IETF існує визначена практика прийняття проекту RFC, що базується на необхідності розгляду декількох незалежних реалізацій запропонованого стандарту.

Всі прийняті IETF стандарти RFC (а також інші важливі матеріали, що заслуговують на увагу) загальнодоступні в середині Internet через електронну пошту, файлові сервери та ін.

У Internet також існує орган, відповідальний за поширення технічної інформації, роботу з реєстрації і підключення користувачів до мережі, а також за рішення ряду адміністративних задач, таких як розподіл адрес. Цей орган називається "Центр мережевої інформації".

4.2. П'ятирівнева архітектура управління в Internet

Під родиною протоколів TCP/IP у широкому сенсі розуміють звичайно весь набір стандартів RFC. Проте загальним і основним елементом для всіх цих протоколів є Internet Protocol (IP). Цей протокол, власне, і реалізує поширення інформації з IP-мережі. Його значення як технологічної основи Internet дуже велике.

Протокол IP здійснює передачу інформації від вузла до вузла мережі у вигляді дискретних блоків-пакетів. При цьому IP не несе відповідальності за надійність доставки інформації, цілісність або зберігання порядку потоку пакетів. Цю задачу вирішують два інших протоколи: TCP (Transmission Control Protocol — протокол керуван-

ня передачею даних) або UDP (User Datagram Protocol — дейтаграмний протокол передачі даних), що "лежать" над IP (тобто використовують процедури протоколу IP для передачі інформації, додаючи до них свою функціональність).

Протоколи TCP та UDP реалізують різноманітні режими доставки даних. TCP — протокол передачі даних разом із з'єднанням, а UDP — дейтаграмний протокол.

Вище, над транспортними протоколами TCP або UDP, знаходяться протоколи, що реалізують ті чи інші прикладні служби, такі як обмін файлами (File Transfer Protocol — FTP) і повідомленнями електронної пошти (Simple Mail Transfer Protocol — SMTP), що забезпечують термінальний доступ до віддалених серверів (TELNET).

Таким чином, ієрархію керування в TCP/IP-мережах можна подати у вигляді п'ятирівневої концептуальної моделі (RFC-791 і RFC-1349), наведеної на рис. 4.1:

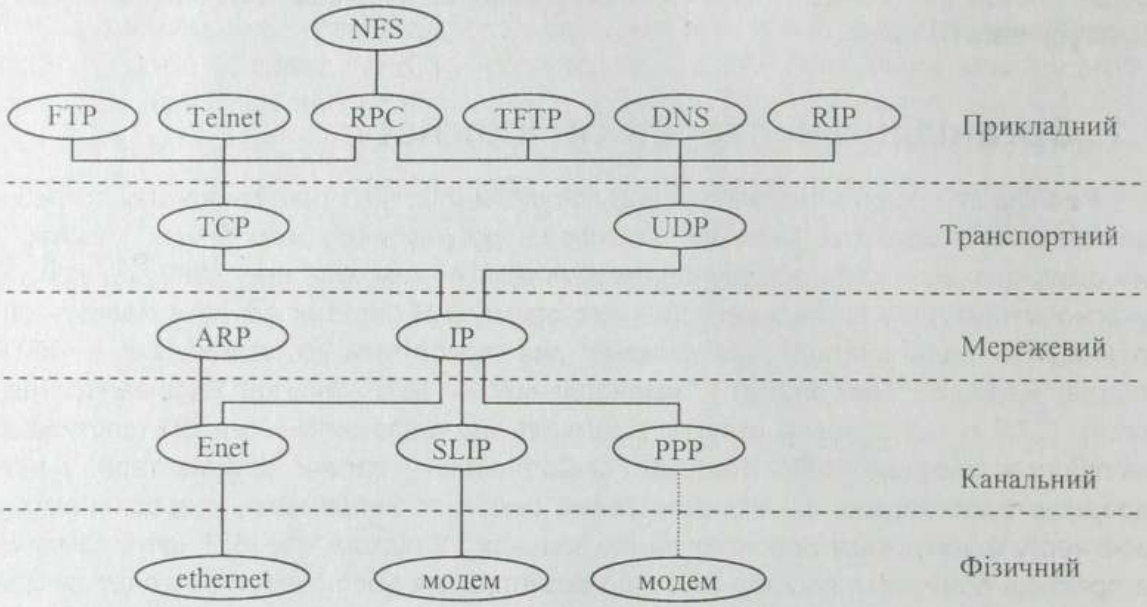


Рис. 4.1. Архітектура і сукупність протоколів TCP/IP вузла зв'язку Internet

Перший рівень — фізичний (апаратне забезпечення) описує те або інше середовище передачі даних.

На другому рівні — канальному (мережевий інтерфейс) — "лежить" апаратно залежне ПЗ, що реалізує поширення інформації на тому або іншому відрізку середовища передачі даних.

Третій рівень — мережевий — це і є протокол IP. Його головна задача — маршрутизація (вибір шляху через множину проміжних вузлів) при доставці інформації від вузла-відправника до вузла-адресата. Друга важлива задача протоколу IP — приховання апаратно-програмних особливостей середовища передачі даних і надання вищім рівням єдиного уніфікованого і апаратно незалежного інтерфейсу для доставки інформації. При цьому досягається канальна (апаратна) незалежність і забезпечується багатоплатформне застосування прикладних програм, що працюють над IP.

При цьому протокол IP не забезпечує транспортну службу в тому сенсі, що не гарантує доставку пакетів, зберігання порядку і єдності, і не відрізняє логічні об'єкти (процеси), що породжують потоки інформації. Це задачі інших протоколів — TCP та UDP, що відносяться до четвертого, транспортного рівня.

Вище — на п'ятому, прикладному рівні — лежать прикладні задачі, що запитують послуги у транспортного рівня.

4.3. Протоколи канального рівня SLIP та PPP

4.3.1. Протокол канального рівня SLIP (Serial Line IP)

Першим стандартом канального рівня, що забезпечував роботу терміналів користувачів по лініям зв'язку і реалізовував послідовну передачу символів, став протокол SLIP (Serial Line IP), розроблений на початку 80-х років. Пізніше цей протокол був підтриманий в ОС UNIX та реалізований у програмному забезпеченні для ПК.

Протокол SLIP характеризується тим, що він забезпечує можливість підключення до Internet через стандартний інтерфейс RS-232, що є у більшості комп'ютерів. На теперішній час SLIP широко використовується у комп'ютерах, які підключені до ліній зв'язку з пропускною спроможністю 1,2–28,8 Кбіт/с.

Псевдоструктура кадру SLIP

За змістом, кадр SLIP структури не має — він тільки передбачає розмежування пакетів IP (пакетів мережевого рівня), що передаються послідовно. Саме цим забезпечується синхронне введення пакетів у канал зв'язку (фізичний рівень).

Протокол SLIP не визначає максимально припустиму довжину "інформаційного поля" кадру, що передається, проте реальний розмір "вкладеного в кадр" пакета IP не повинен перевищувати 1006 байтів. Дане обмеження пов'язане з першою реалізацією протоколу SLIP у відповідних ОС BERKLEY UNIX, і тому дотримання його необхідно для забезпечення сумісності різних реалізацій (версій) SLIP.

Недоліки SLIP:

- не забезпечує обмін адресною інформацією — це обмеження не дозволяє використовувати SLIP для деяких видів мережевих послуг;
- відсутня індикація типу протоколу, пакет якого "вкладається" у кадр SLIP, тому через послідовну лінію по протоколу SLIP можна передавати трафік лише одного мережевого протоколу;
- не передбачені процедури виявлення і коригування помилок — ці функції забезпечують протоколи рівнів, що лежать вище.

Протокол ущільнення службової інформації CSLIP

Для підвищення ефективності використання пропускної спроможності послідовних ліній зв'язку використовуються алгоритми ущільнення даних (наприклад, за рахунок зменшення обсягу службової інформації, що знаходиться в заголовках пакетів IP).

Таку задачу вирішує протокол CSLIP (Compressed SLIP — стиснутий SLIP). При використанні протоколів типу TELNET для доставки одного байту даних потрібно переслати 20-байтовий заголовок пакета IP і 20-байтовий заголовок пакета TCP (разом 40 байтів). Протокол CSLIP забезпечує ущільнення 40-байтового заголовку до 3–5 байтів, тому більшість реалізацій протоколу SLIP підтримують специфікацію CSLIP.

4.3.2. Протокол канального рівня PPP (Point-to-Point Protocol)

Протокол PPP (RFC-1661) був розроблений Інженерною проблемною групою Internet і прийшов на зміну застарілому протоколу SLIP.

Загальна характеристика протоколу PPP

На відміну від SLIP, протокол PPP може працювати не тільки з інтерфейсом RS-232, але і з іншими інтерфейсами між ЗОД (завершене обладнання даних) і АКД (апаратура каналу даних) (RS-422, RS-423 і V.35). Протокол PPP може працювати без керуючих сигналів модемів (таких, як "Request to Send", "Clear to Send", "Data Terminal Ready"). Єдина жорстка вимога, запропонована PPP до лінії зв'язку, — забезпечення дуплексного з'єднання.

Протокол PPP включає в себе:

- механізм обмеження пакетів протоколів мережевого рівня і формування кадрів для передачі по лінії зв'язку;
- протокол Link Control Protocol (LCP, RFC-1471) для встановлення, конфігурування і тестування з'єднання;
- протоколи мережевого керування Network Control Protocol (NCP, RFC-1473 і RFC-1474) для встановлення і конфігурування процедур передачі повідомлень, які надійшли з мереж, що функціонують по різноманітним мережевим протоколам.

Логічна характеристика протоколу PPP

Формат кадру протоколу PPP (рис. 4.2) аналогічний формату кадру протоколу HDLC (High-level Data Link Control — управління каналом високого рівня) і включає:

- прапорець "Flag" (01111110);
- поле "Адреса" (11111111);
- поле "Керування" (00000011);
- поле "Протокол" (2 байти), значення якого визначається типом пакету, що знаходиться в полі "Інформація";
- поле "Інформація" (до 1500 байтів);
- поле "Контрольна сума" (2 байти) — зберігає код CRC (Cyclic Redundancy Code — код контролю за допомогою циклічної надлишковості);
- прапорець.

Прапорець	Адреса 11111111	Керування 00000011	Протокол (2 байти)	Інформація (до 1500 байтів)	Контрольна сума (2 байти)	Прапорець

Рис. 4.2. Формат кадру PPP

Якщо біти кадру PPP передаються послідовно, тоді, у випадку появи між прапорцями послідовностей із п'ятьох бітів "1", після кожної такої послідовності додається біт "0", для того, щоб уникнути комбінації "прапорець". На приймаючій стороні наприкінці "нульові" біти відкидаються.

Якщо кадр PPP передається через фізичний інтерфейс паралельно (блоками з числом бітів, кратним 8, із метою забезпечення циклової синхронізації), тоді, у випадку появи між "прапорцями" байтів із значеннями "TE", "7D" (значення символу "ESC") і значеннями "XX", меншими 20 (значення керуючих символів коду ASCII), відбуваються наступні заміни:

- байт "TE" заміняється на "7D", "5E";
- байт "7B" заміняється на "70", "5D";
- байти "XX" із значеннями, меншими 20, заміняються на "XX", "01".

Процедурна характеристика протоколу PPP

Розглянемо спрощений алгоритм функціонування протоколу PPP. Початкова фаза починає і закінчує процес зв'язку. У випадку появи зовнішньої події (наприклад, готовності апаратного забезпечення здійснити зв'язок) буде ініційована фаза встановлення з'єднання, протягом якого відбувається узгодження різноманітних параметрів з'єднання (обмін кадрами LCP). У випадку неможливості встановлення з'єднання процес припиняється, і протокол перейде у стан початкової фази.

Якщо всі необхідні параметри узгоджені, то буде ініційована фаза аутентифікації, протягом якої проводиться перевірка достовірності учасників сеансу зв'язку (якщо це потрібно). У випадку невдалої аутентифікації процес з'єднання перейде у фазу роз'єднання, яка готує розрив з'єднання.

Якщо фаза аутентифікації пройшла успішно, то протокол переходить до фази передачі даних. У цій же фазі здійснюється обмін даними. У фазі роз'єднання (використовується по закінченні передачі кадрів або у випадку виникнення якихось помилок) припиняється передача кадрів, і протокол PPP переходить у стан початкової фази.

Блок-схема алгоритму функціонування протоколу PPP наведена на рис. 4.3.

Рис. 4.3. Блок-схема алгоритму функціонування протоколу PPP

Переваги протоколу PPP

У порівнянні з протоколом SLIP, протокол PPP є значно більш розвинутим інструментом і має наступні переваги:

- можливість одночасної роботи з різноманітними мережевими протоколами, а не тільки з IP;
- перевірка цілісності даних;
- підтримка динамічного обміну адресами IP;

- можливість "стискання" заголовків пакетів IP і TCP за допомогою алгоритмів, механізм яких схожий на реалізований у протоколі CSLIP.

4.4. Протокол IP

Призначення протоколу IP (RFC-760, RFC-791) аналогічне призначенню протоколу мережевого рівня X.25. Відповідно до концепції IP, множина обчислювальних машин, яка залучена до деякої єдиної глобальної мережі передачі даних (МПД), внутрішня структура якої для користувачів неважлива, може включати декілька фізичних мереж.

Фізичні мережі, по суті, не реалізують протокол IP. Всі функції протоколу IP виконують головні обчислювальні машини (ГОМ), керуючі обчислювальні машини (КОМ) і маршрутизатори. Тому локальні мережі або глобальні телекомунікаційні магістралі (або їхні довільні композиції) рівнозначні і розглядаються як єдине транспортне середовище.

4.4.1. Характеристика протоколу IP

Відомо, що протокол IP:

- Не забезпечує гарантовану доставку інформації (100% ймовірність доставки), оскільки пакет може бути загублений, повторно переданий через один і той же вузол зв'язку, затриманий, доставлений із порушенням порядку. При цьому протокол IP не тільки нікого не повідомляє про ці явища, але і не має інструментів для їх виявлення.
- Забезпечує дейтаграмну доставку (або доставку без встановлення з'єднання), оскільки кожний пакет являє собою незалежний від інших блок даних, що підлягає обробці. Пакети, що послідовно виходять від споживача, можуть розгалужуватись по різним шляхам в мережі, губитись та змінювати порядок.

Треба зазначити, що протокол IP базується на декількох стандартах RFC і визначає:

- формат пакету IP (логічна характеристика протоколу);
- механізми розповсюдження (маршрутизації) пакетів (процедурна характеристика протоколу);
- способи розв'язання конфліктних ситуацій (процедурна характеристика протоколу).

4.4.2. Структура пакета IP

Пакет IP складається із заголовку та блоку даних. Він "працює" лише із заголовками.

Поле "Версія"

4-бітове поле "Версія протоколу IP" використовується для усунення конфліктів, які можуть з'являтись при зміні версії протоколу IP. Якщо у полі "Версія" знаходиться значення, відмінне від поточної версії протоколу, то пакет знищується.

Поле "Довжина заголовку IP"

Поле "Довжина заголовку IP" задає значення довжини заголовку пакета, що вимірюється у 32-бітових словах. Це поле передбачає зміну довжини заголовку у відповідності з полями "Послуги" (змінною довжини), а також "Доповнення поля "Послуги" до 32-бітової межі".

Поле "Категорія обслуговування пакету"

Поле "Категорія обслуговування пакету" включає:

- сегмент "Пріоритет" (3 біти) — може приймати вісім значень від 0 (звичайний пріоритет) до 7 (мережеве керування);
- біти "D", "T", "R" — вказують на тип транспортування, про що "запитує" пакет встановлення цих бітів у стан "1" потребує відповідно низької затримки при передачі пакета, високої пропускної спроможності і високої надійності. Останні два біти не використовуються.

Потрібно відзначити, що поле "Категорія обслуговування пакету" не завжди використовується маршрутизаторами, хоча може застосовуватись для оптимізації транспортної служби.

Поле "Довжина пакету в октетах разом із заголовком"

Поле "Довжина пакету в октетах разом із заголовком" задає повну (включаючи заголовок і дані) довжину пакета, що вимірюється в октетах (байтах). Повна довжина пакета IP, в принципі, може досягати 65 535 байтів.

Протокол IP, як відомо, вирішує проблему обмеження на максимальну довжину кадру, дозволену в тій чи іншій фізичній мережі (MTU — Maximum Transfer Unit), тобто вирішує задачу розподілу (фрагментація) великих пакетів на малі (і навпаки, їх складання). Це потрібно робити в тих випадках, коли на вхід деякої фізичної мережі надходить пакет, довжина якого перевищує значення MTU для даної мережі.

Фрагментація здійснюється наступним чином. Блок даних вихідного (великого) пакета розподіляється так, щоб розмір отриманих фрагментів у сумі з довжиною заголовка не перевищував розміру кадру MTU для фізичної мережі, до якої направляються фрагменти. При цьому фрагменти упаковуються в пакети, заголовки яких дуже схожі на заголовок вихідного пакета.

Щоб зрозуміти, що дані пакети містять фрагменти одного великого пакету, і забезпечити його наступне складання, здійснюється установка спеціальних ознак в поле-індикатор "Ще дані"; байти, по яких "розрізався" вихідний блок даних, розміщуються в полі "Номер байта", на якому зроблена чергова фрагментація вихідного "великого" пакета"; а в поле "Ідентифікатор переданого вихідного "великого" пакета" записується один, загальний для всіх фрагментів, ідентифікатор, що вказує на приналежність фрагментів до одного "великого" блоку даних.

Поле "Час існування пакета в мережі"

Поле "Час існування пакета в мережі" вказує термін, протягом якого пакет повинний існувати в мережі. ГОМ і маршрутизатори, що обробляють даний пакет, змен-

шують значення цього поля за спецалгоритмами. Коли час існування минає, пакет знищується. При цьому джерело повідомлення сповіщається про втрату пакета. Наявність кінцевого терміну існування пакета забезпечує, зокрема, захист від таких небажаних подій, як передача пакета по циклічному маршруту та перевантаження мереж.

Поле "Тип транспортного протоколу TCP/UDP"

Поле "Тип транспортного протоколу TCP/UDP" (8 бітів) вказує протокол рівня, що знаходиться вище, якому призначена інформація з полів даних пакета IP.

Поле " Контрольна перевірка заголовку пакета"

Поле "Контрольна перевірка заголовку пакета" (16 біт) використовується для контролю цілісності заголовку пакета IP.

Поля "Адреса відправника" та "Адреса одержувача"

Поле "Адреса відправника" (32 біти) — це IP-адреса відправника пакета.
Поле "Адреса одержувача" (32 біти) — це IP-адреса одержувача пакета.

Поле "Послуги"

Поле "Послуги" (довжина його змінна) застосовується для вказівки необов'язкових параметрів IP, пов'язаних, наприклад, із режимами безпеки або маршрутизації.

Поле "Доповнення (нулі) поля "Послуги" до 32-бітової межі" (довжина змінна) доповнює заголовок пакета таким чином, щоб він складав ціле число 32-бітових слів.

4.4.3. Адреси IP

Фізичні об'єкти (ГОМ, маршрутизатори, сервери, підмережі) у IP-мережі ідентифікуються за допомогою імен, яки називають ***IP-адресами***.

IP-адреси являють собою 32-бітові ідентифікатори, структура яких оптимізована для рішення основної задачі протоколу IP — маршрутизації. Звичайно, для зручності представлення IP-адрес використовується так зване цифрове написання IP-адрес, коли адреса записується як десяткове уявлення 4 байт, розділених крапками, наприклад, 192.171.153.60.

➢ Адреси IP та їх класи були розглянуті у попередньому розділі в пункті "3.6. Адресація в Internet".

4.4.4. Маршрутизація в IP-мережах

Найважливішою функцією протоколу IP є маршрутизація пакетів. У загальній постановці задачу маршрутизації можна сформулювати в такий спосіб: у деякої ГОМ є пакет із визначеною адресою одержувача. На який мережевий інтерфейс (по якій підмережі, ГОМ або якому маршрутизатору) потрібно передати цей пакет, щоб він дійшов до одержувача?

Блок-схема алгоритму маршрутизації пакетів у вузлі IP наведена на рис. 4.4.

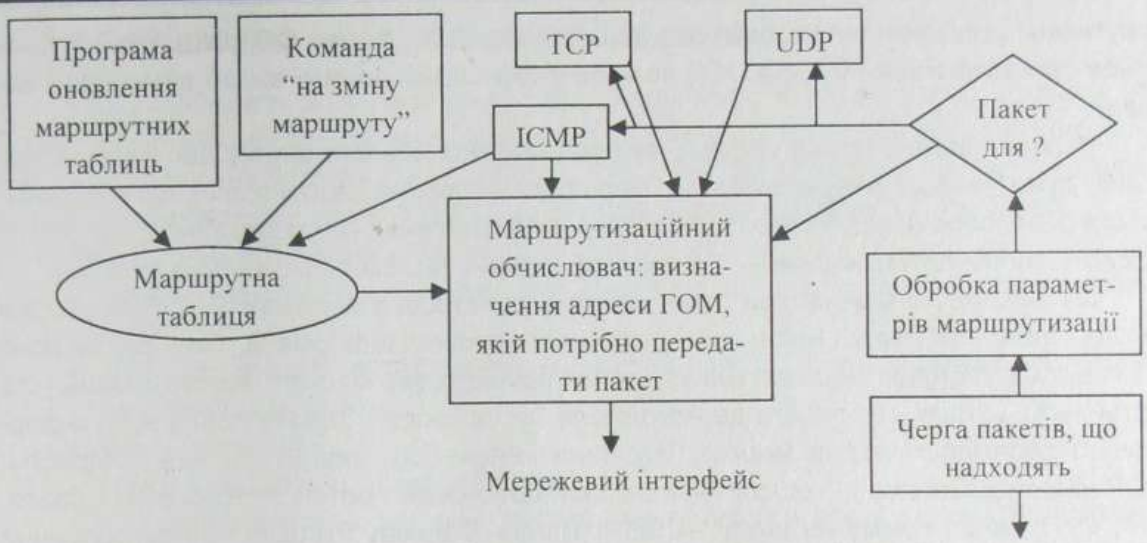


Рис. 4.4. Блок-схема алгоритму маршрутизації пакетів у вузлі IP

Центральним елементом цієї схеми є маршрутизаційний обчислювач. На його вхід надходять пакети від рівнів, що "лежать" вище, (протоколи TCP, UDP), від системи вирішення конфліктних ситуацій — ICMP (Internet Control Message Protocol) (наприклад, контрольні повідомлення про втрату пакета), від рівнів, що лежать нижче, через мережевий інтерфейс.

Маршрутизаційний обчислювач працює з маршрутною таблицею (*routing table*), що вказує маршрут передачі пакета з заданою адресою, тобто спрямовує або в деяку ГОМ (пряма маршрутизація), або деякому маршрутизатору (непряма маршрутизація), або в деяку підмережу. При цьому в маршрутній таблиці визначається фізичний мережевий інтерфейс, через який повинен бути переданий пакет (у ГОМ, що здійснює маршрутизацію, таких мережевих інтерфейсів може бути декілька).

Статична маршрутизація в Internet

Статична маршрутизація жорстко визначає задані маршрутні таблиці. Найбільш придатна вона для ГОМ, що працюють у невеликих мережах, оскільки ГОМ не повинна витрачати багато "сил" на задачі маршрутизації. Маршрутну таблицю в ній складають як невеликий список, що включає ГОМ-"сусідів" і "шлюз по замовчуванню" (*default gateway*), якому ГОМ і віддає всі пакети, маршрутизація яких у неї викликає якінебудь ускладнення.

Цей же "алгоритм маршрутизації на основі неповної інформації" застосовується при взаємодії маршрутизаторів. Маршрутизатор, що обслуговує, наприклад, локальну мережу, зобов'язаний знати усе про свою локальну мережу, проте інформацією про зовнішні мережі він може і не володіти, звертаючись при необхідності до зовнішніх маршрутизаторів, які можуть надати цю інформацію.

Динамічна маршрутизація в Internet

Сутність цього виду маршрутизації полягає у відновленні і коригуванні інформації в маршрутних таблицях конкретної ГОМ (або маршрутизатора) на основі обміну службовою інформацією із суміжними вузлами. Реалізує алгоритми динамічної мар-

шрутизації програма — так званий маршрутизаційний процес ОС UNIX, що запускається при завантаженні ядра ОС і працює у фоновому режимі аж до вимикання машини.

Рішення задачі маршрутизації, як пошуку оптимального шляху доставки інформації, дуже непросте через розподіл маршрутизаторів на базові (*core*), що підключаються безпосередньо до високошвидкісних магістральних каналів, та інші, що обслуговують сполучені підмережі — так звані автономні системи (*autonomous systems*).

Ця задача ускладнюється і тим, що маршрутизація в мережах IP здійснюється в умовах деякої апріорної непевності (на основі неповної інформації, тому що не можна внести в таблицю маршрутизації всю інформацію про мережу через глобальність останньої), а тому досяжність деякого вузла "не прозора". Задачу поширення інформації з досяжності вузлів мережі (усунення непевності) вирішують два протоколи: EGP (Exterior Gateway Protocol — зовнішній шлюзовий протокол, RFC-904) і протокол, що прийшов йому на зміну — BGP (Border Gateway Protocol — прикордонний шлюзовий протокол, RFC-1771).

При виборі маршруту, крім інформації про досяжність вузла, необхідно оптимізувати шлях до нього. Задачу оптимізації в Internet розв'язують ряд протоколів динамічної маршрутизації, які можна поділити на два основних класи, що засновані:

- на підрахунку проміжних ретрансляцій (*hop count*);
- на знанні повної топології мережі (SPF-протоколи — Shortest Path First).

Протоколи на основі підрахунку проміжних ретрансляцій обчислюють найбільш короткі шляхи розповсюдження, визначаючи число проміжних маршрутизаторів на шляху передачі пакета від даного маршрутизатора до одержувача. Це рішення традиційно було першим, але воно не оптимальне, тому що не враховує реальну пропускну спроможність ліній зв'язку.

На основі підрахунку ретрансляцій працюють такі поширені протоколи динамічної маршрутизації, як GGP (Gateway Gateway Protocol — між шлюзовий протокол, RFC-823), що забезпечує обмін інформацією між високопродуктивними базовими маршрутизаторами, підключеними безпосередньо до високошвидкісних магістральних каналів мереж (*backbones*), і RIP (Routing Information Protocol — протокол маршрутної інформації, RFC-1058), що вирішує ту ж задачу для менш значимих маршрутизаторів.

До протоколів другого типу (SPF) відноситься протокол HELLO, що обчислює оптимальний шлях на основі вимірювання мінімального часу затримки передачі пакетів, а також удосконалений протокол OSPF (Open SPF — відкритий SPF, RFC-1583), що передбачає ряд додаткових можливостей:

- обчислення оптимальних маршрутів для конкретних видів обслуговування;
- аутентифікацію вузлів на основі парольної системи;
- вирівнювання навантаження мережі;
- маршрутизацію в підмережах;
- мінімізацію часу пошуку інформації при широкомовних викликах;
- адміністрування віртуальних мереж, топологія яких не враховує фізичну природу каналів зв'язку.

4.4.5. Рішення конфліктних ситуацій в Internet

Якщо відбувається "позаштатна" подія (наприклад, втрата пакета), то для її обробки ініціалізується спеціальний протокол ICMP (Internet Control Message Protocol — протокол керуючих повідомлень в Internet, RFC-792). Він "з'ясовує" природу помилки, формує повідомлення про неї і передає його тій програмі, що передала пакет.

Протокол ICMP виконує наступні основні функції:

- обмін контрольними пакетами для з'ясовування наявності й активності вузлів мережі;
- аналіз досяжності вузла-одержувача, скидання пакетів, що спрямовуються до недосяжних вузлів;
- зміну маршрутів;
- синхронізацію часу у вузлах мережі;
- керування потоком (шляхом регулювання частоти посилки пакетів вузлами-джерелами).

Сутність процедури зміни маршрутів полягає в наступному. Якщо деякий маршрутизатор визначає, що ГОМ використовує неоптимальний шлях для доставки пакета, він за допомогою протоколу ICMP коригує маршрутну таблицю ГОМ. Це один із механізмів автоматичної оптимізації доставки інформації й адаптації мереж TCP/IP до змін топології.

4.5. Протоколи транспортного рівня TCP та UDP

Одержувачем повідомлення є прикладна задача (процес). Процеси змінюються динамічно: вони можуть створюватись і знищуватись. Більш того, при установці зв'язку з деяким процесом не можна бути упевненим у тому, що під час роботи він не буде перерваний або знищений (наприклад, унаслідок перевантаження комп'ютера).

Введення даних, необхідних процесу, і виведення даних здійснюється через логічні (програмно організовані) точки — ***порти***. Процес, як об'єкт, представляється сукупністю портів, через які він взаємодіє з іншими процесами мережі. Будь-яке звертання до процесу у віддаленій ГОМ здійснюється за допомогою адреси, що складається з двох частин: IP-адреси, що ідентифікує ГОМ, і номера порту, що ідентифікує процес.

Всі задачі можна умовно розділити на дві великі групи: відомі всім (*well-known*) та інші. До відомих відносяться задачі (або послуги), що одержали велике поширення. Для них існують заздалегідь визначені порти, закріплені в стандартах Internet. Це так звані добре відомі номери (*well-known numbers*). Виділенням номерів заздалегідь визначених портів займається організація IANA (Internet Assigned Numbers Authority — Агентство по виділенню імен в Internet).

При написанні власної програми в рамках локальної задачі можна вибрати будь-який порт (за винятком зарезервованих) і, знаючи його номер, обмінюватися інформацією по мережі. Природно, що локальність задачі в даному випадку означає обмеженість її розповсюдження серед комп'ютерів у рамках Internet.

В Internet заздалегідь "домовляються" про повну адресу локальної програми шляхом поширення інформації щодо імен комп'ютерів, які підтримують дану програму, і номерів зарезервованих для неї портів.

Визначення одержувача — одна з головних задач транспортних протоколів в Internet.

4.5.1. Протокол UDP

UDP (RFC-768) є дейтаграмним протоколом, що не гарантує доставку і не зберігає порядок надходження дейтаграм. Повідомлення протоколу UDP називають ***абодейтаграмою користувача*** (*user datagram*). Воно складається з заголовка і блоку даних. Заголовок дейтаграми користувача складається з чотирьох 16-бітових полів (рис. 4.4).

0 7	8 15	16 23	24 31
Адреса порту процесу-відправника		Адреса порту процесу-одержувача	
Повна довжина (в октетах) дейтаграми (заголовка і блоку даних) користувача		Контрольна сума	

Рис. 4.4. Формат заголовка дейтаграми протоколу UDP

Поля адрес

Поля "Адреса порту процесу-відправника" і "Адреса порту процесу-одержувача" визначають адреси портів процесу-відправника і процесу-одержувача. Поле "Адреса порту процесу-відправника" має конкретне значення лише в тому випадку, якщо процес-відправник повинен одержати відповідне повідомлення, інакше воно заповнюється нулями.

Поле "Повна довжина дейтаграми"

Поле "Повна довжина дейтаграми" указує повну довжину (в октетах) заголовка і блока даних дейтаграми користувача.

Поле "Контрольна сума"

Поле "Контрольна сума" містить контрольну суму. При її розрахунку враховуються також мережеві адреси. В цілому розрахунок контрольної суми здійснюється наступним чином:

1. Блок даних повідомлення доповнюється нулями до цілого числа 16-бітових слів.
2. Поле "Контрольна сума" заповнюється нулями.
3. Перед повідомленням розміщується псевдозаголовок, структура якого показана на рис. 4.5.

4. Розрахунок контрольної суми виконується по всій сукупності даних, після чого знімаються псевдозаголовки і доповнення нулями, значення контрольної суми розміщується у відповідному полі заголовка, а дейтаграма передається мережевому рівню (протокол IP).

0 7 8 15	16	23	24 31
Адреса IP відправника			
Адреса IP одержувача			
00000000	Код протоколу	Довжина повідомлення	

Рис. 4.5. Формат псевдозаголовку дейтаграми протоколу UDP

ГОМ-одержувач для перевірки контрольної суми дейтаграми робить аналогічні операції.

Розрахунок контрольної суми — операція необов'язкова. У випадку, якщо поле "Контрольна сума" заповнене нулями, то воно сприймається як відмова від розрахунку контрольної суми. Для випадку (рідкісного, але можливого), коли розрахована контрольна сума дорівнює нулю, усі біти поля "Контрольна сума" встановлюються в стан "1".

При відмові від розрахунку контрольної суми варто мати на увазі, що мережевий рівень (протокол IP) обчисляє свою контрольну суму лише для свого заголовку і не контролює цілісність блоку даних. Тому відмова від обчислення контрольної суми протоколу UDP доцільна лише в тому випадку, коли протокол фізичного рівня забезпечує достатньо надійний захист від помилок.

Таким чином, функція протоколу UDP зводиться до розподілу дейтаграм між процесами через відповідні порти і необов'язковий контроль цілісності даних.

4.5.2. Протокол TCP

На відміну від UDP, протокол TCP (RFC-793 і RFC-761) забезпечує повноцінну транспортну службу. Транспортна служба TCP:

- забезпечує доставку даних (при цьому процес передає протоколу дані у вигляді цілісного файлу);
- обробляє дані (не накладає ніяких обмежень на структуру даних);
- забезпечує буферизацію даних — це дозволяє стабілізувати вхідний трафік, що утворюється різноманітними процесами, шляхом вибору оптимального розміру повідомлення;
- забезпечує термінову передачу даних (нехай навіть одного байту);
- організує дуплексні віртуальні з'єднання за допомогою операції попереднього встановлення з'єднання;
- забезпечує можливість передачі керуючої інформації одночасно з потоком даних (*piggybacking*).

Структура блоку TCP

Блок TCP складається з заголовку та поля даних. Формат заголовка блоку TCP наведено на рис. 4.6.

0 7	8 15	16 23	24 31
Адреса порту процесу-відправника		Адреса порту процесу-одержувача	
Номер останнього байту, що передається			
Номер очікуванго байту повідомлення TCP, що йде за останнім вірно прийнятим			
Довжина заголовку	Зарезер-воване	Тип повідомлення	Значення довжини (в октетах) "ковзного" вікна
Контрольна сума			Показник закінчення передачі термінових даних
Послуги			Доповнення нулями до цілого числа 32-бітових слів

Рис. 4.6. Формат заголовка блоку TCP

Поля адрес

Поля "Адреса порту процесу-відправника" і "Адреса порту процесу-одержувача" використовуються для визначення адрес портів процесу-відправника і процесу-одержувача повідомлення.

Поля номерів байтів

Поле "Номер останнього байту, що передається" визначає номер останнього октету в переданому блоці TCP і служить для контролю за порядком проходження блоків і правильного відновлення послідовності блоків одержувачем.

Поле "Номер очікуваного байта повідомлення TCP, що йде за останнім вірно прийнятим" містить номер октету, який одержувач має намір прийняти наступним.

Поле "Довжина заголовку "

Поле "Довжина заголовку" визначає довжину заголовка блоку TCP, яка вимірюється в 32-бітових словах. Довжина заголовка блоку може змінюватися в залежності від значень, встановлених у полі "Послуги".

Поле "Зарезервоване"

Поле "Зарезервоване" — резервні біти (4 біти) для майбутнього (подальшого) використання.

Поле "Тип повідомлення"

Поле "Тип повідомлення" містить службові біти (6 бітів), що визначають тип повідомлення. Біти що розташовані зліва на право і, коли встановлені в "1", позначають:

- **URG** (*urgent*) — термінове повідомлення;
- **ACK** (*acknowledgment*) — "квитанція" на прийнятий блок даних;

- **PSH** (*push*) — вимога відправлення повідомлення без очікування заповнення буферу,
- **RST** (*reset*) — запит на повторне з'єднання;
- **SYN** (*synchronization*) — синхронізація лічильників (використовується при встановленні з'єднання);
- **FIN** (*finish*) — указує, що переданий останній байт

Поле "Значення довжини "ковзного" вікна

Поле "Значення довжини (в октетах) "ковзного" вікна служить для декларації приймального вікна.

Поле "Контрольна сума"

У полі "Контрольна сума" розміщується контрольна сума, розрахована по блоку і псевдозаголовку (розрахунок контрольної суми і сам псевдозаголовок аналогічні UDP, за винятком того, що в поле "Код протоколу" записується код TCP — "6").

Поле "Показник закінчення передачі термінових даних"

Поле "Показник закінчення передачі термінових даних" використовується разом з керуючим бітом **URG**. Число, що розміщено в цьому полі, вказує на кінець термінових даних. Термінові дані передаються позачергово (поза потоком).

Поле "Послуги"

Поле "Послуги" використовується для надання додаткових послуг, наприклад таких, як оптимізація передачі шляхом вибору максимального розміру блоку (MSS — Maximum Segment Size).

Поле "Доповнення нулями"

Поле "Доповнення нулями до цілого числа 32-бітових слів" використовується для доведення розміру заголовка до цілого числа 32-бітових слів.

Процедурна характеристика протоколу TCP

Процедурна характеристика протоколу TCP включає три фази інформаційного обміну: установлення з'єднання, передача даних і роз'єднання. Важливою особливістю процедурної характеристики TCP є те, що на всіх етапах обміну повідомленнями використовується тільки один формат блока, розглянутий вище. Розходження етапів визначається за допомогою кодування поля "Тип повідомлення".

Фаза установлення з'єднання TCP

Механізм установлення з'єднання наступний. Спочатку ПЗ, що реалізує TCP-протокол, завантажується і знаходиться в стані пасивного очікування (комп'ютер включений, але інформація не приймається і не передається).

Процес — ініціатор з'єднання — звертається до своєї ОС із запитом на встановлення з'єднання: на приймання або на передачу. Запит на приймання переводить протокол у стан очікування прийому, в якому протокол TCP очікує встановлення

з'єднання, а запит на передачу — у стан передачі повідомлення з'єднання, що ініціалізується. ОС виділяє процесу-ініціатору адресу порту.

Встановлення з'єднання проводиться в три етапи (рис. 4.7):

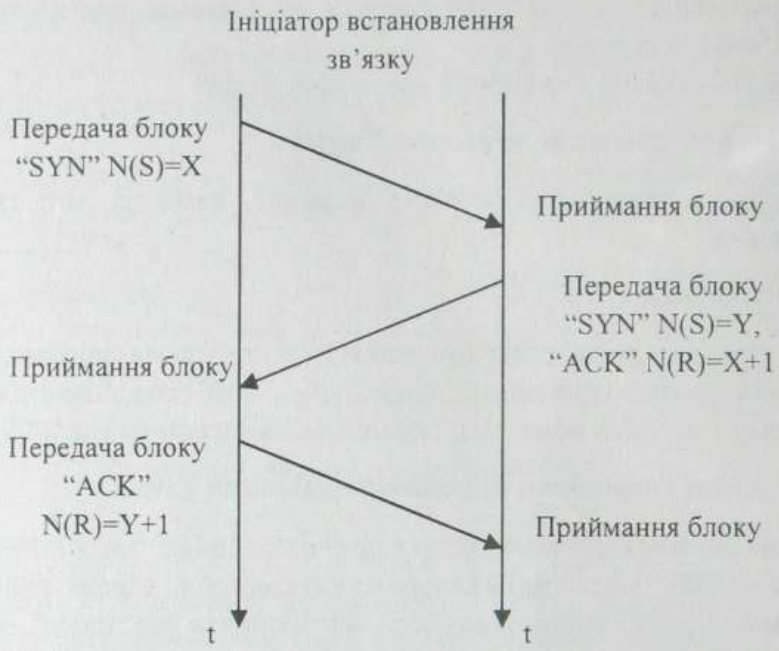


Рис. 4.7. Фаза установлення з'єднання TCP

Ініціатор з'єднання відправляє адресату блок із встановленим у стан "1" бітом "SYN" і передає в цьому повідомленні початкове значення лічильника N(S)=x для переданої їм послідовності блоків.

У цей час процес-адресат виконує функцію пасивного відкриття (*passive open*) з'єднання, вказуючи ОС, що очікує вхідний трафік, і одержує адресу порту. Далі, після одержання блока від процесу-відправника, сторона приймача встановлює лічильник прийнятої послідовності блоків у стан X і, щоб ініціатор переконався, що його вірно зрозуміли, установлює біт "ACK" поля "Тип повідомлення" відповідного повідомлення в стан "1", а лічильнику квитанції присвоює значення N(R)= X+1 (тобто вказує номер байта, котрий одержувач вважає необхідним прийняти). Крім того, у відповідному блоці приймач також встановлює керуючий біт синхронізації послідовності блоків.

Фаза передачі даних

Протокол TCP забезпечує надійну доставку інформації у тому розумінні, що він організує пряме підтвердження (кешування) коректного прийому інформації одержувачем.

Механізм простого квитування

У процесі доставки дані можуть бути загублені або зіпсовані, тому одержувач, якщо він прийняв блок, перевіряє його коректність шляхом розрахунку контрольної суми. Якщо остання правильна (дані не зіпсовані), то адресат відправляє "квитан-

цію"-підтвердження прийому; якщо контрольна сума не сходиться, то "квитанція" не висилається.

Очікування "квитанції" може бути необмеженим. Для виходу з такого стану використовується механізм тайм-ауту. Сутність його полягає в тому, що відправник, передаючи в канал блок, включає лічильник часу й очікує "квитанцію" протягом деякого тимчасового інтервалу (тайм-ауту) з моменту передачі. Після закінчення цього часу відправник вважає, що пакет загублений або зіпсований, і повторює передачу (позитивне квитування з повторною передачею).

Захист від перевантажень

Керування кешуванням методом "ковзного" вікна надає можливість керування потоком із метою запобігання перевантажень у мережі. Розмір вікна (поле "Значення довжини "ковзного" вікна у форматі блока TCP) є не що інше, як число байтів, спрямованих у мережу конкретним джерелом. Змінюючи розмір вікна для множини джерел інформації, можна ефективно управляти числом блоків, що існують у мережі, і за допомогою цього знімати перевантаження на окремих її ділянках.

Цей механізм використовується протоколом TCP для рішення двох цілком різнорідних задач захисту мережі від перевантажень.

Перша задача — ліквідація перевантаження на проміжних вузлах мережі. Її вирішують маршрутизатори, направляючи протоколам кінцевих станцій вимоги на зменшення розмірів вікон.

Друга задача — захист від перевантаження буферу самого протоколу TCP, що приймає дані. Одержувач, квитує деяку послідовність блоків, повідомляє відправнику, яку кількість байтів інформації він готовий безконфліктно прийняти. Тим самим забезпечується захист приймального пристрою від перевантаження (особливо це важливо у випадках, коли продуктивність джерела і приймача інформації істотно різняться). Цей метод називається ***декларацією приймального вікна*** (*window advertisement*). Якщо відправник "не справляється" з вхідним потоком, то він може декларувати вікно нульового розміру, відмовляючись від прийому інформації.

Відмова від прийому даних може увійти в суперечність із вимогою активних процесів зробити деякі термінові дії, наприклад негайно припинити передачу. У цьому випадку відправнику дозволяється послати дані з позначкою "Терміново", що одержувач зобов'язаний прийняти, нехай навіть йому введеться пожертвувати частиною наявних в приймальному буфері даних.

Метод декларування приймального вікна містить у собі логічне протиріччя, що виникає в тому випадку, якщо одержувач декларує нульовий розмір вікна для ліквідації свого перевантаження. Відправнику не можна посилати дані, оскільки одержувач не може їх прийняти, а одержувач не має повноважень проявити ініціативу і повідомити про те, що перевантаження ліквідоване. Розв'язується це протиріччя наступним чином: відправник інформації, що одержав декларацію вікна нульового розміру, має право періодично відправляти блоки з "нульовим" інформаційним полем, "нагадуючи про себе", а одержувач, квитуючи такий блок, може декларувати приймальне вікно ненульової довжини.

Фаза роз'єднання

Роз'єднання відбувається з ініціативи однієї із сторін, що посилає запит на завершення з'єднання.

4.6. Протоколи прикладного рівня: TELNET і FTP

Процедури транспортного інтерфейсу є основою для протоколів взаємодії процесів, що дозволяють реалізувати такі прикладні функції, як доступ терміналів до процесів, передачу файлів, електронну пошту тощо. Ці функції реалізуються за рахунок взаємодії як мінімум двох процесів, що виконуються одним або різними комп'ютерами, і відповідних протоколів: віддаленого доступу (зв'язок із терміналами), віртуального терміналу, передачі файлів та ін. Зазначені протоколи знаходяться над транспортним рівнем п'ятирівневої моделі Internet і називаються ***протоколами прикладного рівня***.

4.6.1. Протокол TELNET

Протокол віддаленого доступу TELNET описаний у RFC-854 (травень 1983 року). Його автори — Дж. Постел і Дж. Рейнолдс — у введенні до документу визначили призначення протоколу наступним чином: "... Дати загальний опис, наскільки це можливо, дуплексного, восьмибітового інформаційного обміну, головною метою якого є забезпечення стандартного засобу взаємодії кожної програми (процесу) із терміналами будь-якого типу, а також для організації взаємодій виду "термінал-термінал" і "процес-процес". TELNET є протоколом прикладного рівня і знаходиться над протоколами транспортного рівня (TCP, UDP).

Протокол TELNET дозволяє користувачу підключитися до будь-якої ГОМ (сервера) і працювати з ним із свого комп'ютера так, якби вона була віддаленим терміналом цієї ГОМ.

TELNET вирішує наступні основні задачі:

- визначає інтерфейс віртуального мережевого терміналу (NVT — Network Virtual Terminal), що подає структуру даних (породжуваних і відображених терміналом), алфавіт, керуючі символи і порядок обміну керуючою інформацією і даними, що дозволяє клієнту і серверу абстрагуватися від власних апаратних особливостей і врегулювати необхідний формат даних;
- регламентує порядок установки й узгодження необхідних параметрів при організації з'єднання й у процесі обміну інформацією;
- встановлює і підтримує дуплексний міжтермінальний зв'язок, що дозволяє клієнту і серверу рівноправно ініціювати передачу даних, узгоджувати параметри і т.ін.
- TELNET, відповідно до того, що він є прикладною програмою по відношенню до ОС локального серверу, дозволяє звертатись до віддаленої ОС або забезпечує функціонування клавіатури і монітору для іншого процесу на віддаленому сервері.

Як правило, TELNET розміщується на прикладному рівні, а не в ядрі ОС, що має свої переваги і недоліки. Головна перевага — реалізаційна взаємозалежність ОС та

протоколу, а недолік — деяка втрата у ефективності (зниження швидкодії в інтерактивному режимі).

Додаткові функції TELNET

При запуску ОС ГОМ установлення необхідних внутрішніх і зовнішніх параметрів відбувається автоматично, проте протоколом TELNET передбачено виконання додаткових сервісних функцій, що забезпечують більш точне узгодження необхідних параметрів терміналів (процесів) при веденні інформаційного обміну.

Набір регульованих параметрів достатньо широкий і включає кодування даних (7- або 8-бітове), режим передачі (дуплексний або напівдуплексний), тип терміналу і т.ін. Керуючі символи для установки параметрів у TELNET наведені в табл. 4.1.

Таблиця 4.1

Символ параметру	Десятковий код	RFC	Призначення
Transmit binary	0	856	Передача даних у двійковій формі
Echo	1	357	Луна-пакет на прийняття дані
Supress-Ga	3	858	Скасування команди "GA" після передачі даних
Status	5	859	Запит параметрів TELNET з віддаленого терміналу
Timing-Mark	6	860	Запит вставки тимчасових маркерів для синхронізації взаємодіючих процесів
Terminal-Type	24	884	Запит типу терміналу
End-Of-Record	25	885	Запит на завершення передачі даних кодом "EOR"
Linemode	34	1116	Установка локального редагування рядків і порядкової передачі на віддаленому терміналі

4.6.2. Протокол віддаленого доступу "rlogin"

Процедура узгодження параметрів рівнозначна в тому змісті, що сервер і абонент в однаковій мірі можуть ініціювати встановлення необхідних параметрів. Крім того, механізм узгодження параметрів припускає на різних кінцях з'єднання взаємодію різноманітних версій протоколу TELNET, тобто наповнених різноманітними наборами додаткових сервісних функцій (доходить до того, що один з партнерів не володіє ніякими додатковими функціями і базується тільки на NVT).

Розвитком протоколу віддаленого доступу в родині систем UNIX є протокол віддаленого доступу "rlogin" (RFC-1282). Цей протокол відрізняється від TELNET більш тісним зв'язком з ОС. Зокрема, для "rlogin" характерне розпізнавання прав доступу користувачів (він не потребує повторного введення паролю при послідовному доступі від однієї ГОМ до іншої, якщо користувач коректно виконав операцію входу на першу з доступних йому ГОМ).

Крім того, "rlogin" може використовувати стандартний вхід/вихід ОС віддалених ГОМ; інтерпретувати стандартні помилки ОС; розпізнавати параметри налаштування програмної оболонки як на локальній, так і на віддаленій ГОМ; передавати частину цих параметрів від локальної на віддалену ГОМ тощо.

4.6.3. Протокол FTP

Протокол FTP (RFC-959) забезпечує:

- програмний доступ до віддалених файлів — для роботи програм надається командний інтерфейс;
- інтерактивний доступ до віддалених файлів — користувач після виклику FTP потрапляє в інтерактивну оболонку, з якої за допомогою ряду команд може виконувати достатньо великий набір функцій,
- перетворення даних — FTP дозволяє клієнту описати формат збережених даних;
- аутентифікацію — FTP перевіряє ім'я користувача, його пароль і права доступу.

Протокол FTP, що базується на транспортній службі TCP, підтримує множину FTP-з'єднань із різноманітними абонентами. У свою чергу, будь-яке FTP-з'єднання складається з двох з'єднань: керуючого і доставки даних. Керуюче з'єднання підтримується від початку до кінця сеансу зв'язку з клієнтом. З'єднання доставки даних динамічно встановлюється на час передачі кожного файлу. Звичайно, обидва з'єднання підтримуються двома різноманітними процесами ОС.

Для підтримки керуючого з'єднання FTP використовується протокол TELNET у рамках NVT. Механізм зв'язку з портами відрізняється для клієнта і сервера. FTP-сервер працює на відомих портах 21 (керуюче з'єднання) і 20 (з'єднання доставки даних). FTP-абонент може вибрати й узгодити на локальному комп'ютері номери портів, що дозволяє йому підтримувати декілька FTP-з'єднань з одним сервером.

Команди протоколу FTP

Процедурна характеристика програмного інтерфейсу FTP досить проста. Абонент спрямовує серверу текстові команди, що складаються з імені команди і, можливо, параметрів. Команди інтерфейсу FTP наведені в табл. 4.2.

Таблиця 4.2

Команда	Параметри	Призначення
ABOR		Завершує з'єднання FTP і передачу даних
LIST	filelist	Завершує список файлів і каталогів
PASS	password	Передає пароль клієнта серверу
PORT	n1, n2, n3, n4, n5, n6	Передає серверу адреси протоколу IP (перші чотири байти) і адреси порту клієнта (останні два байти)
QUIT		Здійснює відключення від серверу
RETR	filename	Запитує прийом файлу із серверу
STOR	filename	Запитує передачу файлу на сервер
SYST		Запитує тип ОС серверу
TYPE	type	Призначає тип файлу: A — ASCII, I — зображення
USER	username	Передає ім'я користувача на сервер

На ці команди сервер посилає відповіді — текстові рядки, що для зручності використання в програмах починаються із символів тризначного коду, а далі йде текст,

що розкриває значення коду. Тризначний код відповіді інтерпретується порозрядно. Перший символ визначає загальний тип відповіді: позитивний (1...3) або негативний (4...5), проміжний (1, 2, 4) або остаточний (3, 5). Другий символ указує на причину помилки; третій конкретизує подію.

FTP для інтерактивного доступу

Для користувача інтерактивний FTP виглядає як самостійна "командна програмна оболонка". При виклику протоколу FTP за допомогою командного рядка з'являється запрошення `FTP>`, після якого можуть вводитися різноманітні команди (табл. 4.3).

Таблиця 4.3

Код команди	Призначення
`dir, Is`	Показує вміст віддаленого каталога
`Mdir, mis`	Показує вміст декількох каталогів
`cd`	Здійснює перехід в інший (вкладений) каталог
`cdup`	Здійснює перехід у каталог, що лежить вище
`led`	Змінює робочий каталог на локальній машині
`mkdir`	Створює каталог на сервері
`rmdir`	Знищує (видаляє) каталог на сервері
`delete`	Знищує файл на сервері
`Mdefete`	Знищує декілька файлів на сервері
`get`	Копіює файл із серверу на клієнтський комп'ютер
`mget`	Копіює декілька файлів із серверу на клієнтський комп'ютер
`put`	Копіює файл із клієнтського комп'ютера на сервер
`mput`	Копіює декілька файлів із клієнтського комп'ютера на сервер
`Rename`	Перейменовує файл на сервері
`Type`	Встановлює тип переданих даних
`Ascii`	Встановлює текстовий тип переданих даних
`Binary`	Встановлює двійковий тип переданих даних
`Close, disconnect`	Завершує FTP-сеанс
`Bye`	Завершує FTP-сеанс і виходить з оболонки

Для одержання інформації про всі (або про одну) команди FTP необхідно ввести команду `help ім'я_команди`. Якщо команда `help` викликана без параметрів, то виводиться перелік команд FTP; якщо ж зазначене ім'я команди, то з'явиться стисле пояснення її призначення.

На комп'ютері (з програмною оболонкою Windows) інтерактивний FTP реалізується по-іншому — частіше усього у вигляді двовіконного інтерфейсу, що відображає стан локальної і віддаленої машин і виконує обмежений набір команд FTP.

"Анонімний" FTP

Internet надає таку послугу, як "анонімний" файловий доступ (anonymous FTP), що не потребує від користувача імені. Як правило, він забезпечує доступ до серверів з безкоштовною загальнодоступною (*public domain*) інформацією. Клієнт може бути

не зареєстрований на такому сервері як користувач, тому йому пропонується увійти під умовним ім'ям "anonymous" і ввести в якості пароля адресу своєї електронної пошти (істинність якої може не контролюватись). Після цього абонент починає без перешкод обмінюватися інформацією із сервером.

Якщо "анонімний" файловий доступ до серверу заборонений, тоді зв'язатись з цим сервером можливо в сеансі TELNET під ім'ям користувача "guest", для якого пароль не потрібен.

4.6.4. Протокол TFTP

Для функціонування служби передачі файлів із використанням транспортного протоколу UDP застосовується протокол TFTP (Trivial FTP — простий протокол доставки файлів).

Відповідно до цього протоколу, усі файли передаються блоками по 512 байт. Перші два байти (октети) кожного блоку містять код операції. Для встановлення з'єднання використовуються два повідомлення: "Запит на читання файлу" і "Запит на запис файлу". Після прийому запиту починається передача даних.

Кожне повідомлення "Передача даних" містить номер блоку даних і самі дані (512 байт). На кожний блок даних сторона, що приймає, відповідає квитанцією "Підтвердження прийнятого блоку". При виникненні помилки генеруються спеціальна квитанція "Повідомлення про помилку". В результаті сторона, що приймає і передає, встановлюють тайм-аут на чекання такого блока. Якщо блок даних або квитанція губиться, то сторона, що передає, після закінчення тайм-ауту повторює передачу блоку, а сторона, що приймає, — передачу квитанції.

4.7. Протокол NFS

Подальшим розвитком служб передачі файлів є протокол мережевої файлової системи NFS (Network File System, RFC-1094), розроблений компанією Sun Microsystems. Він забезпечує множинний розподілений доступ до файлів через мережу в оперативному режимі (on-line). За допомогою NFS користувач (процес) практично не розрізняє локальні і віддалені файли. Коли процес звертається до файлу, ОС визначає, до якого локального або мережного файлу звертається клієнт, і переадресує керування відповідно локальній або мережевій файловій системі.

4.7.1. Віддалений виклик процедур і перетворення даних

Протокол NFS реалізується одночасно з двома іншими протоколами: віддаленого виклику процедур (RPC — Remote Procedure Call, RFC-1057) і зовнішнього уявлення даних (XDR — eXternal Data Representation, RFC-1014). Такий розподіл дозволяє розробнику прикладних програм незалежно звертатись до кожного з протоколів і використовувати лише необхідний набір функцій. У сукупності розробник одержує потужний інструмент для створення розподілених систем.

При розробці, наприклад, системи "клієнт-сервер" протокол RPC дозволяє визначити на станції користувача деякі процедури як віддалені. Ці процедури будуть включені в серверне ПЗ і декларовані як доступні для віддаленого виклику. Компіля-

тори автоматично включать у програму абонента код, що реалізує протокол RPC і відповідає за взаємодію розподілених частин системи.

Протокол XDR також полегшує розробку розподілених програм для різнорідних програмно-апаратних середовищ. Він бере на себе функцію врахування особливостей платформ, на яких працює. Наприклад, якщо клієнт і сервер використовують представлення цілих чисел із різноманітним порядком проходження молодших і старших символів (байтів), то XDR усуває дане розходження.

4.8. Електронна пошта

Електронна пошта (ЕП) — важлива інформаційна послуга в Internet. Це одне із самих масових засобів електронної комунікації. Практично кожен користувач Internet має свій "поштовий ящик".

Основними документами служби ЕП є стандарт RFC-821, який визначає протокол SMTP (Simple Mail Protocol — простий поштовий протокол) та RFC-822, який визначає формат поштового повідомлення.

4.8.1. Протокол SMTP

SMTP є протоколом прикладного рівня і працює разом із протоколами транспортного рівня TCP. Окрім вказаного протоколу (і разом із ним) використовується протокол Unix-Unix-CoPy (UUCP). Відмінність даних протоколів є в їх місії з точки зору ліній зв'язку, що застосовуються.

При використання SMTP програма відправки поштових повідомлень намагається встановити оперативний доступ (режим "on-line") до "поштової скриньки" абонента і передати листа до неї. Іншими словами, якщо мережа передачі даних у змозі забезпечити режим "on-line", тоді застосовується протокол SMTP. При відсутності такої можливості застосовується протокол UUCP. Останній реалізує засіб комутації повідомлень (принцип "stop-go"), при якому "лист", передається по ланцюжку через декілька поштових серверів, поки не досягне "скриньки" одержувача.

Процедурна характеристика протоколу SMTP

Взаємодія користувачів у рамках SMTP показана на рис. 4.8. Взаємодія будується по принципу дуплексного зв'язку, який встановлюється між відправником та одержувачем поштового повідомлення. При цьому відправник ініціює з'єднання і посилає запити на обслуговування, а одержувач відповідає на ці запити.

Дисципліни роботи і команди протоколу SMTP

Обмін повідомленнями й інструкціями в SMTP здійснюється в ASCII-коді. Протоколом визначено декілька видів (дисциплін) взаємодії:

- відправлення поштового повідомлення (MAIL);
- перенаправлення поштового повідомлення (forwarding);
- визначення наявності зазначеного користувача (веріфікація, VRFY і EXPN);
- пряма розсилка повідомлень (SEND, а також SGML — Send or Mail, SAML — Send and Mail);

- розсилка в режимі опосередкованого повідомлення;
- обмін "ролями" між відправником і одержувачем (TURN).

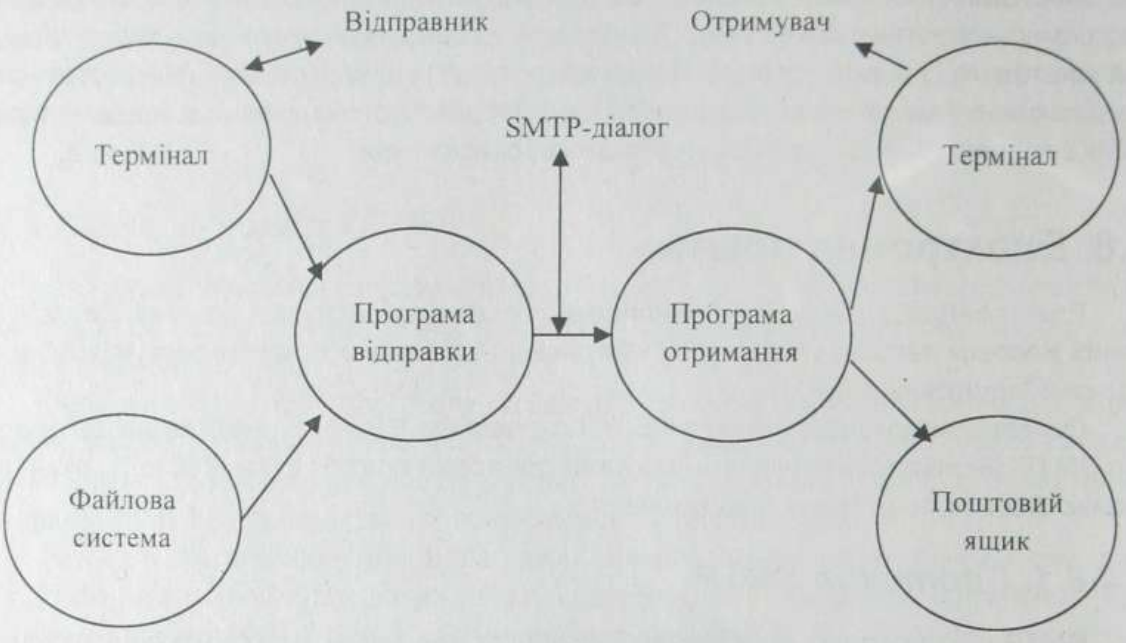


Рис. 4.8. Блок-схема інформаційної взаємодії по протоколу SMTP

Організація діалогу SMTP дуже нагадує організацію діалогу FTP. У SMTP також використовуються текстові команди клієнта, на які сервер відповідає рядком, що починається з відповіді у вигляді коду з трьох цифр. При передачі пошти (MAIL і SEND) використовується як мінімум набір команд, наведений в табл. 4.4:

Таблиця 4.4

Встановлення з'єднання	HELLO — визначає початок діалогу; MAIL — визначає відправника пошти; RCPT — визначає одержувача пошти
Фаза передачі даних	DATA — показує, що за нею з'явиться тіло повідомлення
Роз'єднання	QUIT — визначає завершення діалогу

Діалог ведеться через з'єднання TELNET на порт 25.

4.8.2. Стандарт MIME

Стандарт MIME (Multipurpose Internet Mail Extensions — розширення пошти Internet, RFC-1521) призначений для опису тіла поштового повідомлення в Internet. Він визначає розширення форматів даних тіла повідомлення у порівнянні з RFC-822, що припускає тільки рядки в ASCII-коді. MIME реалізується лише на частинах системи користувача, залишаючись цілком прозорим для всіх існуючих програм поштових агентів.

Типи даних, що підтримуються стандартом MIME, наведені в табл. 4.5.

Таблиця 4.5

Тип	Формат	Опис
Text	Plain	Неформатований текст
	Richtext	Текст із простою шрифтовою розміткою
	Enriched	Вдосконалений richtext
Multipart	Mixed	Послідовна обробка частин тіла повідомлення
	Parallel	Паралельна обробка частин тіла повідомлення
	Digest	Короткий зміст повідомлення
	Alternative	Складальне "тіло" повідомлення, в якому однакові семантичні частини
Message	RFC-822	Включення повідомлення в форматі SMTP
	Partial	Включення частини повідомлення
	External-	Включення вказівки на інше поштове повідомлення
Application	Octet-	Послідовність даних (октетів даних)
	Postscript	Постскриптум (коментарі)
Image	Jpeg	Графічний формат 5О 10918
	Gif	Графичний формат CompuServe
Audio	Basic	Звук в восьмибітовому форматі ISDN
Video	Mpeg	Відеоформат ISO 1 1 172

Стандарт MIME розроблений як розширювана специфікація, у якій припускається, що число типів даних буде рости в міру розвитку форм представлення даних. Безмежне збільшення таких типів неприпустимо. Кожен новий тип обов'язково повинний бути зареєстрований в IANA.

4.9. Система телеконференцій USENET

Internet є великим полем для практичного застосування багатьох інформаційних технологій. При цьому необхідно враховувати, що число останніх невпинно росте, і для кожної з них розробляється, досліджується і приймається відповідний стандарт (RFC).

Однією з найстарших таких технологій є система телеконференцій USENET, корені створення якої ідуть у 1979 рік, коли в Університеті штату Північна Кароліна два місцевих аспіранти Том Траскот і Джим Елліс на основі протоколу UUCP створили систему обміну новинами між двома комп'ютерами. Далі ця система стала активно розвиватися і у теперішній час набула форми глобальної розподіленої інформаційної системи.

4.9.1. Протокол NNTP

Протокол NNTP прийшов на зміну UUCP. Його метою було впорядкувати інформаційний обмін між серверами USENET. Головна відмінність NNTP від UUCP полягає в тому, що при використанні останнього всі нові повідомлення передавалися від

одного серверу до іншого, а потім — до наступного і т. ін. Це часто призводило до появи в мережі декількох однакових баз даних із новинами, що при цьому або нікому на даному сервері не були потрібні, або вже були в наявності.

Зрозуміло, що мережа в таких випадках завантажувалась вкрай неефективно. Протокол NNTP зазначені недоліки "усунув" за рахунок локального застосування інтерактивного режиму інформаційного обміну новинами між серверами, зміст якого зводиться до наступного: обмін інформацією про наявні новини по тій або іншій темі (предмету) і замовлення конкретних повідомлень проводяться лише у випадку наявності запитів на останні.

Протокол NNTP визначає:

- запитувально-відповідальний механізм обміну повідомленнями між серверами і безпосередньо між клієнтом (програмою перегляду новин) і сервером (процедурна характеристика протоколу);
- набір команд і відповідей на них у коді ASCII (логічна характеристика протоколу), при цьому кожна команда складається з ідентифікатора команди і параметрів.

Протоколом NNTP використовуються два типи серверів: центральний (*master*) і другорядний (*slave*). Центральний сервер забезпечує централізоване збереження новин у локальних мережах. На цьому сервері зберігаються списки користувачів груп новин і списки конференцій. Як правило, до даного типу серверів підключаються невеликі групи користувачів, що звертаються до серверу за допомогою спеціальних клієнтських програм для перегляду новин.

Другорядний сервер обслуговує велику кількість користувачів і зберігає тільки останні надходження новин. За всією іншою інформацією (по запитах) він звертається до центрального серверу. У цьому і полягає сутність "локального використання інтерактивного режиму", тобто користувач другорядного серверу звертається до нього в інтерактивному режимі, і якщо необхідної інформації немає, те сервер установлює з'єднання з необхідним сервером (тим, що тримає необхідну інформацію), і після цього користувач одержує доступ до необхідних тематичних новин.

По командах `ARTICLE`, `BODY`, `HEAD`, `STAT` запитуються статті або їхні частини. Існує два засоби запиту статті: по її ідентифікатору (вказується в заголовку) або по номеру статті в групі. Команда `ARTICLE` повертає заголовок і, через порожній рядок, — текст статті; `BODY` — лише тіло статті; `HEAD` — лише заголовок, а по команді `STAT` установлюється поточна позиція в групі по ідентифікатору статті. При цьому ніякої інформації не повертається.

По команді `GROUP` вибирається група новин. При цьому покажчик статті в групі встановлюється на перший запис у групі. По команді `HELP` можна одержати список дозволених для використання команд.

4.9.2. Протокол HTTP

Даний протокол є протоколом прикладного рівня і забезпечує обмін гіпертекстовою інформацією в Internet. HTTP реалізує різноманітні форми доступу, що базуються на URI (Uniform Resource Identifier — універсальний ідентифікатор ресурсу) у формі URL і універсальному засобі іменування інформаційних ресурсів URN (Universal Resource Name).

Відповідно до цього протоколу повідомлення по мережі передаються у форматі, схожому на поштове повідомлення Internet або формат повідомлень MIME. Протокол HTTP забезпечує взаємодію клієнтських програм і програм-шлюзів, що дозволяють доступ до інформаційних ресурсів ЕП Internet (SMTP), групам новин (NNTP), файловим архівам (FTP) і іншим системам.

У основі такого інформаційного доступу лежать спеціальні уповноважені (розміщені в серверах) програми (проксі), що дозволяють передавати інформацію між різноманітними інформаційними службами без втрат. Обмін даними в HTTP організований за принципом "запит-відповідь". Програма користувача (процес) ініціює взаємодію з уповноваженою програмою (сервером) і надсилає запит (містить метод доступу, адресу URI, версію протоколу, повідомлення про тип переданої інформації, інформацію клієнта і, можливо, тіло повідомлення клієнта), що має вигляд:

```
Повний_запит:=Рядок_запиту (Загальний_заголовок | Заголо-
вок_запиту | Заголовок_позначення_інформаційного_ресурсу)
символ_нового_рядка [тіло_інформаційного_ресурсу].
```

Наприклад: `POST http:://polyn.net.ki3e.su/cgi-bin/test HTTP/1.0`

На такий запит сервер (уповноважена програма) відповідає рядком стану (містить версію протоколу, код повернення і повідомлення, за формою схожий на MIME і містить інформацію серверу, метаінформацію і тіло повідомлення), що має вигляд:

```
Повна_відповідь:=Рядок_стану (Загальний_заголовок | Заголо-
вок_відповіді | Заголовок_інформаційного_ресурсу) сим-
вол_нового_рядка [Тіло_інформаційного_ресурсу].
```

Наприклад: `HTTP/1.0 200 Success`

При роботі в Internet для обслуговування HTTP-запитів використовується 80-й порт TCP/IP. Відповідно до алгоритму функціонування протоколу, користувач (процес) установлює з'єднання і чекає відповіді від серверу (процес, породжений уповноваженою програмою). Після відправки відповіді сервер ініціює роз'єднання. Отже, при передачі складних гіпертекстових сторінок з'єднання може встановлюватися декілька разів.

Форми доступу

У даному випадку під формою інформаційного доступу розуміють обсяг і отримання відповіді на запит. Найбільш поширеними формами доступу в Internet є:

- **GET** — у цьому випадку "замовник" одержує дані, представлені у форм URI у запиті на інформаційний ресурс;
- **HEAD** — використовується для одержання інформації про сам інформаційний ресурс;
- **POST** — використовується для передачі на сервер великого обсягу інформації (анотування існуючих ресурсів, посилка поштових повідомлень, робота з різноманітними формами інтерфейсів зовнішніх баз даних і програм).

4.9.3. Універсальний міжмережевий інтерфейс CGI

Загальний шлюзовий інтерфейс CGI (Common Gateway Interface) був розроблений Університетом штату Іллінойс (США). Головне його призначення — забезпечен-

ня єдиної форми інформаційного потоку й обміну даними між сервером і процесом (прикладною програмою).

CGI, з одного боку, визначає порядок (алгоритм) взаємодії сервера з прикладною програмою, при якому сервер є ініціатором, а з іншого боку — механізм реального обміну даними і керуючими командами в цій взаємодії.

Сутність інтерфейсу CGI полягає в тому, що це — набір спеціальних програм (CGI-сценарії), що виконують роль шлюзів при обміні даними з іншими інформаційними системами. Іншими словами, при звертанні користувача на сервер за конкретними даними останній запускає CGI-сценарій (шлюз) для коректного звернення за необхідною інформацією в іншу систему (базу даних).

➢ Приклади програмування для CGI наведені в наступному розділі.

4.10. Протокол мережевого керування SNMP

Керування ІОМ (інформаційно обчислювальна мережа) і будь-якою іншою телекомунікаційною мережею стало актуальним ще починаючи з першого телефону, коли всі з'єднання здійснювалися операторами.

З розвитком і ускладненням ІОМ системи керування такими мережами перетворювалися у виділені мережі керування. Інакше кажучи, для керування ІОМ виділяється спеціальний ресурс, і керуюча інформація циркулює в мережі "незалежно" від інформаційних потоків. Звичайно, термін "незалежно" — умовний, тому що службова інформація залежить від інформаційних повідомлень користувачів мережі в тому змісті, що службова (адреса, вид повідомлення, швидкість і ін.) частина повідомлення визначає керуючий інформаційний потік, але, з погляду формування і передачі останнього, ці два потоки незалежні.

При другому способі реалізації процесів керування такий ресурс не виділяється, а вся керуюча інформація передається усередині інформаційного потоку. Причому процес передачі керуючих команд (повідомлень) може бути організований одночасно (повідомлення передаються з визначеною періодичністю) або асинхронно (передаються в міру необхідності).

Таким чином, перший спосіб більш потужний по своїм можливостям і заснований на створенні і використанні системи спеціальних апаратно-програмних засобів керування конкретними пристроями СПД (системи передачі даних). Другий спосіб базується лише на обробці деяких (вибіркових) даних, що характеризують стан мережевого пристрою і надходять всередині інформаційного потоку. Обидва способи мають свої переваги і недоліки.

В Internet з метою створення єдиного підходу до керування устаткуванням, залученим до мереж IP, був розроблений простий протокол мережевого керування SNMP (Simple Network Management Protocol).

4.10.1. Загальна характеристика протоколу SNMP

З точки зору загальних теоретичних позицій, у процесі керування виділяють:

- об'єкт керування — ким управляють;
- суб'єкт керування — хто керує;

- прямий канал — середовище, по якому передається керуючий вплив суб'єкта на об'єкт;
- зворотній канал — середовище, по якому передається реакція об'єкта після керуючого впливу суб'єкту або його поточний стан.

Таким чином, сутність керування зводиться до цілеспрямованої діяльності суб'єкта по збору, накопиченню й обробки інформації про стан об'єктів, формуванню відповідних рішень і впливів, доведенню їх до об'єктів і одержанню результатів реагування об'єктів на керуючий вплив. Вочевидь, що ціль керування — домогтися максимальної ефективності функціонування об'єкта (його елементів) керування по заданим критеріям (критерію).

Протокол SNMP функціонує за принципом "запит-відповідь":

1. Суб'єкт керування формує і посилає об'єкту керування стандартне повідомлення-запит для одержання інформації про об'єкт.
2. Об'єкт керування формує відповідь на запит і посилає її суб'єкту.
3. Суб'єкт на основі отриманої інформації про стан об'єкта формує і посилає йому стандартне повідомлення про зміну параметрів.
4. Об'єкт керування, отримавши повідомлення на зміну своїх параметрів, посилає повідомлення-пприпинення і робить відповідну власну реконфігурацію.

Існують дві основні версії протоколу SNMP: власне SNMP, який звичайно називають першою версією (SNMPv1), і друга версія протоколу SNMP (SNMPv2). Також відома вся родина таких протоколів, серед яких захищений SNMP (Secure SNMP) і ін. Проте, ми зупинимося на більш загальних принципах функціонування на прикладі протоколу SNMPvl (SNMPv2) і розглянемо його специфічні особливості.

Кожна версія протоколу включає опис власне протоколу (засобу передачі даних) і опис об'єктів керування. Відповідно до концепції SNMP, керування в Internet будується на прямому, а не на прямому впливі на керований пристрій і базується на передачі непрямих параметрів.

Сутність такого підходу можна зрозуміти за допомогою наступного простого прикладу: суб'єкт не може "реально" здійснити перезавантаження ОС ГОМ або перенаправити переданий пакет інформації, але може встановити лічильник часу (таймер) на заданий час, після закінчення якого ОС буде перезавантажена, або внести зміни в маршрутну таблицю.

Для опису якогось об'єкта керування необхідно описати всі пов'язані з ним параметри, на які може впливати суб'єкт керування. Формальний опис об'єктів здійснюється за допомогою мови Abstract Syntax Notation One (ASN.1). Для завдання імені об'єкта вказується його ідентифікатор (OBJECT IDENTIFIER), що представляє собою послідовність цілих чисел.

Всі ідентифікатори організуються ієрархічно у виглядії деревоподібної структури. На верхньому рівні присутні три об'єкти: `iso`, `ccitt` і `joint-iso-ccitt`. У об'єкті `iso` є, наприклад, група ідентифікаторів для різноманітних організацій під ім'ям `org`. Вважається, хоча офіційно це і не зафіксовано, що першим елементом у підгрупі `dod` є Internet. Таким чином, об'єкт `internet` має ідентифікатор "1.3.6.1":

```
internet OBJECT IDENTIFIER ::= {iso(1) org(3) dod(6) 1}.
```

Для повного опису об'єкта необхідно зазначити не тільки ідентифікатор, але і його тип і змістовне визначення. Для цього в Internet існує база керуючої інформації MIB (Management Information Base). У ній описані всі об'єкти відповідно до граматики ASN. 1. У суб'єкта керування завжди є текстове уявлення даних, розміщених у MIB, що дозволяє йому не лише правильно інтерпретувати отримані дані, але й доводити до користувача інформацію про об'єкти в зручній для нього формі. Природно, об'єкт керування текстовим варіантом MIB не володіє.

Стандарт RFC-1907 визначає сукупність об'єктів, що є предметом розгляду в SNMP. У цій сукупності є частина об'єктів, що обов'язкова для будь-якої реалізації.

Існуючі версії протоколу SNMP підтримують роботу з базою MIB-II (друга версія). Групи `system`, `interfaces`, `ip` і `ictp` є обов'язковими. Інші групи необхідні для пристроїв, що підтримують відповідний протокол.

У Internet мають місце пристрої (ГОМ, маршрутизатори, сервери і т. ін.), що працюють по стандартах, відмінних від TCP/IP. Тому в структурі об'єктів стандарту RFC-1907 є спеціальна група `private (internet 4)`, що включає підгрупу `enterprises, privatel`. Така структура дозволяє будь-якій фірмі-виробнику, що бажає зробити свої об'єкти доступними по протоколу SNMP, описати їх на відповідному рівні ієрархії об'єктів керування.

Так, наприклад, компанія Sun Microsystems ввела у свої програмно-апаратні комплекси, що реалізують протокол SNMP, можливість керування ними шляхом зміни відповідних системних параметрів. Це дозволяє за допомогою протоколу SNMP запросити завантаження процесора, вільний дисковий простір і інші характеристики комп'ютерів, що забезпечує простоту побудови системи, що інформує персонал про перевантаження обчислювальних потужностей і дефіцит дискового простору.

4.10.2. Логічна характеристика SNMP (SNMPv1)

Протокол SNMPv1 описаний у стандарті RFC-1157. Повідомлення SNMP складається з трьох полів (рис. 4.9):

- "Version" — версія протоколу;
- "Community" — співтовариство;
- "SNMP PDU" — блок даних;
- "Community" — застосовується для обмеження доступу до керуючої системи.

<table>
<tr><td colspan="2">"Version"</td><td colspan="2">"Community"</td><td colspan="3">"SNMP PDU"</td></tr>
<tr><td>"PDU type"</td><td>"Enterprise"</td><td>"Agent address"</td><td>"Generic trap"</td><td>"Specific trap"</td><td>"Time stamp"</td><td>"Variables"</td></tr>
<tr><td>"Name 1"</td><td>"Value 1"</td><td>"Name 2"</td><td>"Value 2"</td><td></td><td>"Name N"</td><td>"Value N"</td></tr>
</table>

Рис. 4.9. Структура повідомлення SNMP (на прикладі повідомлення переривання "Trap")

Блок даних "SNMP PDU" починається з опису його типу ("PDU"), що і визначає структуру блока. У загальному вигляді блок даних складається з декількох обов'яз-

кових полів і довільного числа параметрів (імен змінних — "Name") і значень ("Value") параметрів.

В протоколі SNMPv1 визначені наступні типи повідомлень: "Get", "GetNext", "Sell", "GetResponse" і "Trap".

Повідомлення "Get" і "GetNext" призначені для запиту інформації. Повідомлення "Set" служить для зміни параметрів.

Повідомлення "GetResponse" посилається об'єктом керування у відповідь на будь-який запит суб'єкта керування і містить у собі інформацію про помилки і змістовну частину відповіді.

Повідомлення "Trap" (припинення) формується суб'єктом керування з яких-небудь "внутрішніх" причин. Зокрема, даний тип повідомлення використовується для перезавантаження ОС комп'ютера.

Для одержання службової інформації про об'єкт керування (повідомлення "Get" і "GetNext") формується повний ідентифікатор потрібного об'єкта, що доповнюється значенням "0". Наприклад, щоб одержати значення часу роботи системи після останнього перезавантаження (`sysUpTime` із групи `system`), у запиті формується ідентифікатор "1.3.6.1.2.1.1.3.0" (`iso org dod internet mgmt mib-2(1) system sysUpTime 0`).

Деякі об'єкти описуються не параметризовано, а подані у вигляді таблиць (матриць) — наприклад, таблиці маршрутизації. Доступ до таких об'єктів у SNMP здійснюється іншим способом: формується ідентифікатор, що складається з ідентифікатора елемента таблиці, до якого добавляється значення індексу (індексів).

Наприклад, для одержання фізичної адреси другого інтерфейсу пристрою, що знаходиться під керуванням, необхідно запитати значення змінної `tfPhysAddress` (6) із таблиці `ifTable` (2) групи `interfaces` (`mib-2 2`). Таблиця складається зі стовпчиків `ifEntry` (1). Повний ідентифікатор для запиту буде мати вид: "1.3.6.1.2.2.2.1.6." (`iso org dod internet mgmt mib-2(2) interfaces ifTable ifEntry ifPhys Address 2`).

Особлива роль у процесі керування відводиться повідомлення "GetNext", що дозволяють одержати інформацію про об'єкт, якщо немає в наявності його повної адреси. Наприклад, необхідно дізнатися адресу маршрутизатора, що використовується "по замовчуванню". Для цього повинен бути вказаний потрібний елемент таблиці з потрібним індексом, проте індекс невідомий.

Повідомлення "GetNext" дозволяє запросити такий елемент в ієрархічній структурі об'єктів керування. Іншими словами, у даному повідомленні вказується ідентифікатор групи (або підгрупи) об'єктів, а у відповідь надходить інформація про весь масив об'єктів, у якому обов'язково буде знаходитись IP-адреса шуканого маршрутизатора.

У процесі функціонування мережі може виникнути необхідність в активізації самого керованого пристрою. Звісно, що така ситуація можлива при непрямому впливі суб'єкта на об'єкт керування (наприклад, коли потрібно перезавантаження пристрою) або при виникненні позаштатного режиму роботи самого об'єкта (наприклад, коли має місце розрив зв'язку). У цьому випадку об'єкт керування передає повідомлення "Trap" (припинення), що інформує суб'єкт про позаштатну ситуацію.

Для повідомлення "Trap" передбачені сім видів (кодів) припинень. Наприклад, код "7" позначає вид припинення, що використовується для апаратно-програмних

комплексів конкретної фірми-виробника. У формат повідомлення "Trap" (див. рис. 4.9) також входять адреса агента (поле "Agent address"), який передав повідомлення, час відправлення повідомлення (поле "Timestamp"), код фірми-виробника апаратури (поле "Enterprise") і додаткова інформація, що складається з довільного числа пар "параметр" (поле "Name") - "значення параметру" (поле "Value"). У такий спосіб може бути передана інформація, наприклад, про зникнення зв'язку в каналі.

4.10.3. Недоліки протоколу SNMPv1

Серед недоліків протоколу SNMPv1 можна виділити наступні.

Незахищеність

Практично протокол SNMPv1 не має механізму захисту від несанкціонованого доступу до об'єктів керування, що дозволяє порушнику безперешкодно впливати на елементи IOM

Висока надмірність службової інформації

Для виконання функцій контролю і керування великим числом об'єктів, суб'єкт повинний формувати і передавати велику кількість повідомлень-запитів із великим числом аналізованих параметрів. Це призводить до додаткового навантаження на IOM (необхідний достатньо великий ресурс мережі для обміну службовою інформацією) і може істотно впливати на ефективність (пропускну спроможність) функціонування всієї мережі.

4.10.4. Протокол SNMPv2

На зміну протоколу SNMPv1 прийшов SNMPv2, що частково усуває недоліки першої версії (зокрема, посилена безпека протоколу) через великий вплив на нього протоколів Secure SNMP і віддаленого моніторингу в мережі (RMON).

В даний час SNMPv2 передбачає використання наступних баз даних: MIB-II, RMQN, M2M Остання з зазначених баз характеризує об'єкти керування, за допомогою яких можлива побудова розподіленої системи керування.

З точки зору структурування даних у SNMPv2 всі об'єкти керування подані у формі ієрархічної деревоподібної структури. Водночас склад об'єктів істотно розширився. У SNMPv2 (крім MIB-II) додані об'єкти, що регламентують обмін даними по протоколу SNMP, введені описи посередників, що управляють IOM, засобів моніторингу і розмежування доступу.

З метою скорочення обсягів переданої службової інформації, SNMPv2 передбачає новий тип повідомлення-запиту — "масовий" (*bulk*), в якому можуть указуватися дві групи об'єктів (параметрів). При відповіді на кожний об'єкт (параметр) першої групи видається наступний за ним об'єкт (параметр) і його значення, а при відповіді на кожний об'єкт (параметр) другої групи — всі наступні за ним об'єкти (параметри) і їхні значення відповідно до ієрархії об'єктів керування (із погляду побудови ідентифікаторів).

Для підвищення захищеності протоколу в SNMPv2 передбачається розмежування доступу до керування IOM. В зв'язку з цим протоколом визначена група мере-

жевих пристроїв, що має характерні для неї спеціальні атрибути, серед яких: адреса IP учасника групи, алгоритми аутентифікації, шифрування й ін. Власне кажучи, сама назва групи характеризує конкретний набір її атрибутів

Також для забезпечення безпеки в IOM визначається множина об'єктів керування, до яких необхідно мати доступ. Ці об'єкти перераховані в базі даних перегляду (View Database Group), і їм присвоюються символічні імена, а в базі даних привілеїв доступу (Access Privilege Database Group) визначена відповідність між цими іменами і назвами (іменами) груп.

Кожне повідомлення-запит SNMP перевіряється на предмет правомірності доступу відправника до потрібної інформації (перевірка відбувається за допомогою локальної бази даних, що містить адреси суб'єктів керування). Більш того, повідомлення-запит може бути зашифровано. В даний час аутентифікація здійснюється, як правило, за допомогою алгоритму MD5, а шифрування — алгоритму DES.

Одними з найважливіших етапів процесу керування є збирання, накопичення і обробка інформації про стан керованих об'єктів. Ці функції в SNMPv2 реалізуються на основі системи моніторингу. Центральним елементом системи моніторингу є база Manager-to-Manager MIB (RFC-1451), основу якої складає група `snmpM2Mobjects`, яка включає три таблиці і декілька параметрів.

Таблиця `alarm` визначає умови (значення параметрів), при яких необхідна реакція суб'єкта керування з метою відновлення нормального функціонування елементів IOM. Кожній умові (параметру) привласнене ім'я (назва) і дається його стисла характеристика, крім того, встановлюється інтервал часу, протягом якого необхідне "спостереження" за зміною даного параметру у "об'єкта, що спостерігається" (адреса якого задається власним ідентифікатором) з метою визначення необхідності впливу на керований об'єкт.

Дві інші таблиці визначають зміст і порядок дій суб'єкту керування при позаштатних ситуаціях, що виникають на об'єкті керування і наведені в таблиці `alarm`.

Використовуючи інформацію групи `nmpM2Mobjects`, можна побудувати такий алгоритм поведінки об'єкта керування, при якому він буде сповіщати суб'єкта керування про події, що відбуваються в ньому, які потребують втручання, що у свою чергу звільняє від постійного запитування суб'єктом об'єкта. Тому кожний об'єкт керування, що підтримує SNMPv2, може функціонувати подвійно: з одного боку, залишатися об'єктом, а з іншого боку — виконувати функції суб'єкта.

4.11. Ретрансляція кадрів (Frame Replay)

Ретрансляція кадрів (FR — Frame Relay) — це метод синхронної доставки повідомлень у мережах передачі даних із комутацією пакетів. Спочатку розробка стандарту FR була орієнтована на цифрові мережі інтегрального обслуговування (ISDN — Integrated Services Digital Networks), проте пізніше стало ясно, що FR може бути застосований в якості комунікаційного стандарту і в інших широкомасштабних СПД. До числа його переваг насамперед необхідно віднести малий час затримки, простий формат кадрів, що містять мінімум керуючої інформації, незалежність від протоколів верхніх рівнів EMBBC (еталонна модель взаємодії відкритих систем) і ін.

Розробкою і дослідженням стандартів FR у даний час займаються чотири міжнародні організації: Форум Ретрансляції Кадрів (FRF — Frame Relay Forum), ANSI (American National Standards Institute — Американський національний інститут по стандартизації), Міжнародна спілка електрозв'язку (ITU-T) і "Група Чотирьох" — G4 (Digital Equipment Corporation, Northern Telecom, Stratacom і Cisco Systems).

Оригінальний стандарт G4 (1990 р.), що дійсно поклав початок промисловому застосуванню FR, був заснований на проекті стандартів ANSI і мав невеликі відмінності від останнього. Стандарти FRF і ITU-T, що з'явилися декілька пізніше, ідентичні стандартам ANSI

Будь-який міжнародний стандарт завжди має (і завжди буде мати) величезний набір прикладних реалізацій у різноманітних СПД, що, як правило, призводить до створення багатьма фірмами-виробниками апаратно-програмних засобів, що найчастіше несумісні. Багатьма міжнародними організаціями починалися спроби подолання такої ситуації, Результатом однієї такої спроби, початої FRF, є проект, що узгоджує набір стандартів ANSI, обов'язковий для виконання членами FRF і реалізації більшістю фірм-виробників. У січні 1992 року ця угода була дороблена Технічним комітетом FRF і затверджено зборами членів FRF.

4.11.1. Логічна характеристика протоколу FR

FR є біт-орієнтованим синхронним протоколом, що використовує "кадр" у якості основного інформаційного елемента, і в цьому розумінні дуже схожий на протокол HDLC. Проте FR не забезпечує усі функції протоколу HDLC, і тому багато які з елементів кадру FTOLC виключені з основного формату кадру FR (у кадрі FR адресне поле і поле керування HDLC об'єднані в одне адресне поле).

Прапорець обов'язковий при формуванні будь-якого кадру FR.

Поле адреси містить наступні біти:

- **Біт "опитуванння/фінал"** (Command/Response — CR). Цей біт протоколом FR не використовується, але може застосовуватись в програмах користувача і "прозоро пропускається" апаратно-програмними засобами мережі FR.
- **Біт розширення адреси** (Extended Address — EA). Можливе розширення адресного поля (заголовку) на ціле число додаткових октетів із метою вказівки адреси, що складається з більш, ніж 10 бітів. Біт EA встановлюється наприкінці кожного октету заголовка, і якщо він має значення "1", то це означає, що даний октет у заголовку останній. Стандарт FRF рекомендує використовувати заголовок, що складається з двох октетів. У цьому випадку біт EA в першому октеті буде встановлений в "0", а в другому — в "1".
- **Біт повідомлення (сигналізації) приймача про явне перевантаження** (Forward Explicit Congestion Notification — FECN). Цей біт встановлюється в "1" апаратурою каналу даних (АКД) мережі для повідомлення одержувача повідомлення про те, що відбулося перевантаження в напрямку передачі кадру, що містить цю ознаку. Біт FECN встановлюється АКД мережі FR і не є обов'язковим для терміналів абонентів.
- **Біт повідомлення (сигналізації) джерела про явне перевантаження** (Backward Explicit Congestion Notification — BECN). Цей біт встановлюється в "1" АКД мережі для повідомлення джерела про те, що відбулося переванта-

ження в оберненому напрямку щодо напрямку передачі кадру, який містить цю ознаку. Біт BECN встановлюється АКД мережі FR і не є обов'язковим для терміналів абонентів.

- **Біт дозволу скидання** (Discard Eligibility — DE). Цей біт встановлюється в "1" у випадку явного перевантаження вхідного трафіку і вказує на те, що даний кадр може бути знищений у першу чергу стосовно інших кадрів, що не мають даної ознаки. Біт DE може бути встановлений у "1" або АКД мережі FR, або користувачем (тобто користувачу надане право вибирати, якими кадрами "він може пожертвувати"), при цьому повторна установка не припускається. Це передбачено для того, щоб вузли комутації мережі FR могли знищувати при перевантаженнях не лише кадри з установленим бітом DE.
- **Інформаційне поле**. Інформаційне поле містить дані користувача і складається з цілого числа октетів. Максимальний розмір для цього поля визначений стандартом FRF і складає 1600 октетів (мінімальний розмір — 1 октет).
- **Перевірочна послідовність кадру** (Frame Check Sequence — FCS). Перевірочна послідовність кадру використовується для виявлення можливих помилок при його передачі і складається з двох октетів

Всі зазначені вище поля повинні бути присутнім у кожному кадрі FR, що передається між двома замикаючими системами.

Протокол FR не передбачає передачу сигнальних повідомлень: немає командних кадрів (супервізорних), як у HDLC. Для передачі службової інформації використовується спеціально виділений канал сигналізації (ВКС), усередині якого передаються супервізорні кадри.

Інше важливе розходження між FR і HDLC — відсутність будь-якої нумерації послідовності переданих (прийнятих) кадрів. Це є результатом того, що протокол FR не має ніяких механізмів для підтвердження правильно прийнятих кадрів.

4.11.2. Процедурна характеристика протоколу FR

Протокол FR дуже простий у порівнянні з HDLC і включає невеликий набір правил і процедур організації інформаційного обміну. Основна процедура протоколу наступна: якщо отриманий кадр не зіпсований, то він повинен бути спрямований далі по призначенню відповідним маршрутом. Якщо виникають проблеми, пов'язані з навантаженням у межах мережі, то вузлам мережі FR дозволяється відмовитися від яких-небудь кадрів, щоб знизити гостроту проблеми.

Вузлу зв'язку мережі FR дозволено знищувати зіпсовані кадри, не повідомляючи при цьому користувача. Зіпсованим вважається кадр, у якого:

- немає коректного обмеження прапорцями;
- у складі менше, ніж п'ять октетів між прапорцями;
- у складі немає цілої кількості октетів після знищення бітів прозорості;
- поле адреси зіпсовано;
- містить неіснуючий DLCI;
- перевищений припустимий максимальний розмір.

Проте в деяких варіантах стандартів FR можливе примусова обробка кадрів, що перевищують припустимий максимальний розмір.

Обов'язкові процедури FR:

- "міжкадрове заповнення" прапорцями в моменти відсутності даних для передачі;
- резервування одного DLCI для інтерфейсу локального керування і сигналізації (LMI — Local Management Interface);
- утримання поля даних користувача в будь-якому кадрі не повинно піддаватись будь-якій обробці з боку мережі FR, за винятком даних LMI.

У мережах FR для високої ефективності використання пропускної спроможності фізичних ліній і каналів зв'язку і з метою запобігання перевантаження вузлів зв'язку і всієї мережі FR використовується метод статистичного мультиплексування кадрів. Він полягає в постійному "спостереженні" з боку мережі FR за потоком заявок від користувачів на передачу повідомлень, поточним станом завантаження мережі (ліній, каналів і вузлів зв'язку), перерозподілом вільного (і що визволяється) ресурсу пропускної спроможності між реальними потребами в ній абонентів і наданні останнім каналів інформаційного обміну. Даний метод забезпечує синхронну передачу повідомлень користувача у високошвидкісний канал зв'язку на основі попередніх домовленостей.

Використання попередніх домовленостей відбувається наступним чином:

- по-перше, абонент вибирає (і оплачує) пропускну спроможність порту і гарантованої швидкості передачі даних;
- по-друге, вузол доступу до мережі FR (FRAD) вимірює реальний споживаний абонентом ресурс пропускної спроможності каналу зв'язку;
- по-третє, якщо цей ресурс, перерахований у реальну швидкість передачі інформації, не перевищує CIR, то кадри передаються без змін; якщо перевищує, але не більш, ніж пропускна спроможність порту, то в кадрах FRAD біт DE встановлюється в "1", що дозволяє їх видалити при виникненні перевантаження (це право має й абонент, який сам вирішує, які кадри для нього найменш важливі); якщо перевищує пропускну спроможність порту, то кадри знищуються незалежно від будь-яких умов.

Розділ 5

Програмування для Internet

"Якщо факти не підтверджують теорію, їх треба вилучити"
(Закон Майєрса)

"Теорія — бездоганна, якщо її не можна перевірити... Практикою"
(Народна мудрість)

Популярність Internet зростає з кожним днем, а разом з нею зростають і вимоги до ресурсів та якості їх представлення. Тепер інтереси користувачів не обмежуються електронною поштовою скринькою і пошуком якої-небудь інформації в WWW, вони прагнуть створювати власні Web-сторінки, навіть Web-вузли. Новачки організують прості HTML-сторінки, не звертаючи особливої уваги на стиль і дизайн. Але, згодом, у них з'являється бажання зробити щось ефектне і цікаве. І тоді новоспеченому Web-майстру доводиться більш досконало вивчати мови розмітки (DHTML, HTML, XML, VRML, XHTML) і мови створення клієнтських сценаріїв (JavaScript, VBScript).

Згодом у нього можуть з'явитись і нові вимоги (наприклад, поспілкуватись із відвідувачами свого вузла). І в цьому випадку його вже не задовольнить, якщо в тіло сторінки просто додасться посилання на його поштову скриньку. Так, він вирішить, що непогано було б одержати досить повну інформацію про відвідувачів вузла (імена, e-mail, телефони, факси, адреси тощо), підрахувати кількість відвідувань, зібрати різні думки і, нарешті, створити базу даних. Але для реалізації всього цього HTML і, навіть, VBScript чи JavaScript стає недостатньо, тому що ці мови призначені для опису сценаріїв, що виконуються на комп'ютері клієнта. Отже, потрібно мати хоча б мінімальне уявлення про принципи виконання програм та архітектуру Internet.

На сучасному етапі розвитку Internet-технологій використовується клієнт-серверна архітектура. Тобто, комп'ютери мережі можна поділити на дві групи: сервери та клієнти. Клієнти — це комп'ютери користувачів, з яких вони виконують навігацію на просторах Internet в пошуках певних ресурсів. Сервери — це комп'ютери, на яких містяться Web-сторінки та інформаційно-програмні ресурси.

В зв'язку з цим, мови програмування для Internet можна розподілити на дві основні групи: мови опису сценаріїв — завантажуються на комп'ютер користувача разом із сторінкою і на ньому виконуються, та мови динамічного створення сторінок та обробки інформації — більш потужні, розміщуються та виконуються на сервері. Останні, в свою чергу, використовують технологію CGI, що реалізовує інтерфейс обміну інформацією між клієнтом і програмою-обробником. Ці мови називають універсальними сценарними мовами. До їх складу входять PHP, Perl, Java, C++, Python, Tcl тощо

Слід зауважити, що на даний момент ведуться розробки нових Internet-технологій і, можливо, через декілька років вже забудуть про архітектуру "клієнт-сервер" і мови програмування, що з нею пов'язані дуже тісно. Одним з прикладів є розробки

технології .NET, яка передбачає рівноправний обмін між комп'ютерами мережі. Також ведуться розробки і нових програмних засобів (мова Curl).

5.1. Огляд мов програмування для Internet

5.1.1. HTML

Мова гіпертекстової розмітки HTML (HyperText Markup Language) була запропонована Тімом Бернерсом-Лі в 1989 році в якості одного з компонентів технології розробки розподіленої гіпертекстової системи World Wide Web.

Коли Т. Бернерс-Лі запропонував свою систему, у світі інформаційних технологій спостерігався підвищений інтерес до нового і модного у той час напрямку — гіпертекстових систем. Сама ідея, але не термін, була введена В. Бушем у 1945 році в пропозиціях щодо створення електромеханічної інформаційної системи Memex. Незважаючи на те, що Буш був радником по науці президента Рузвельта, ідея не була реалізована. У 1965 році Т. Нельсон ввів у обіг сам термін "гіпертекст", розвинув і навіть реалізував деякі ідеї, пов'язані з роботою над "нелінійними" текстами.

У 1968 році винахідник маніпулятора "миша" Д. Енжильбард продемонстрував роботу із системою, що має типовий гіпертекстовий інтерфейс, і, що цікаво, проведена ця демонстрація була з використанням системи телекомунікацій. Однак виразно описати свою систему він не зміг.

У 1975 році ідея гіпертексту знайшла втілення в інформаційній системі внутрішнього розпорядку атомного авіаносця "Карл Уїнстон". Роботи в цьому напрямку продовжувались, і час від часу з'являлись реалізації типу Hypercard фірми Apple чи HyperNode фірми Xerox. У 1987 була проведена перша спеціалізована конференція Hypertext'87, матеріалам якої був присвячений спеціальний випуск журналу "Communication ACM".

Ідея гіпертекстової інформаційної системи полягає в тому, що користувач має можливість переглядати документи (сторінки тексту) у тому порядку, у якому йому це більше подобається, а не послідовно, як це прийнято при читанні книг. Тому Т. Нельсон і визначив гіпертекст як нелінійний текст. Досягається це шляхом створення спеціального механізму зв'язку різних сторінок тексту за допомогою гіпертекстових посилань, тобто в звичайного тексту є посилання типу "попереднє-наступний-попередній", а в гіпертексті можна побудувати ще як завгодно багато інших посилань. Улюбленими прикладами фахівців з гіпертексту є енциклопедії, Біблія, системи типу "Help".

Простий, на перший погляд, механізм побудови посилань виявляється досить складною задачею, тому що можна побудувати статичні посилання, динамічні посилання, асоційовані з документом у цілому чи тільки з окремими його частинами, тобто контекстні посилання. Подальший розвиток цього підходу приводить до розширення поняття гіпертексту за рахунок інших інформаційних ресурсів, включаючи графіку, аудио- і відео-інформацію, до поняття гіпермедіа.

Розробники HTML повинні були вирішити дві задачі:

- надати дизайнерам гіпертекстових баз даних простий засіб створення документів;

- зробити цей засіб достатньо могутнім, щоб відобразити уяви щодо інтерфейсу користувача гіпертекстових баз даних.

Перша задача була вирішена за рахунок вибору теговой моделі опису документу. Така модель широко застосовується в системах підготовки документів для друку. Прикладом такої системи є добре відома мова розмітки наукових документів Tex, запропонована Американським Математичним Суспільством, і програми її інтерпретації.

До моменту створення HTML існував стандарт мови розмітки друкованих документів — SGML (Standard Generalised Markup Language), що і був узятий за основу HTML. Передбачалось, що таке рішення допоможе використовувати існуюче програмне забезпечення для інтерпретації нової мови. Однак, будучи доступним широкому колу користувачів Internet, HTML зажив своїм власним життям. Імовірно, багато адміністраторів баз даних WWW і розробники програмного забезпечення для цієї системи мають досить неясну уяву про стандартну мову розмітки SGML.

Мова HTML дозволяє визначати структуру електронного документа з поліграфічним рівнем оформлення; результуючий документ може містити найрізноманітніші теги: ілюстрації, аудио- і відео-фрагменти і т. ін. Мова має в своєму складі розвинуті засоби для специфікування декількох рівнів заголовків, шрифтових виділень, різних груп об'єктів, наприклад, словники, чи каталоги меню для розміщення ілюстрацій і інших фрагментів, а також безліч інших можливостей.

Другим важливим моментом, що вплинув на долю HTML, став вибір в якості основи гіпертекстової бази даних звичайного текстового файлу, що зберігається засобами файлової системи операційного середовища комп'ютера. Такий вибір був зроблений під впливом наступних факторів:

- такий файл можна створити в будь-якому текстовому редакторі на будь-якій апаратній платформі в середовищі будь-якої операційної системи;
- до моменту розробки HTML існував американський стандарт для розробки мережевих інформаційних систем — Z39.50, у якому в якості одиниці зберігання вказувався простий текстовий файл у кодуванні LATIN1, що відповідає US ASCII.

Таким чином, гіпертекстова база даних у концепції WWW — це набір текстових файлів, розмічених мовою HTML, що визначає форму представлення інформації (розмітка) і структуру зв'язків цих файлів (гіпертекстові посилання).

Такий підхід припускає наявність ще одного компонента технології — інтерпретатора мови. У WWW функції інтерпретатора розділені між сервером гіпертекстової бази даних і інтерфейсом користувача.

Сервер, крім доступу до документів і обробки гіпертекстових посилань, здійснює також препроцесорну обробку документів, у той час як інтерфейс користувача здійснює інтерпретацію конструкцій мови, зв'язаних із представленням інформації.

Якщо перша версія мови (HTML 1.0) була спрямована на представлення мови як такої, де опис її можливостей носив скоріше рекомендаційний характер, то друга версія мови (HTML 2.0) фіксувала практику використання конструкцій мови. Версія HTML++ надавала нові можливості, розширюючи набір тегів HTML у бік відображення наукової інформації і таблиць, а також поліпшення стилю компонування зображень і тексту, а версія 3.2 була покликана упорядкувати всі нововведення і погодити

їх з існуючою практикою. Крім цього, у версії 3.2 була зроблена спроба формалізації інтерфейсу користувача гіпертекстової розподіленої системи.

У порівнянні з версією 2.0, HTML 3.2 дозволяє реалізувати відображення таблиць (контейнер <TABLE>...</TABLE>), виконання мобільних кодів (<APPLET...> ... </APPLET>), обтікання графіки текстом, а також відображення верхніх і нижніх індексів (^{...}; _{...}).

Крім цих можливостей, що фіксують поточну практику використання HTML, сучасні програми перегляду HTML-документів дозволяють реалізувати і ряд інших можливостей розмітки тексту, що описані в стандарті HTML 3.0 і розширеннях HTML фірм-розробникив програмного забезпечення:

- розмітка математичних формул (HTML 3.0);
- додаткові контейнери заголовку (HTML 3.0; Netscape Extensions; Microsoft Extensions);
- додаткові атрибути стандартних контейнерів тіла документу (ALIGN; BGCOLOR; TARGET і т. ін.);
- розбивка сторінки на кадри;
- відкриття додаткових вікон і ін.

Крім можливостей розмітки тексту, включення мультимедіа і формування гіпертекстових зв'язків, що вже існували в попередніх версіях HTML, у версію 4.0 включені додаткові засоби роботи з мультимедіа, мови програмування, таблиці стилів, спрощені засоби друку зображень і документів, що стають більш доступними для всіх користувачів HTML 4.0 і вище. Ці доповнення служать інтернаціоналізації WWW і поширенню її по усьому світі. Крім цього, для керування сценаріями перегляду Web-сторінок Website (гіпертекстової бази даних, виконаної в технології World Wide Web) можна використовувати мови програмування цих сценаріїв типу JavaScript, Java і VBScript.

➢ Перелік тегів та відповідних атрибутів мови HTML розглядається в додатку В.

5.1.2. DHTML

DHTML (Dynamic HTML) — це набір нововведень у Microsoft Internet Explorer 4.0 і вище, що дозволяють автору динамічно змінювати стилі, зміст і оформлення сторінки, створювати інтерактивні документи, що взаємодіють з користувачем у реальному масштабі часу.

Динамічні стилі — одна з основ DHTML. Шляхом нескладних функцій можна змінювати зовнішній вигляд будь-якого елементу в будь-який момент часу, наприклад, колір посилання зміниться, коли користувач наведе на нього покажчик миші. Кожен об'єкт HTML має набір властивостей, що визначають його зовнішній вигляд. Усі вони об'єднані в підмножині Style. Доступ до цих властивостей можна отримати двома шляхами: через мову HTML — загальноприйнятий набір описів для HTML-документу, чи через кілька інших Web-мов — JavaScript, JScript чи VBScript. За допомогою динамічних стилів можна змінювати будь-яку візуальну властивість об'єкту: колір, розмір, видимість і багато іншого. Виконувати ці операції легко за допомогою мов сценаріїв.

5.1.3. XML

XML — це скорочення від eXtensible Markup Language (розширювана мова розмітки). Мова XML розроблялась як засіб збереження потужності і гнучкості SGML, але без його чисельних складностей. Будучи обмеженою формою SGML, мова XML, зберігає більшість найбільш часто використовуваних можливостей SGML. Зберігаючи ці позитивні якості, у мові XML відсутні найбільш складні засоби SGML, що роблять розробку авторських документів і програмного забезпечення досить трудомістким процесом.

➢ Більш докладно мова XML розглядається нижче в пункті 5.3.

5.1.4. VRML

Мова VRML (Virtual Realty Modelling Languagy) призначена для опису тривимірних зображень і оперує об'єктами, що описують геометричні фігури і їхнє розташування в просторі.

VRML-файл являє собою звичайний текстовий файл, що інтерпретується браузером. Оскільки більшість браузерів не має вбудованих засобів підтримки VRML, для перегляду VRML-документів необхідно підключити допоміжну програму — VRML-браузер.

Як і у випадку з HTML, той самий VRML-документ може виглядати по-різному в різних VRML-браузерах. Крім того, багато розробникив VRML-браузерів додають нестандартні розширення VRML у свій браузер.

Існує чимало VRML-редакторів, що роблять процес створення VRML-документів більш зручним і швидшим, однак нескладні моделі можна створити за допомогою найпростішого текстового редактора.

5.1.5. XHTML

XHTML — це родина сучасних типів документів і модулів, що відтворюють і розширюють HTML 4.0. Родина XHTML-типів документів грунтується на XML і розраховано на роботу з агентами користувачів на основі XML.

Мова XHTML є першим типом документу в сімействі XHTML. Вона є трансформацією трьох типів документів HTML як програм XML і призначена використовуватись в якості мови для змісту, що задовольняє XML і, при дотриманні деяких простих правил, працює з агентами користувача, що задовольняють мові HTML. Розробники, що переводять свій зміст на XHTML, одержують наступні переваги:

- XHTML-документи задовольняють XML і їх можна переглядати, редагувати і перевіряти стандартними програмними засобами XML;
- можна написати такі XHTML-документи, що діють так само добре (чи навіть краще) в наявних агентах користувача мови HTML, а також у агентах користувача мови XHTML;
- XHTML-документи можуть використовувати програми (наприклад, сценарії та аплети), що спираються на об'єктну модель документу (DOM — Document Object Model) HTML чи об'єктну модель документу XML;

- зрозвитком родини XHTML, документи, що задовольняють XHTML, є інтероперабельними серед різних версій середовища XHTML.

Родина XHTML є наступним етапом в еволюції Internet. Переходячи на XHTML, розробники змісту можуть перейти у світ мови XML з усіма його перевагами, забезпечуючи при цьому пряму та зворотну сумісність.

5.1.6. AWK

AWK — утиліта призначена для простих, автоматичних і обчислювальних маніпуляцій над даними. Досить нескладні операції часто необхідно виконати над цілими пакетами файлів, а писати для цього програму на одній із стандартних мов програмування є складною справою. Оптимальне рішення проблеми — використання спеціальної утиліти AWK, що включає в себе просту і зручну мову програмування, що дозволяє вирішувати задачі обробки даних за допомогою коротких програм, що складаються з двох-трьох рядків.

Утиліта AWK споконвічно поєднувала властивості утиліт UNIX — `sed` і `grep`. Надалі її можливості значно розширились. Утиліта AWK була створена в 1977 році американськими авторами Альфредом Ахо, Брайаном Керніганом та Питером Уейнбергером.

Принцип роботи AWK полягає в тому, що ця утиліта сканує стандартний набір вхідних файлів, і виконує вказані дії над рядками, що задовольняють заданому зразку. Рядок може містити максимум до 256 символів.

5.1.7. JavaScript

Сучасні гіпертекстові інформаційні системи умовно можна представити у вигляді сукупності декількох компонентів: систем збереження гіпертекстових об'єктів, систем відображення гіпертекстових об'єктів, систем підготовки гіпертекстових об'єктів і систем програмування перегляду сукупності гіпертекстових об'єктів. З цього погляду, технологія WWW вже до 1996 року одержала закінчений, функціонально повний вид. Першими були розроблені системи збереження і перегляду (1989–1991), що продовжують розвиватись і в даний час. Після 1990 року стали з'являтись перші системи підготовки документів. Нарешті, у 1995 році були запропоновані перші мови керування сценаріями перегляду.

Програмування процедури перегляду гіпертекстової бази даних не є винаходом Netscape, Microsoft чи Sun. Практично всі локальні гіпертекстові системи в тому чи іншому ступені мають програмні засоби маніпулювання гіпертекстовими об'єктами.

У ряді випадків уся гіпертекстова база даних може бути представлена як одна велика програма, у якій гіпертекстові вузли — це програмні модулі, а зв'язки між ними — це передача керування від одного модуля іншому.

Переваги такого підходу перед традиційною статичною розміткою очевидна: гнучкість побудови гіпертекстової мережі, можливість створення програм прокручування фрагментів бази даних, генерація складених гіпертекстових об'єктів з існуючих елементарних компонентів. Динамічні об'єкти можуть бути легко отримані зі статичних, тому що у випадку існування браузера система може бути переведена з інтер-

активного режиму перегляду гіпертекстової бази даних у пакетний, коли дії оператора будуть замінятись командами програми.

Браузери гіпертекстових сторінок традиційно називають сценаріями (*scripts*), за аналогією з файлами, що виконуються, написаними для командних інтерпретаторів типу `sh`. Власне, як це було і раніш у локальних системах, у програмуванні перегляду гіпертекстових документів WWW існують два підходи: створення сценарієв, що інтерпретуються браузером, чи компіляція байт-коду.

Перший підхід не виходить за рамки традиції WWW, відповідно до якої для розробки гіпертекстової сторінки потрібний лише звичайний текстовий редактор, і сам гіпертекстовий документ повинен легко читатись користувачем. Другий підхід дозволяє підвищити ефективність виконання програми і захищеність коду від несанкціонованих модифікацій. Як перший, так і другий способи базуються на об'єктно-орієнтованому підході до програмування.

Ідея JavaScript дуже проста. Всі операції, які можна виконувати в програмі на JavaScript, описують дії над добре відомими і зрозумілими об'єктами, якими є елементи робочої області браузера і контейнери мови HTML. Власне об'єктна орієнтованість JavaScript на цьому і закінчується. Ніяких класів об'єктів, а тим більше, успадкування в JavaScript немає. Є тільки об'єкти з набором властивостей і набір функцій над об'єктами, що називаються методами.

Крім методів існують і інші функції, більше схожі на функції з традиційних мов програмування, що дозволяють працювати із стандартними математичними типами чи керувати процесом виконання програми. Ще в JavaScript є події — аналог програмних припинень. Ці події також орієнтовані на роботу в WWW, наприклад завантаження сторінки в робочу область браузера чи вибір гіпертекстового посилання. Використовуючи події, автор гіпертекстової сторінки і програми, що її відображає, може організувати перегляд динамічних об'єктів.

Для вбудовування сценаріїв у тіло HTML-документу використовується контейнер `SCRIPT`. Не всі браузери здатні розпізнавати і виконувати сценарії, тому саме тіло сценарію міститься в контейнері коментарю.

5.1.8. Java

Труднощі у створенні ПЗ пов'язані з розмаїтістю архітектур машин, операційних систем, графічних оболонок і т. ін. Крім того, програми повинні працювати в розподілених системах. Стрімке зростання технологій, пов'язаних з Internet, WWW і "електронною комерцією", додатково ускладнюють цю задачу. Популярний нині об'єктно-орієнтований підхід сам по собі не вирішує цих проблем, і більш того, часто додає нові.

Запропонований фірмою Sun Microsystems підхід, а саме система програмування на основі мови Java, має наступні характеристики:

- мова програмування об'єктно-орієнтована, у той же час досить проста для освоєння;
- цикл розробки програм скорочений за рахунок того, що система побудована на основі інтерпретатора;
- отриманий код може бути перенесений між безліччю платформ комп'ютерів і операційних систем;

- за рахунок вбудованої системи контролю пам'яті програміст звільняється від необхідності явного керування нею;
- в інтерактивній графічній програмі вдається досягти високої продуктивності за рахунок вбудованої в систему багатопоточності;
- програми легко супроводжуються і модифікуються, тому що модулі можуть бути завантажені з мережі;
- в програми вбудована система безпеки, що не допускає незаконного доступу і проникнення вірусів.

➢ Більш докладно мова Java розглядається нижче в пунктах 5.4 та 5.5.

5.1.9. PHP

Мова PHP була започаткована наприкінці 1994 року Расмусом Ледорфом. Ранні версії мови використовувались на його домашній сторінці для того, щоб стежити за тим, хто переглядав його інтерактивне резюме. Перша комерційна версія стала доступна десь на початку 1995 року і була відома як Personal Home Page Tools. Вона складалась з дуже спрощеного синтаксичного аналізатора, що "розумів" лише кілька спеціальних макрокоманд і ряд утиліт, що тоді були в загальному використанні на домашніх сторінках: гостьові книги, лічильники і деякі інші доповнення.

Мова PHP може використовуватись для створення вихідних HTML-файлів, а також для створення файлів в форматі GIF, чи навіть більш зручних потоків зображень. Крім того, PHP може сприймати файли, завантажені з будь-якого браузера, що відповідає стандартам RFC-1867. PHP дозволяє маніпулювати файлами, мати повний контроль над тим, кому дозволяти завантажувати файли і які операції повинні бути здійснені з файлом, якщо він був завантажений. Екран завантаження файлу може бути організований створенням спеціальної форми.

PHP підтримує HTTP cookies. Cookies — механізм для збереження даних у віддаленому браузері і, таким чином, — ідентифікації користувачів. Використовувати файли cookie дозволяє функція `SetCookie()`. Cookies — це частина HTTP-заголовку, так що функція `SetCookie()` повинна бути викликана до того, як браузеру послана яка-небудь інформація для висновку. Будь-який cookie, відправдений від клієнта, буде автоматично перетворений у змінну PHP так само, як і дані методів GET і POST.

PHP підтримує ряд різних баз даних (як в режимі роботи у власній системі команд, так і через ODBC), включаючи: Adabas D, MySQL, dBase, Oracle, Empress, PostgreSQL, FilePro, Solid, Informix, Sybase, InterBase, Velocis, mSQL, Unix, dbm тощо.

У PHP є 4 типи помилок і попереджень. Це:

- 1 — нормальні помилки функції (Normal Function Errors);
- 2 — нормальні попередження (Normal Warnings);
- 4 — помилки синтаксичного аналізатора (Parser Errors);
- 8 — повідомлення (Notices) — попередження, які можна проігнорувати, але вони можуть вказувати на наявність помилки.

Більш докладно мова PHP розглядається нижче в пункті 5.2.

5.1.10. Perl

Perl призначений для виконання задач командних сценаріїв Unix у тих випадках, коли вони занадто важкі чи складні для програмування на іншій мові, наприклад на C.

Perl — інтерпретована мова, пристосована для обробки довільних текстових файлів, пошуку в них необхідної інформації і видачі повідомлень. Мова Perl також зручна для написання різних системних програм. Ця мова проста у використанні, ефективна, але про неї важко сказати, що вона елегантна і компактна. Perl поєднує у собі кращі риси C, shell і awk, тому для тих, хто знайомий з цими засобами програмування, вивчення Perl не складе особливих труднощів.

Синтаксис виразів Perl близький до синтаксису C. На відміну від більшості утиліт ОС UNIX, Perl не накладає обмежень на обсяг оброблюваних даних і, якщо вистачає ресурсів, то весь файл обробляється як один рядок. Рекурсія може бути довільної глибини. Хоча Perl пристосований для сканування текстових файлів, він може обробляти так само і двійкові дані. Perl дозволяє використовувати регулярні вирази, створювати об'єкти, вставляти в програму на C чи C++ частини коду на Perl, а також дозволяє здійснювати доступ до баз даних, у тому числі Oracle.

Perl підтримує три типи даних:

- скаляри;
- масиви скалярів;
- асоціативні масиви скалярів.

Звичайні масиви, як і в мові C, індексуються числами, починаючи з нуля. Асоціативні масиви індексуються рядками. Прості скаляри завжди починаються зі знаку долару "$", навіть у тому випадку, коли ми звертаємось до елемента масиву.

Кожен тип даних має свій іменний простір, тому можна використовувати те саме ім'я одночасно для скалярної змінної, чи масиву без сумніву, що відбудеться помилка. Perl розрізняє великі і маленькі букви. Імена, що починаються з букви або знаку підкреслення, можуть надалі містити в собі цифри чи знаки підкреслення. Імена, що починаються з цифри, можуть надалі містити лише цифри. Імена, що починаються не з букви чи цифри або підкреслення, повинні складатись тільки з одного символу. Більшість таких імен зарезервовано, наприклад $$ є ідентифікатором поточного процесу.

Інтерпретація команди чи величини часто залежить від вимог контексту. Існує два основних контексти: скалярний і списковий. Деякі операції повертають список величин, якщо в контексті мається на увазі список, і одну величину, якщо контекст скалярний.

➢ Приклад використання мови Perl розглядається нижче в пункті 5.8.

5.1.11. Curl

Мова розроблена в Массачусетському технологічному інституті (MIT), і як повідомляють її розробники, поєднує в собі простоту мов Web-розмітки, таких як HTML, і потужність мов для розробки Web--програм, таких як Java.

В даний час створення великого Web-сайту припускає використання декількох різних мов та інструментаріїв — для гіпертекстової розмітки (HTML), для Web-форм (CGI, Perl), для динамічної та інтерактивної графіки (Java Script, Flash) і т. ін. Відповідно і "виконання" такої Web-сторінки може зажадати одночасного запуску декількох різних трансляторів і навіть завантаження додаткових даних з інших серверів, що викликає значні затримки в завантаженні сайту.

Мова Curl покликана об'єднати всю функціональність інших Web-мов. Творець сайту може написати на цій мові і звичайну розмітку тексту, і досить складний сценарій для інтерактивної сторінки з тривимірною графікою. При цьому розробники стверджують, що завантаження програм на Curl може відбуватись в 10 разів швидше, ніж завантаження аналогічних програм на Java, оскільки Web-інтерфейс користувача реалізується на клієнтському комп'ютері, а також розміри програм на Curl будуть значно менші.

5.1.12. Python

В даний час значну роль відіграють інтерпретовані мови, оскільки велика продуктивність персональних комп'ютерів забезпечує достатню швидкість виконання програм, що інтерпретуються, а єдиною істотною перевагою мов програмування, що компілюються, є високошвидкісний код, що ними створений. Коли швидкість виконання програми не є критичною величиною, найбільш правильним вибором буде інтерпретована мова, як більш простий і гнучкий інструмент програмування.

У зв'язку з цим, підвищений інтерес являє розгляд мови програмування Python, що була створена Гуідо ван Россумом на початку 90-х років.

Python є інтерпретованою, об'єктно-орієнтованою мовою програмування. Вона надзвичайно проста і містить невелике число ключових слів, разом з тим дуже гнучка і виразна. Ця мова більш високого рівня ніж Pascal, C++ і, відповідно, C, що досягається, в основному, за рахунок вбудованих багаторівневих структур даних (списки, словники, кортежі).

Безсумнівною перевагою є те, що інтерпретатор Python реалізований практично на всіх платформах і операційних системах. Першою такою мовою був C, однак його типи даних на різних машинах могли займати різну кількість пам'яті і це ставало деякою перешкодою при написанні програм. В мові Python такий суттєвий недолік відсутній.

Наступна важлива риса — здатність мови розширюватись. Цьому приділяється велике значення і, як пише сам автор, мова була задумана саме як розширювана. Це означає, що існує можливість вдосконалення мови самими програмістами.

Наступною перевагою є наявність великої кількості модулів, що підключаються до програми та забезпечують різні додаткові можливості. Такі модулі можуть бути написані як на C, так і на Python будь-яким досить кваліфікованим програмістом. Як приклад, можна навести наступні модулі:

- Numerical Python — розширені математичні можливості, такі як маніпуляції з векторами і матрицями;
- Tkinter — побудова програм з використанням графічного інтерфейсу користувача (GUI) на основі широко розповсюдженого на X-Windows Tk-інтерфейсу;

- OpenGL — використання великої бібліотеки графічного моделювання двох- і тривимірних об'єктів Open Graphics Library корпорації Silicon Graphics. Даний стандарт підтримується в операційних системах родини Microsoft Windows.

Недоліком мови Python є порівняно невисока швидкість виконання програми, що обумовлено тим, що мова є інтерпретованою. Однак, як вже зазначалось, на сучасноме етапі розвитку комп'ютерних технологій щвидкість виконання програми не є критичною величиною.

Особливості мови:

- Python, на відміну від багатьох мов (Pascal, C++, Java і т. ін.), не вимагає опису змінних. Вони створюються в місці їхньої ініціалізації, тобто при першому присвоєнні змінній якого-небудь значення. Отже, тип змінної визначається типом значення, що присвоюється. У цьому відношенні Python нагадує Basic. Тип змінної не є незмінним. Будь-яке переприсвоювання для неї коректно і це призводить лише до того, що типом змінної стає тип нового значення, що присвоюється.
- У таких мовах як Pascal, C і C++ організація списків додавала деяких труднощів. Для їхньої реалізації доводилось добре вивчати принципи роботи з покажчиками і динамічною пам'яттю. Навіть маючи високу кваліфікацію, програміст, щоразу заново реалізуючи механізми створення, роботи і знищення списків, міг легко допустити помилки. Через це були створені деякі засоби для роботи зі списками. Наприклад, в Object Pascal (Delphi) — клас `TList`, що реалізує списки; для C++ розроблена бібліотека STL (Standard Template Library), що містить такі структури як вектори, списки, стеки, черги та інше. Однак, такі засоби є не у всіх мовах і їхніх реалізаціях. Одною з відмітних рис Python є наявність таких вбудованих у саму мову структур як кортежі (*tuple*), списки (*list*) і словники (*dictionary*).
- Python на відміну від Pascal, C і C++, не підтримує роботу з покажчиками, динамічною пам'яттю та адресною арифметикою. У цьому мова схожа на Java. Як відомо, покажчики служать джерелом помилок, які трудно знайти, і робота з ними відноситься більше до програмування на низькому рівні. Для забезпечення більшої надійності і простоти покажчики не включені в Python.
- Одною з особливостей Python є те, як відбувається присвоєння однієї змінної іншій, тобто коли з обох сторон від оператору "=" містяться змінні.

5.1.13. Tcl

Tcl — це текстова мова з простим синтаксисом, у першу чергу призначена для подачі команд інтерактивним програмам, такимь як текстові редактори. Її легко вивчати, а досягши визначеного рівня ознайомлення з мовою, можна дуже швидко створювати високоякісні програми. На цій мові також можна програмувати процедури, тим самим, доповнюючи безліч вбудованих команд мови.

Бібліотечний пакет Tcl можна вбудовувати в прикладні програми. Бібліотека Tcl складається з аналізатора мови Tcl, підпрограм, що реалізують вбудовані команди, і процедур, що дозволяють програмам розширювати Tcl додатковими командами для роботи цієї програми. Така програма генерує команди Tcl і передає їх аналізатору Tcl для виконання.

Можна генерувати команди за допомогою читання даних із вхідного джерела за допомогою прив'язки рядків команд до елементів інтерфейсу користувача, наприклад, кнопкам, пунктам меню, комбінаціям клавіш. Бібліотека Tcl розкладає отримані команди на складові поля і безпосередньо виконує вбудовані команди. Для виконання команд, реалізованих програмою, Tcl робить виклик програми.

У багатьох випадках команди рекурсивно запускають копії інтерпретатора Tcl. Усі процедури, команди циклів і умов працюють у такий спосіб. Мова можна використовувати для з'єднання воєдино блоків, виконаних на мовах системного програмування. У програмі ці блоки здобувають вид команд мови сценаріїв. Tcl можна легко вмонтувати в існуючу програму, за рахунок чого стане можливим керувати поведінкою цієї програми і вбудовувати в неї інші блоки, наприклад, графічний інтерфейс.

Прикладна програма одержує три переваги, використовуючи Tcl як командну мову:

- По-перше, Tcl надає стандартний синтаксис і користувачі, що знають Tcl, зможуть легко давати команди будь-якій, заснованій на Tcl, програмі.
- По-друге, на Tcl можна програмувати саму програму: все, що потрібно від програми — це надати кілька своїх специфічних команд низького рівня. Tcl надає багато команд-утиліт і крім цього, загальний інтерфейс програмування для створення складних командних процедур. Використовуючи все це, прикладні програми будуть "врятовані" від необхідності самостійно відтворювати таку ж функціональність заново.
- По-третє, Tcl можна використовувати як загальну мову для спілкування програм між собою. Комунікації між програмами не вбудовані в існуюче ядро мови, але різноманітні програмні бібліотеки, такі, як інструментальний набір Tk, дозволяють програмам передавати команди одна іншій. Все це дозволяє програмам працювати спільно на більш високому якісному рівні, ніж це було можливе раніше.

5.1.14. C/C++

Мови C/C++ являють собою дуже потужний засіб написання програм, в тому числі CGI-програм для Internet. Ці мови є дуже відомими й поширеними, щоб про них розповідати, можна лише зробити деякі зауваження, що стосуються написання програм для Internet.

По-перше, сам процес написання програм для Internet на C/C++ майже нічим не відрізняється від стандартних прийомів програмування. Різниця полягає в тому, що для створення CGI-програм програмісту потрібно працювати із стандартними потоками вводу-виводу.

По-друге, ці мови використовуються (в більшості випадків) для створення більш серйозних та складних програм, ніж простий пошук або обробка текстів чи форм. Слід зауважити, що для вирішення нескладних задач, таких як обробка форм або текстових файлів, C/C++ на практиці не використовуються.

5.2. Мова PHP

PHP — це мова серверних сценаріїв, що вбудовується в HTML та інтерпретується і виконується на сервері. Приклад:

```
<html>
<head>
<title> Example </title>
</head>
<body>
<? php echo "Hi, I'm a PHP script!"; ?>
</body>
</html>
```

PHP є препроцесором HTML, і його робота побудована за схемою, наведеною на рис. 5.1.

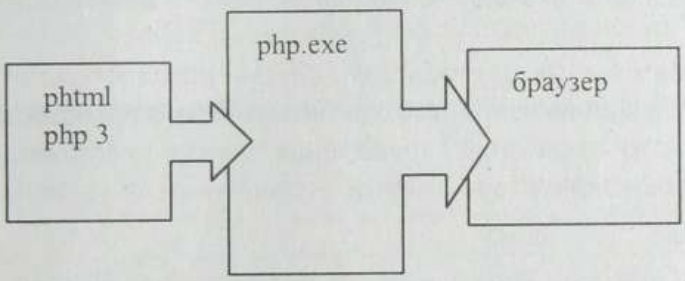


Рис. 5.1. Схема роботи PHP

До того, як сервер "віддасть" файл браузеру, його переглядає препроцесор. Для того, щоб це відбувалось, файли, що піддаються обробці препроцесором, повинні мати визначене розширення (це `.phtml` або `.php3`, або `.php4`, але ці значення можна поміняти) і містити (це не обов'язкова вимога) код для препроцесору. Перед відправленням сторінки PHP-код аналізується на сервері і браузеру видається результат у виді знову ж таки HTML-сторінки, що може сильно відрізнятись від тієї, що зберігається на сервері. Звичайні ж сторінки, що мають розширення `.html` / `.htm`, WeЬ-сервер буде відправляти браузеру без будь-якої обробки.

Основна відмінність PHP від CGI-скриптів, написаних на інших мовах, типу Perl або C полягає в тому, що в CGI потрібно самому писати вихідний HTML-код, а використовуючи PHP — лише вбудовувати свою програму в готову HTML-сторінку, використовуючи відкриваючий і закриваючий теги `<?` і `?>`.

PHP називається мовою серверних сценаріїв на відміну від JavaScript / JScript / VBScript, що є мовами клієнтських сценаріїв. Це означає, що PHP-сценарій виконується на сервері, а клієнту передається результат його роботи, тоді як у JavaScript код цілком передається на клієнтську машину і тільки там виконується браузером.

Деякими аналогами PHP є мови ColdFusion і embPerl. Усі ці мови дозволяють розміщати код, виконуваний на Web, всередині HTML-сторінок.

У PHP включена підтримка багатьох баз даних, що робить написання Web-програм з використанням БД простим.

На додаток до всього, PHP розуміє протоколи IMAP, SNMP, NNTP, POP3 і навіть HTTP, а також має можливість працювати із сокетами і спілкуватись з іншими протоколами.

Переваги PHP:

- PHP-сценарій добре масштабується;
- PHP працює як частина Web-серверу;
- синтаксис PHP дуже схожий на синтаксис C;
- у цій мові немає строгої типізації даних;
- немає необхідності в діях по виділенню/звільненню пам'яті;
- на додаток до своєї безкоштовності (MySQL вимагає придбання ліцензії при використанні її в комерційних цілях) зв'язка PHP–MySQL є крос-платформною. Це означає, що можна, працюючи в Windows, розробляти програми, призначені для роботи під Unix;
- PHP може працювати як зовнішній CGI-процес, або як звичайний інтерпретатор сценаріїв, або як модуль, що підключається до Web-серверів Apache або IIS;
- існує велика кількість документації і списків розсилок, до яких можна звернутись у випадку виникнення якихось питань. Знайдені помилки виправляються досить швидко, пропозиції і зауваження завжди вислухаються, розглядаються, і якщо вони виявляться цінними — реалізуються у нової версії PHP.

Недоліки PHP:

- Основним недоліком PHP 3, є те, що по своїй ідеології PHP завжди був орієнтований на написання невеликих сценаріїв. Незважаючи на те, що ядро кілька разів переписувалось, PHP 3 не придатна для використання у складних проектах — при обробці великих сценаріїв продуктивність системи різко падає. Однак цей недолік ліквідований у ядрі PHP 4, що, за словами розробника, призначене для роботи у великих проектах.
- PHP є мовою, що інтерпретується, і, внаслідок цього, не може зрівнятись по швидкості з мовою C, код якої компілюється. Однак при написанні невеликих програм, що притаманно PHP-проектам, коли весь проект складається з багатьох невеликих сторінок з кодом, набирають сили накладні витрати на завантаження в пам'ять і виклик CGI-програми, написаної на C.
- Невелика база готових модулів у PHP 3. У PHP 4 розробники передбачили спеціальний архів, аналогічний CPAN.

5.2.1. Синтаксис мови PHP

Код `<?` означає початок блоку команд, який треба обробити і виконати. Закінчується блок обмежником `?>`. Іншими словами символи `<?` і `?>` виконують роль дужок. Усе, що знаходиться поза ними, PHP-ядро пропускає і відправляє в Web без будь-якої обробки, виконуючи лише те, що знаходиться всередині цих "дужок". Потужність PHP полягає в тому, що PHP-код можна розміщувати в будь-якому місці HTML-сторінки.

Символ `$` (долар) повідомляє PHP, що перед ним змінна. Змінні, крім рядків, можуть містити числа і масиви. У будь-якому випадку, люба змінна завжди позначається символом `$`.

Основні функції PHP

Функція **echo()** — виводить рядок, що міститься у лапках або змінну, яка йде слідом.

Функція **print()** — відправляє текст у браузер. Функція echo працює так само, як і print, однак дозволяє додавати до першого текстового рядка інші рядки, розділяючи їх комами.

Функція **printf()** — відображає числа у визначеному форматі, наприклад, виводить дробове число з визначеною кількістю нулів після коми, тому у функції printf використання дужок обов'язково.

Обробка даних HTML-форм

PHP легко обробляє дані, отримані від HTML-форм. Кожне поле вводу має атрибут NAME, що буде переданий в оброблювач разом із своїм значенням.

Існує два методи передачі даних: GET і POST. Їхня відмінність полягає в тому, що при використанні методу GET значення полів приєднуються до URL, зазначеному в атрибуті ACTION.

У табл. 5.1 перераховані можливі елементи вводу, що використовуються у формах.

Таблиця 5.1

Тип	Опис
TEXT	Поле вводу тексту
SELECT	Вибір зі списку
RADIO	Радіо-кнопка використовується для вибору одного із запропонованих варіантів
CHECKBOX	Кнопка-прапорець — використовується для вибору варіанту
SUBMIT	Кнопка, яка ініціює виклик обробки форми
IMAGE	Зображення — використовується як кнопка типу SUBMIT
<TEXTAREA>	Область вводу тексту

Для поля типу TEXT введене значення передається у вигляді: ім'я=значення. В оброблювачі значення можна одержати із змінної $ім'я.

Для поля тепу SELECT значення береться з атрибуту VALUE обраного елемента <OPTION>. Елемент <SELECT> може мати атрибут MULTIPLE, що дозволяє вибирати кілька значень зі списку. У цьому випадку до імені елементу <SELECT> необхідно додати пари квадратних дужок: ім'я[]. Рядок буде виглядати наступним чином:

```
ім'я[]=значення&[]==значення...,
```

а в сценарію доступ до обраних значень можна здійснити як до елементів масиву $ім'я.

У випадку, якщо не задані атрибути VALUE, то передаватись буде те, що міститься між тегами <OPTION> і </OPTION>.

Для поля типу RADIO значення буде здобуватись із атрибуту VALUE, рядок виглядає аналогічно елементу типу TEXT. Доступ із сценарію, теж аналогічний. Якщо це значення не встановлене, то буде передане значення on.

Для поля типу CHECKBOX: якщо прапорець встановлений, то передається значення on, якщо прапорець не встановлений, то змінна не передається взагалі. Таким чином, установку прапорця в сценарію можна перевірити, порівнявши значення змінної $ім'я з "on". Змінна і рядок виглядають аналогічно елементу типу TEXT.

Кнопка SUBMIT теж може передавати значення в оброблювач. Значення встановлюється з атрибуту VALUE. Все інше аналогічно полю типу TEXT.

Для поля типу IMAGE в оброблювач передаються два значення: ім'я.x і ім'я.y, що являють собою координату покажчика миші щодо верхнього лівого кута зображення. Рядок виглядає наступним чином:

```
ім'я.x=значення &ім'я.y=значення.
```

У сценарію встановлюються змінні $ім'я_x і $ім'я_y.

Елемент типу <TEXTAREA> абсолютно аналогічний елементу типу TEXT.

При пересиланні рядкових значень вони перекодуються спеціальним образом. Усі символи, крім алфавітно-цифрових і знаку підкреслення "_" заміняються знаком відсотка "%" і двома шістнадцятковими цифрами коду. Пробіли замінюються на знак "+". При установці змінних у сценарію виробляється зворотне декодування.

Приклад роботи з формами:

```
<HTML>
<HEAD>
<TITLE>Запит інформації</TITLE>
</HEAD>
<BODY>
<CENTER>
Бажаєте більше знати про наші товари?
<P>
<TABLE WIDTH = 400><TR><TD align = right>
<FORM ACTION = "email.php3" METHOD = "POST">
Ваше ім'я: <BR>
<INPUT TYPE = "text" NAME = "rame"
       SIZE = "20" MAXLENGTH = "30">
<P>
Ваш e-mail: <BR>
<INPUT TYPE = "text" NAME = "email"
       SIZE = "20" MAXLENGTH = "30">
<P>
Мене цікавлять:
<SELECT NAME = "preference">
<OPTION value = "Яблука">Яблука
<OPTION value = "Апельсини">Апельсини
</SELECT>
<P>
<INPUT TYPE = "submit" VALUE = "Відправити запит!">
</FORM>
</TD></TR></TABLE></CENTER>
</BODY>
</HTML>.
```

Назвемо цей файл request.html. В ньому вказано, що данні форми будуть оброблятись, файлом email.php3. Наведемо його зміст:

```
<?
/* Цей сценарій отримує змінні з request.html */
PRINT "<CENTER>";
PRINT "Привіт, $name.";
PRINT "<BR><BR>";
PRINT "Дякую за вашу зацікавленість.<BR><BR>";
PRINT "Вас цікавлять $preference.
       Інформацію щодо них ми вишлемо вам на e-mail: $email.";
PRINT "</CENTER>";
?>
```

Функції PHP для роботи з БД MySQL

int_mysql_connect (string hostname, string username, string password) — створює з'єднання з MySQL, де hostname — ім'я хосту, на якому знаходиться база даних; username — ім'я користувача; password — пароль користувача. Функція повертає параметр типу int, що більше 0, якщо з'єднання пройшло успішно, і дорівнює 0 у протилежному випадку.

int_mysql_select_db (string database_name, int link_indentifier) — вибирає базу даних для роботи, де database_name — ім'я бази даних; link_indentifier — ідентифікатор з'єднання, що отримано у функції mysql_connect (параметр необов'язковий, якщо він не вказується, то використовується ідентифікатор від останнього виклику mysql_connect). Функція повертає значення true або false.

int mysql_query (string query, int link_indentifier) — виконує запит до бази даних, де query — рядок, що містить запит. Функція повертає ідентифікатор результату або 0, якщо відбулась помилка.

int mysql_result (int result, int i, column) — повертає значення поля в стовпці column та в рядку i.

int mysql_close (int link_indentifier) — закриває з'єднання з MySQL. Функція повертає значення true або false.

int mysql_num_rows (int result) — повертає кількість рядків у результаті запиту, де result містить ідентифікатор результату запиту.

Тепер файл email.php3 буде мати наступний вигляд:

```
<?
/* Цей сценарій отримує змінні з request.html */
/* Деякі змінні */
$hostname = "localhost";
$username = "myusername";
$password = "mypassword";
$dbName = "products"";
/* Таблиця MySQL, в якій зберігаються дані */
$userstable = "clients";
/* e-mail адміністратора */
```

```
$adminaddress = "administration@me.com";
/* створити з'єднання */
MYSQL_CONNECT ($hostname, $username, $password) OR
      DIE ("Не можу створити з'єднання");
@mysql_select_db ("$dbName") or
       die ("Не можу вибрати базу даних");
PRINT "<CENTER>";
PRINT "Привіт, $name.";
PRINT "<BR><BR>";
PRINT "Дякую за вашу зацікавленість.<BR><BR>";
PRINT "Вас цікавлять $preference.
       Інформацію щодо них ми вишлемо вам на e-mail: $email.";
PRINT "</CENTER>";

/* Відправляємо на e-mail */
mail ($email, "Запит на інформацію", "$namen \ n
      Дякую за вашу зацікавленість! \ n
      Вас цікавлять $prefeence \ n
      Мы їх розповсюджуємо безкоштовно. Зверніться в
      найближчий філіал їх компанії й отримайте ящик
      цього продукту. \ n");
mail ("administration@me.com",
      "Був запит щодо інформації.",
      "$name цікавили $preference \ n"
       e-mail-адреса: $email. \ n");
/* Вставити інформацію про клієнта в таблицю */
$query = "INSERT INTO $userstable VALUES ('$name', '$email',
'$preference')";
$result = MYSQL_QUERY ($query);
PRINT ("Інформація про вас занесена в базу даних.");
/* Закрити з'єднання */
MYSQL_CLOSE ();
?>
```

Напишемо сценарій apple.php3.

```
<? /* Сцнарій показує клієнтів, які яблука люблять більше,
     ніж апельсини */
$hostname = "localhost";
$username = "myusername";
$password = "mypassword";
$dbName = "products";
/* Таблиця My SQL, в якій зберігаються дані */
$userstable = "clients";
/* створити з'єднання */
@mysql_select_db ("$dbName") or
       die ("Не можу вибрати базу даних");
```

```
/* Вибрати всіх клієнтів-яблучників */
$query = "SELECT * FROM $userstable WHERE choice = "Яблука"";
$result = MYSQL_QUERY ($query);
/* Як багато знайшлося таких? */
$number = MYSQL_NUMROWS ($result);
/* Надрукувати всіх у форматовоному вигляді */
$i= 0;
IF ($number == 0) {
PRINT "<CENTER><P> Любителів яблук нема </CENTER>";
} ELSEIF ($number > 0) {
PRINT "<CENTER><P> Кількість любителів яблук:
      $number  <BR><BR>";
WHILE ($i < $number) {
$name = mysql_result ($result, $i, "name");
$email = mysql_result ($result, $i, "email");
PRINT "Клієнт $name любить яблука.<BR>";
PRINT "Його e-mail: $email.";
PRINT "<BR><BR>";
$i++;
}
PRINT "</CENTER>";
}
?> .
```

5.3. Мова XML

У 1998 консорціум W3C затвердив RFC для XML 1.0. XML (Extensible Markup Language) — нова утворена від SGML мова розмітки документа, що дозволяє структурувати інформацію довільного типу.

XML — мова розмітки, яка описує об'єкти даних. Вона використовується для опису граматик інших мов, та слідкування за правильністю складання документу.

Набору тегів для XML не існує. Ця мова скоріше спрямована для розмітки і дозволяє вільно розширити набір тегів, які потрібні. Завдяки використанню тегів, процес створення XML-документу дуже схожий на процес створення документу HTML. Різниця полягає у тому, що в XML можна створювати необхідні теги по мірі необхідності. Мало того, розробники поклопотались про те, щоб можна було розробляти XML-документ будь-якою мовою, яка може бути кодована в unicod.

У розробників одразу ж виникає питання, а навіщо ця мова потрібна? Де сфера її використання? Спробуємо відповісти на ці питання.

Способи використання мови XML:

- Ця мова може бути корисною для розробників складних інформаційних систем, що зв'язані потоками інформації різнорідної структури. У цьому випадку XML виконує роль універсального формату для обміну даними між окремими компонентами.

- XML — базовий стандарт для нової мови опису ресурсів, що дозволяє спростити проблеми в Web, пов'язані з пошуком необхідної інформації, створення електронних бібліотек і т. ін.
- Завдяки тому, що XML описує данні довільного типу та використовує представлення спеціальної інформації (математичної, хімічної, біологічної). Це означає, що XML може слугувати потужним інструментом для розповсюдження інформації у Web.
- XML-документи можуть використовуватись у якості проміжного формату даних. Звичайна схема взаємозв'язку між серверними задачами та базами даних залежить від конкретної СУБД та діалектів SQL. Якщо результати пропонувати в деякому універсальному текстовому форматі, тоді СУБД стане прозорою з точки зору клієнта.
- Використання таблиці стилів XSL дозволяє забезпечити незалежне від конкретного пристрою відображення (візуалізацію) XML-документа.
- XML може бути використаний у звичайних програмах для збереження та обробки структурованих даних в універсальному форматі.

Алі навіщо для цього використовувати саме XML? Справа у тім, що XML дозволяє просто й легко робити те, що за допомогою інших засобів буде дуже незручно або громіздко. Отже використання XML обумовлене:

- уніфікованою структурою для обробки та обміну даними;
- можливістю зберегти логіку даних, що розміщені в документі;
- використанням єдиної розмітки для правильної інтерпретації об'єктів у документах;
- можливістю використання існуючих документально-орінтованих інструментів та об'єднання їх із можливостями баз даних.

5.3.1. Правила побудови XML-документу

У загальному випадку XML-документи повинні задовольняти наступним вимогам:

- у заголовку документа розміщено XML-заголовок, в якому вказується мова розмітки документа, номер версії, що використовується, та додаткова інформація;
- кожен відкриваючий тег, що визначає деяку область даних у документі, обов'язково повинен мати закриваючий тег;
- XML враховує регістр букв;
- вкладення тегів суворо контролюється, тому необхідно слідкувати за порядком їх слідування;
- всі значення атрибутів мають бути взяті в лапки;
- вся інформація що розміщена між початковим та кінцевим тегами, розглядається як дані, тому враховуються усі символи форматування (пробіли, символи переведення на новий рядок, табуляція не ігноруються, як це зроблено в HTML).

В якості прикладу розглянемо невеликий XML-документ. Порівнявши його з прикладом, написаним на HTML, можна швидко побачити різницю. XML дозволяє представити структуру даних без допоміжних засобів. Теги початку та теги кінця є

основними в XML-розмітці, елементам можна присвоїти атрибут (як, наприклад, у HTML тег <table> може мати атрибут align="left|center|right"). При цьому кількість таких атрибутів необмежена. Наприклад,

```
<publication topic="networking" circulation="controlled">.
```

XML-документ може містити посилання на інші об'єкти, що повинні починатись з "&" та закінчуватись ";". Наприклад,

```
<article>
  &introduction;
  &body;
  &sidebar;
  &conclusion;
  &resources;
</article> .
```

Наступним видом розмітки є коментарі, які в XML виділяються так саме, як і в HTML. Приклад:

```
<!-i Це є коментар -i> .
```

Інструкціям передує знак питання, який повідомляє програмі аналізу документа необхідну інформацію. Наприклад, інструкція <?xml version = 1.0?> повідомляє, що документ складений за допомогою XML версії 1.0.

Можна також додавати символьні дані. Тег, що позначає початок, є <![CDATA[, а що позначає кінець —]]>. Ця частина документу розглядається виключно як символьні дані (частина HTML-документу). Наприклад:

```
<![CDATA[
<b> це буде виділено жирним шрифтом </b>
]]> .
```

Нижче наведено приклад XML-документу, що описує книжки (файл html.xml).

```
<?xml version="1.0" encoding="windows-1252" ?>
<?xml-stylesheet type="text/xsl" href="html.xsl"?>
<!-- Вказівник на XSL-файл, де знаходиться
     інструкція з форматування -->
<LIBRARY>
  <BOOK>
    <NAME>Apache</NAME>
    <PRICE>37</PRICE>
  </BOOK>
  <BOOK>
    <NAME>Perl Cookbook</NAME>
    <PRICE>49</PRICE>
  </BOOK>
  <BOOK>
    <NAME>JDBC</NAME>
    <PRICE>27</PRICE>
  </BOOK>
  <BOOK>
```

```
    <NAME>Programming SERVLETS</NAME>
    <PRICE>55</PRICE>
  </BOOK>
  <BOOK>
    <NAME>XML</NAME>
    <PRICE>35</PRICE>
  </BOOK>
</LIBRARY> .
```

5.3.2. Документ опису типів (DTD)

Якщо теги XML використовуються лише заради зручності тільки на власному Web-вузлі розробника, то від цього було б мало користі. Немає ніякого сенсу у тому, які імена привласнені тегам, тому що їх значення відомі лише розробнику. Якщо ж треба надати дані зовнішньому світу, то елементи та атрибути повинні бути зрозумілі й іншими користувачам XML-документу. Тобто теги потрібно задокументувати. Для цього можна використати документ опису типів DTD (Document Type Definition).

Для того, щоб використовувати дані з XML-документу, треба написати програму, яка б аналізувала цей документ. DTD можуть зберігатись на початку XML-документа. Це робиться наступним чином:

```
<?xml version="1.0"?>
<! DOCTYPE journal SYSTEM "mydtd.dtd">
...
```

При перевірці документу, програма-аналізатор в першу чергу шукає DTD-опис у самому XML-документі. Опція `standalone="yes"` вказує аналізатору на те, що DTD-опис не можна завантажити із зовнішнього носія. Наприклад:

```
<?xml version="1.0" standalone="yes" ?> .
```

DTD може бути також розміщено в окремому файлі. В цьому разі елемент опису типу має наступний вигляд:

```
<! DOCTYPE journal [
<!ELEMENT journal (contacts, issues, authors)>
...
]>
...
```

DTD-опис може бути взагалі відсутній. Відмітимо, що програми-аналізатори можна розділити на дві великі групи:

- веріфікуючі — такі, що використовують DTD-опис для визначення коректності документу;
- не веріфікуючі — відповідно такі, що не перевіряють коректність документу.

Наприклад, якщо описується мова та її граматика на основі DTD, то для аналізу документів, що написані цією мовою, потрібна програма-аналізатор, яка перевіряє коректність складання документа. Це буде веріфікуюча програма.

DTD-опис дозволяє програмам-аналізаторам перевіряти правильність відносин між елементами та здійснювати контроль за типами.

У будь-якому випадку, використовуючи універсальні XML-аналізатори, можна бути впевненим у тому, що при синтаксичній коректності заданих в документі конструкцій мови програма-аналізатор зможе правильно розпізнати обумовлені ними елементи документу і передати їх прикладній програмі, що виконує необхідні дії по відображенню.

Таким чином після аналізу документу в більшості випадків буде надана об'єктна модель, що відображає зміст документу, і засоби, необхідні для роботи з ним (переміщення по дереву елементів).

Розглянемо приклад коректного XML-документу:

```
<?xml version="1.0"?>
<! DOCTYPE journal [
<!ELEMENT contacts (address, tel+, email?)>
<!ELEMENT address (street, appt)>
<!ELEMENT street PCDATA>
<!ELEMENT appt (PCDATA | EMPTY)*>
<!ELEMENT tel PCDATA>
<!ELEMENT email PCDATA> ]>
....
<contacts>
<address>
<street>Marks avenue</street>
<appt id="4"> </address>
<tel>12-12-12</tel>
<tel>46-23-62</tel>
<email>info@j.com</email>
</contacts> .
```

Елемент у DTD-описі визначається за допомогою тегу `!ELEMENT`, у якому вказується назва елементу та структура його змісту. Наприклад, тег `<!ELEMENT tel PCDATA>` говорить про те, що дана конструкція — опис елементу XML. В середині задається назва елементу — `tel` та тип його змісту. Зміст `tel` визначається за допомогою спеціального маркера `PCDATA` (що означає *parseable character data* — будь яка інформація з якою може працювати програма-аналізатор). Існує ще два типи конструкцій, що визначають типи елементу: `EMPTY` та `ANY`. Тип `EMPTY` указує на те, що елемент має бути пустим. Наприклад, для частини XML-документу

```
<tel> <work/>22-33-22</tel>
```

DTD-опис буде наступним:

```
<!ELEMENT work EMPTY> .
```

Тип `ANY` указує на те, що зміст елементу спеціально не описується.

Можуть бути використані більш складні конструкції, наприклад:

```
<!ELEMENT appt (PCDATA | EMPTY)*> .
```

Символ "*" вказує на те, що елемент `appt` може зустрічатись скільки завгодно разів, а може бути відсутній.

Послідовність "дочірніх" елементів для конкретного об'єкту задаються у вигляді списку назв елементів, що розділені комами. При цьому для заданя кількості однотипних елементів використовують спеціальні символи "+", "*", "?". Наприклад:

```
<!ELEMENT contacts (address, tel+, email?)> .
```

Цей тег говорить про те, що address є обов'язковим елементом та може зустрічатись лише раз, елемент tel зустрічається один та більше разів, email — елемент, що є оптиціональним, тобто він може зустрічатись, а може бути відсутній.

Списки атрибутів елемента задаються за допомогою ключового слова !ATTLIST. У ньому можна задати назви атрибутів, типи їх значень та деяку додаткову інформація. Наприклад для елементу <article> можна використати наступний опис:

```
<!ATTLIST article
id ID #REQUIRED
about CDATA #IMPLIED
type (actual | review | teach ) 'actual' ''
> .
```

У даному прикладі для елементу article визначені три атрибути: id, about та type, що мають типи ID (ідентифікатор), CDATA та список можливих значень відповідно. Узагалі існує шість можливих типів атрибутів:

- CDATA — значенням можуть бути довільні символьні дані;
- ID — визначає унікальний ідентифікатор у документі;
- IDREF (IDREFS) — указує, що значенням атрибута повинна виступати назва (або декілька таких назв, що розділені пробілом в останньому випадку) унікального ідентифікатора, визначеного в даному елементі;
- ENTITY (ENTITIES) — значенням атрибуту повинна бути назва (або список назв якщо використовується ENTITIES) компонентів (макровизначень), що визначені в даному документі;
- NMTOKEN (NMTOKENS) — вмістом елемента може бути лише одне окреме слово (тобто цей параметр є обмеженим варіантом CDATA);
- список припустимих значень — визначається список значень, що може мати даний атрибут.

Також у визначені атрибута можна використовувати наступні параметри:

- #REQUIRED — визначає, чи є атрибут обов'язковим, який повиннн бути заданий в усіх елементах даного типу;
- #IMPLIED — атрибут не є обов'язковим;
- #FIXED "значення" — вказує, що атрибут повинен мати лише визначене значення, однак самий опис атрибуту не є обов'язковим, у процесі аналізу його значення в будь-якому випадку буде передане програмі-аналізатору;
- значення — задає значення по замовчуванню.

Визначення компонентів (макровизначень)

Компонент (*entity*) являє собою визначення, зміст якого може бути повторно використане в документі. В інших мовах програмування подібні елементи називаються

макровизначеннями. DTD-компоненти створюються за допомогою інструкції !ENTITY:

```
<!ENTITY hello ' Ми ради вітати Вас!' > .
```

Програма-аналізатор, переглядаючи в першу чергу зміст області DTD-визначень, обробить цю інструкцію і при подальшому розборі документу буде використовувати вміст DTD-компонента в тому місці, де буде зустрічатись його назва. Таким чином, тепер у документі можна використовувати вираз `&hello;` , що буде замінено на рядок "Ми ради вітати Вас".

У загальному випадку, усередині DTD можна задати три типи макровизначень:

- **Внутрішні макровизначення** — призначені для визначення рядкової константи. З їхньою допомогою можна організовувати посилання на інформацію, що часто змінюється. Це робіть документ більш зручними для читання. Внутрішні компоненти включаються в документ за допомогою символу "&" . У XML існує п'ять попередньо встановлених внутрішніх символьних констант:
  - `<` — символ "<";
  - `>` — символ ">";
  - `&` — символ "&";
  - `&apos;` — символ апострофа;
  - `&quot;` — символ подвійних лапок.
- **Зовнішні макровизначення** — вказують на вміст зовнішнього файлу, причому змістом можуть бути як текстові, так і двійкові дані. У першому випадку в місці використання макросу будуть вставлені текстові рядки, у другому — бінарні дані, що аналізатором не розглядаються й використовуються зовнішніми програмами. Приклад:

```
<!ENTITY logotype SYSTEM "/image.gif" NDATA GIF87A> .
```

- **Макровизначення правил** — макровизначення можуть використовуватись лише всередині області DTD і позначаються спеціальним символом "%", що вставляється перед назвою макросу. При цьому зміст компонента буде поміщено безпосередньо в текст DTD-правила.

  Наприклад, для наступного фрагмента документа:

```
<!ELEMENT name (PCDATA)>
<!ELEMENT title (PCDATA | name)*>
<!ELEMENT author (PCDATA | name)*>
<!ELEMENT article (title, author)*>
<!ELEMENT book (title, author)*>
<!ELEMENT bookstore (book | article)*>
<!ELEMENT bookshelf (book | article)*> ,
```

  можна використовувати більш коротку форму запису:

```
<!ELEMENT name (PCDATA)>
<! ENTITY %names 'PCDATA | name'>
<!ELEMENT article (%names;)*>
<!ELEMENT book (%names;)*>
<!ENTITY %content 'book | article'>
<!ELEMENT bookstore (%content;)*>
<!ELEMENT bookshelf (%content;)*> .
```

Макровизначення часто використовуються для опису параметрів у правилах атрибутів. У цьому випадку з'являється можливість використовувати однакові визначення атрибутів для різних елементів:

```
<!ENTITY %itemattr 'id ID #IMPLIED src CDATA'>
<!ENTITY %bookattr "ISDN ID #IMPLIED type CDATA'>
<!ENTITY %artattr " size CDATA'>
<!ELEMENT book (title, author,content)*>
<!ATTLIST book %itemattr  %bookattr;>
<!ELEMENT article (title, author, content)*>
<!ATTLIST article %itemattr  %artattr;>
```

5.3.3. XML з точки зору програмування

Створювати нову мову, подібну до HTML, немає сенсу. XML був створений скоріше для програмування. Треба зауважити, що насправді сам по собі XML не є дуже важливим для програмування. Більш важливим є інтерфейс прикладного програмування DOM (Document Object Model — об'єктна модель документу).

DOM — це специфікація інтерфейсу програмування для HTML та XML. Важливим було саме стандартизувати інтерфейс роботи з документами (доступ до елементу, маніпулювання ним). При цьому термін "документ" має широке розуміння — це одиниця інформаційного опису, яку можна закодувати на XML. Тобто XML розглядають як документ, а DOM — як спосіб маніпулювати цим описом.

У рамках DOM програміст може створювати документи, одержувати доступ до їх складових частин (елементів), додавати, видаляти елементи і змінювати їхній зміст. Головним у DOM є універсальний, незалежний від мови програмування інтерфейс прикладного програмування Web-документів.

У специфікації DOM спеціально наголошено питання про те, що застосування DOM не має на увазі модель даних типу "дерево" чи "ієрархія дерев". Мова XML, для якої розроблена модель DOM, базується на представленні даних у вигляді ієрархії дерев, але для маніпулювання структурами такого типу не обов'язково використовувати саме цю модель даних. Можна обійтись списками або іншими моделями, що однозначно можна відобразити на дерева.

Таким чином, DOM визначає:

- інтерфейси і об'єкти для представлення документу й маніпулювання їм;
- семантику інтерфейсів і об'єктів, включаючи їхню поведінку й атрибути;
- взаємини та об'єднання інтерфейсів з об'єктами.

Головне, що потрібно знати — це те, що DOM будує модель документу (як множину вузлів, що зв'язані між собою відношенням "предок – син") та інтерфейс для роботи з елементами цієї моделі. Зауважимо, що інтерфейс — це фактично опис класу об'єктів, і задача програміста полягає в тому, щоб реалізувати цей інтерфейс у конкретній мові програмування.

Об'єктна модель документу повинна слугувати з'єднальною ланкою між програмуванням і логічною структурою документа, формою його представлення на носіях. Логічна структура й форма представлення описується в рамках XML. Програмування Web — це мови Java, JavaScript, JScript тощо. Стандарт DOM дозволяє формально

описати правила маніпулювання структурою XML-документів і їхнім змістом у рамках об'єктно-орієнтованого програмування, що реалізують перераховані мови.

5.3.4. Стильові таблиці

Розглянемо можливість відображення XML-файлів у вікні браузера. Це часто буває потрібно, оскільки XML-мова є зручною для представлення будь-якої інформації. Що робити для того, щоб користувач мав змогу в зручному вигляді отримати будь-яку інформацію? Для відповіді на це питання необхідно розглянути стильові таблиці.

Стильовими таблицями прийнято називати спеціальні інструкції, що керують процесом відображення елемента у вікні програми-клієнта (наприклад, у вікні браузера). Запропоновані в якості рекомендація W3C, каскадні стильові таблиці (CSS-Cascading Style Sheets) уже багато років використовуються Web-розробниками для оформлення Web-сторінок. Підтримка CSS найбільш відомими на сьогоднішній день браузерами дозволила використовувати стильові таблиці для рішення найширшого спектру задач — від оформлення домашньої сторінки до створення великого корпоративного Web-вузла.

Слово "каскадні" у визначенні CSS означає можливість об'єднання окремих елементів форматування шляхом вкладених описів стилю. Наприклад, атрибути тексту, задані всередині тегу `<body>`, будуть поширюватись на вкладені теги доти, доки в них не зустрінуться стильові описи, що скасовують чи доповнюють поточні параметри. Таким чином, використання таблиць CSS у HTML було дуже ефективно — відпадала необхідність явного завдання тегів форматування для кожного з елементів документа.

Для XML-документів застосовують інший формат стильових таблиці — XSL. Розглянемо їх відмінності:

- дозволяють визначати оформлення елементу в залежності від його місця розташування всередині документа, тобто до двох елементів з однаковою назвою можуть застосовуватист різні правила форматування;
- мовою, що лежить в основі XSL, є XML, а це означає, що XSL більш універсальна, і в розробників з'являється можливість використовувати засоби контролю коректності складання таких стильових таблиць;
- таблиці XSL не є каскадними, подібно CSS, тому що надзвичайно складно забезпечити "каскадованість" стильових описів, чи, іншими словами, можливість об'єднання окремих елементів форматування шляхом вкладених описів стилю, у ситуації, коли структура вихідного документу заздалегідь невідома і він створюється в процесі самої розробки. Однак у XSL існує можливість задавати правила для стилів, за допомогою яких можна змінювати властивості стильового оформлення, що дозволяє використовувати досить складні прийоми форматування.

Принцип обробки XML-документів стильовими таблицями полягає в наступному: при аналізі XSL-документа програма-аналізатор обробляє інструкції цієї мови і кожному елементу, знайденому в XML-дереві ставить у відповідність набір тегів, що визначають форматування цього елементу. Іншими словами, задається шаблон форматування для XML-елементів, причому сам цей шаблон може мати структуру від-

повідного фрагменту XML-документу. Взагалі кажучи, при побудові XSL-документу можна користуватись будь-якими тегами, які "розуміє" програма перегляду Web-сторінок.

Правила XSL

XSL-документ являє собою сукупність правил побудови, кожне з який виділене в окремий блок, обмежений тегами <rule> і </rule>. Правила визначають шаблони, по яких кожному елементу XML ставиться у відповідність послідовність HTML-тегів, тобто усередині них містяться інструкції, що визначають елементи XML- документу і теги форматування, застосовані до них.

Елементи XML, до яких буде застосовувано форматування, позначаються в XSL тегом <target-element/>. Для вказівки елементу з конкретною назвою, тобто визначення класу елемента, можна використовувати атрибут type="<ім'я_елемента>" .

Наведемо приклад найпростішого XSL-документу, що визначає форматування для фрагменту <flower>rose</flower>:

```
<xsl>
<rule>
<target-element type="flower"/>
<p color="red" font-size="12">
<children/>
</p>
</rule>
</xsl> .
```

Інструкція <target-element> вказує на те, що дане правило визначає елемент. Параметром type="flower" задається назва XML-елементу, для якого буде використовуватись це правило. Програма-конвертор буде використовувати HTML-теги, розміщені всередині блоку <rule></rule> для форматування XML-елементу, якому "призначався" поточний блок. У тому випадку, якщо для якогось елементу XML шаблон не визначається, у вихідний документ будуть додані теги форматування за замовчуванням (наприклад, <DIV></DIV>).

Процес розбору XSL-правил є рекурсивним, тобто якщо в елемента є дочірні елементи, то програма буде шукати визначення цих елементів, розташованих "глибше" у дереві документу. Вказівкою на те, що необхідно повторити процес розбору XML-документу, тепер уже для дочірніх елементів, є інструкція <children/>. Дійшовши до неї, аналізатор вибере з ієрархічного дерева XML-елементів потрібну гілку і знайде в XSL-шаблонах правила, що визначають форматування цих елементів, які лежать нижче. У тому випадку, якщо замість <children> вказується інструкція <empty/>, програма закінчить рух по цим гілкам і повернеться назад у "батьківське" правило. При цьому поточне правило ніякої інформації у вихідному HTML-документі змінювати не буде, тому що <empty/> у даному випадку означає, що вміст елемента відсутній.

Розглянемо приклад XSL-документа для опису книжок (файл html.xsl).

```
<?xml version="1.0"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl">
```

```
<xsl:template match="/">
  <html>
  <body>
    <table border="1" cellpadding="4" cellspacing="0">
      <tr bgcolor="#999999" align="center">
        <th>Name</th>
        <th>Price</th>
      </tr>
      <xsl:for-each select="LIBRARY/BOOK">
      <tr>
        <td><xsl:value-of select="NAME"/></td>
        <td><xsl:value-of select="PRICE"/></td>
      </tr>
      </xsl:for-each>
    </table>
  </body>
  </html>
</xsl:template>
</xsl:stylesheet> .
```

Тепер можна відкоригувати XML-документ, який описує книжки, що був наведений раніше:

```
<?xml version="1.0" encoding="windows-1252" ?>
<?xml-stylesheet type="text/xsl" href="html.xsl"?>
<!-i покажчик на XSL-файл, що містить
     інструкцію з форматування -i>
<LIBRARY>
...
```

Тепер можна запустити в браузері XML-документ, в результаті до нього застосується XSL-опис.

5.3.5. XML в порівнянні з базами даних

Чому для зберігання даних використовується саме формат XML, а не база даних? Справа в тому, що для багатьох випадків база даних занадто громіздка. Щоб використовувати БД необхідно встановити і підтримувати окремий процес на сервері, що часто вимагає також і встановки і підтримки адміністратора бази даних. Необхідно вивчити SQL і писати SQL-запити, що конвертують дані з реляційної БД в об'єктну структуру, і навпаки. Якщо ж зберігати дані в XML-файлі, то відпадає необхідність використовувати подібну надбудову. Крім того дані можна редагувати, використовуючи будь-який текстовий редактор замість складного інтерфейсу бази даних. XML-файли також легко відновлювати, поширювати їх серед друзів або завантажувати клієнтам. Також можна легко поновлювати дані на сайті, використовуючи FTP.

Більш абстрактною перевагою XML є те, що цей формат може бути використаний для конструювання структури даних значно більш наочним чином. Непотрібно буде застосовувати редактор відносин елементів або нормувати схему, що створю-

ється. Якщо є один елемент, що містить інший, то можна вказати це безпосередньо у форматі, а не використовувати спільну таблицю.

Відзначимо, однак, що для багатьох програм використання лише файлової системи буде недостатньо. В разі великої інтенсивності відновлень даних, файлова система може "створювати" помилки і руйнуватись при безлічі одночасних спроб запису. Бази даних, звичайно, для запису даних підтримують транзакції. Крім того, бази даних — це сучасний засіб виконання складних запитів, особливо, якщо ці запити можуть час від часу мінятись. Бази даних будують індекси і змінюють їх відповідно до того набору даних, що коригується. Реляційні бази даних мають також безліч інших переваг, включаючи "багату" мову запитів, великі засоби конструювання схем, перевірену маштабованість, прорблений контроль доступу тощо.

Однак, з точки зору малих і середніх Web-сайтів, призначених для публікацій, більшість операцій доступу до даних є читанням, а не записом; дані, хоча і великі, змінюються рідко; немає необхідності в складному пошуку, а якщо така необхідність і з'являється, то пошук виконується окремою пошуковою програмою. У такому випадку переваги використання реляційної бази даних невеликі, а переваги використання об'єктно-орієнтованої моделі даних — значні.

Нарешті, можливо використовувати перетворювач результатів SQL-запитів у XML-потоки, так що можна отримувати результати в обох представленнях. В такому разі XML стає більш дружнім до програміста переднім шаром "дійсної" бази даних, що використовується для збереження й пошуку даних (в якості прикладу такої техніки можна навести Oracle's XSQL).

5.4. Java-технологія в Internet

Створення мови Java — це дійсно один із самих значних кроків вперед в області розробки середовищ програмування за останні 20 років. Мова HTML була необхідна для статичного розміщення сторінок у WWW. Мова Java потрібна для якісного "стрибка" в створенні інтерактивних продуктів для мережі Internet.

Три ключових елементи об'єдналися в технології мови Java і зробили її відмінною від усього, що існує на сьогоднішній день:

- Java надає для широкого використання свої ***аплети*** (*applets*) — невеликі, надійні, динамічні, що не залежать від платформи, активні мережеві програми, що вбудовуються у Web-сторінки. Аплети Java можуть надбудовуватися і поширюватися користувачами з такою ж легкістю, як будь-які документи HTML;
- Java використовує переваги об'єктно-орієнтованої розробки програм, сполучаючи простий і знайомий синтаксис з надійним і зручним у роботі середовищем розробки. Це дозволяє широкому колу програмістів швидко створювати нові програми і нові аплети;
- Java надає програмісту багатий набір класів об'єктів для ясного абстрагування багатьох системних функцій, які використовуються при роботі з вікнами, мережею і для вводу-виводу. Ключова риса цих класів полягає в тому, що вони забезпечують створення широкого спектру системних інтерфейсів, які незалежать від платформи абстракцій, що використовуються.

5.4.1. Історія створення Java

Мова Java народилась як частина проекту створення сучасного ПЗ для різних побутових приладів Реалізація проекту була почата мовою C++, але незабаром виник ряд проблем, найкращим засобом боротьби з якими була зміна самого інструментарію — мови програмування. Стало ясно, що необхідна платформенно незалежна мова програмування, що дозволяла б створювати програми, які не доводилось би компілювати окремо для кожної архітектури і можна було б використовувати на різних процесорах під різними операційними системами.

Народженню мови Java передувала така історія. У 1990 році розробник із компанії Sun Microsystems Патрік Нофтон зрозумів, що йому набридло підтримувати сотні різних інтерфейсів програм, що використовувались у компанії, і він повідомив виконавчому директору Sun Microsystems і своєму другу Скоттові Макнілі про свій намір перейти працювати в компанію NeXT. Макнілі, у свою чергу, попросив Нофтона скласти список причин свого невдоволення.

Нофтон, хоча і не розраховував на те, що хтось зверне увагу на його лист, все ж таки виклав свої претензії, нещадно розкритикувавши недоліки Sun Microsystems, зокрема, архітектуру ПЗ NeWS, яка на той час розроблялась. На подив Нофтона, його лист мав успіх, він був розісланий всім провідним інженерам Sun Microsystems, що не забарились відгукнутися і висловити гарячу підтримку своєму колезі і схвалення його поглядів на ситуацію в компанії. Звернення було схвалено і вищим керівництвом компанії, а саме Білом Джоєм, засновником Sun Microsystems, і Джеймсом Гослінгом, начальником Нофтона.

У той день, коли Нофтон повинен був піти з компанії, було прийняте рішення про створення команди Green ведучих розробників з метою подолання викладених у листі недоліків. Команда із шести чоловік прийняла рішення стосовно розробки нової об'єктно-орієнтованої мови програмування, що була названа Oak (дуб), на честь дерева, що росло під вікном Гослінга.

Незабаром компанія Sun Microsystems перетворила команду Green у компанію First Person. Нова компанія мала найцікавішу концепцію, але не могла знайти їй застосування Після ряду невдач ситуація для компанії різко змінилася: був анонсований браузер Mosaic, з якого почався бурхливий розвиток Internet.

Нофтон запропонував використовувати Oak у створенні Internet-програм. Так Oak став самостійним продуктом, незабаром був написаний Oak-компілятор і Oak-браузер WebRunner. У 1995 році компанія Sun Microsystems прийняла рішення оголосити про новий продукт, перейменувавши його в Java (єдине розумне пояснення назві — любов програмістів до кави). WebRunner був перейменований у HotJava, і компанія Netscape стала на підтримку Java-продуктів.

5.4.2. Безпека

World Wide Web "висунула" Java на передній край програмування, і Java, в свою чергу, сильно вплинула і навіть змінила "обличчя" Internet, розширивши спектр об'єктів, які можуть розповсюджуватись у кіберпросторі. Програми нової форми — аплети — завантажуються з віддаленого серверу і можуть запускатись динамічно, тобто без участі користувача. До появи Java такий підхід був неприпустимий з міркувань безпеки та портабельності.

В архітектурі аплетів зроблено ряд штучних обмежень, які роблять їх цілком безпечними. Перш за все, Java є інтерпретованою мовою і простір ресурсів Java-програми обмежений так званою віртуальною Java-машиною (VJM), яка може контролювати поведінку програми і захищати систему від сторонніх ефектів, які можуть виникати з вини аплета. Крім того, в мові Java є додаткові обмеження, які не дозволять аплету стати "троянським конем". Зокрема, Java-аплет не може отримати доступ до локального жорсткого диску. При такій спробі генерується виключна ситуація.

5.4.3. Ефективність

Оскільки аплети Java інтерпретуються, а не компілюються, то їх виконання на різних платформах значно полегшується. В цьому випадку достатньо створити для кожної платформи виконуючу Java-систему. Якщо існує така система для даної операційної системи, то будь-яка Java-програма може виконуватись в даному середовищі без додаткової компіляції на цій платформі. Проте Java не є інтерпретованою мовою в чистому розумінні.

Програма на Java компілюється. Результатом роботи компілятора Java є байткод (*bytecode*). ***Байткод*** — це оптимізований набір команд, призначений для виконання уявним пристроєм — віртуальною Java-машиною. В такий спосіб витрати на інтерпретацію зводяться до мінімуму, оскільки байткод вже є оптимізованим, і досягається досить висока продуктивність Java-програм.

Наведені вище особливості дають підставу розглядати Java не як ще одну мову програмування, а як окрему інформаційну технологію. Таким чином інтерпретація — це найлегший шлях до перенесення програм реалізованих в Java технології. Незважаючи на те, що мова Java була розроблена в розрахунку на інтерпретацію, технічно немає нічого такого, що б перешкоджало компіляції байткоду в виконуючий код.

До байткоду, який пересилається по мережі, застосовується динамічна компіляція, але це ніяк не впливає на переносимість та безпеку, оскільки роботу програми все ще контролює виконуюча система. Такий підхід застосовано в багатьох виконуючих система Java, що забезпечує продуктивність на рівні оптимізованого коду C++.

Мова Java є однією з наймолодших в сімействі мов програмування і була розроблена з розрахунку на те, щоб професійний програміст міг легко її опанувати та ефективно використовувати. За основу Java взятий синтаксис C++ — безсумнівно однієї з найбільш популярних мов програмування сучасності. Проте, Java — це цілком самостійна мова програмування, і при її створенні не йшлось про будь-яку сумісність з C++. Тому деякі механізми реалізовані в Java інакше, а деякі зовсім відсутні.

Ідеологічно ж Java побудована дещо інакше ніж C++. Розробники Java орієнтувались на досвід розробки програм на C++ і прагнули позбутись її недоліків. Так, в Java відсутнє перевантаження операторів а також автоматичне приведення несумісних типів.

Взагалі, інтерфейси Java більш прості та прозорі для розуміння. Написати на Java програму з графічним інтерфейсом значно легше. Звичайно, простота інтерфейсів компенсується меншою гнучкістю, бібліотека Java не така багата, як стандартні бібліотеки C/C++. Але згадаймо, що мова Java задумана для використання на рі-

зних платформах і тому реалізує в собі найбільш стандартні можливості задля полегшення адаптації під конкретне середовище.

5.4.4. Об'єктно-орієнтована спрямованість

Від C++ Java успадкувала потужний механізм об'єктно-орієнтованого програмування. Оскільки Java розроблялась "з чистого аркуша", тобто не було потреби забезпечувати сумісність з попередніми версіями, розробники мали повну свободу мислення. Як наслідок був сформований ясний і прагматичний підхід до об'єктів.

Вільно переймаючи ідеї, які реалізовувалися протягом останніх десятиріч, мові Java вдалося знайти рівновагу між парадигмою "все є об'єктом" і прагматичним підходом. Об'єктна модель Java проста і легко розширюється, в той час як прості типи (цілі) зберігаються як дані, які не є об'єктами, що дозволяє значно підвищити швидкість при їх обробці.

5.4.5. Доступність інструментарію та ефективність розробок

Зазначена вище простота програмування на Java є причиною того, що розробки на Java коштують дешевше, ніж на аналогічних більш потужних мовах програмування. Цьому ж сприяє і переносимість програм на Java, оскільки ліквідуються витрати пов'язані з адаптацією програми на конкретній платформі. До того ж інтегровані програми-оболонки для розробки Java-програм коштують набагато дешевше ніж аналогічні продукти для C++ та Delphi. А набір інструментарію для пакетної компіляції Java програм JDK (Java Development Kit) є взагалі безкоштовний. Тому платформу Java можна рекомендувати як ідеальну для створення некомерційних програмних продуктів, зокрема для галузі освіти.

5.4.6. Багате об'єктне середовище

Середовище Java — це щось набагато більше, ніж просто мова програмування. В неї вбудований набір ключових класів, що містять основні абстракції реального світу, з яким доведеться мати справа у прикладних програмах. Основою популярності Java є вбудовані класи-абстракції, що зробили цю мову дійсно незалежною від платформи. Бібліотеки, подібні MFC/COM, OWL, VCL, NeXTStep, Motif і OpenDoc, прекрасно працюють на своїх платформах, однак сьогодні головною платформою стає Internet.

5.4.7. Стійкість до помилок

Багатоплатформеність середовища Web висуває надзвичайно високі вимоги до надійності програм. Як наслідок, при розробці Java пріоритет був відданий можливості створення стійких до помилок програм.

Як вже згадувалось, Java є строго типізованою мовою програмування. Виконуюча система Java бере на себе "прибирання сміття", тобто автоматично звільняє пам'ять, яка була розподілена динамічно. Звичайно, це дещо знижує ефективність коду, але запобігає типових помилок, коли програміст забуває звільнити виділену пам'ять, або, навпаки, звільняє пам'ять, яка ще використовується.

Java підтримує об'єктно-орієнтовану обробку виключних ситуацій подібно до C++. Але на відміну від C++, в Java обробка виключних ситуацій є обов'язковою. Тобто неможливо скомпілювати програму, яка відкриває файл, не обробивши можливі помилки типу "файл не знайдено", які виникають при цьому. Добре написана Java-програма може сама обробляти всі помилки під час виконання.

5.4.8. Підтримка багатопоточності

Java розроблялась з орієнтацією на вимоги до створення інтерактивних програм, які працюють з Internet. З цією метою Java підтримує багатопоточне програмування, яке дозволяє легко розробляти програми, що виконують багато процесів одночасно Виконання Java-програми засновано на елегантному, але в той самий час високоорганізованому рішенні багатопроцесової синхронізації, яке дозволяє створювати високоефективні інтерактивні системи.

5.4.9. Незалежність від архітектури

Основним питанням для розробників Java стало питання довготривалості та переносимості. Одна з головних проблем, з якою зустрілися програмісти, полягала в відсутності гарантій того, що написана сьогодні програма завтра працюватиме з тим же успіхом, причому на тій самій машині. Оновлення операційної системи, модернізація процесора та зміна об'єму оперативної пам'яті можуть призвести до збою програми.

Розробники Java, прагнули змінити цю ситуацію і прийняли декілька важливих рішень відносно мови Java та процесу виконання Java-програми. Їх мета полягала в тому, щоб "одного разу написане працювало всюди, в будь-який час і завжди". Внаслідок цього Java є системою, яка легко розширюється за рахунок створення нових стандартних класів та бібліотек.

5.4.10. Переваги інтерпретованості

Як вже згадувалось, Java дозволяє створювати незалежні від платформи програми шляхом компіляції в проміжне представлення, яке називається байткодом. Багато попередніх спроб знати розв'язання проблеми незалежності від платформи були зроблені за рахунок продуктивності. Інтерпретуючі системи, подібні до BASIC, Perl, страждають на майже неподоланний дефіцит продуктивності. Це було враховано при створенні Java.

Незважаючи на те, що Java є інтерпретованою мовою, генерація байткодів була ретельно оптимізована в такий спосіб, щоб одержуваний байткод, можна було легко перекладати в машинний код, який працює з дуже високою продуктивністю. Виконуючі системи такого роду не втрачають жодних переваг переносимого коду.

5.4.11. Розподіленість

Мова Java призначена для створення програм, які працюють в розподіленому середовищі Internet на базі протоколів TCP/IP. Насправді доступ до ресурсів відрізняється від доступу к файлу. Крім того в Java реалізований засіб передачі повідомлень в межах внутрішнього адресного простору. Це дозволяє забезпечити віддалене

виконання процедур. Ці інтерфейси включені у пакет RMI (Remote Method Invocation), який привносить високий рівень абстракції в програмування для середовища "клієнт-сервер".

Java-програми несуть у собі значний обсяг інформації про типи часу виконання (*run-time type information*), яка використовується для дозволу доступу до об'єктів під час роботи програми. Це дозволяє забезпечити безпечну та оптимальну динамічну компоновку. В такий спосіб досягається захищеність середовища виконання аплетів.

5.4.12. Пакети Java API

Перерахуємо пакети Java API:

- `java.applet` — класи для реалізації аплетів;
- `java.awt` — класи для роботи з графікою, текстом, вікнами і GUI;
- `java.awt.datatransfer` — класи для забезпечення передачі інформації;
- `java.awt.event` — класи й інтерфейси для обробки подій;
- `java.awt.image` — класи для обробки зображень;
- `java.awt.peer` — GUI для забезпечення незалежності від платформи;
- `java.beans` — API для моделі компонентів JavaBeans;
- `java.io` — класи для різних типів вводу-виводу;
- `java.lang` — класи ядра мови (робота з рядками, тригонометричні функції, обробка виключень, процеси і т.ін);
- `java.lang.reflect` — класи Reflection API;
- `java.math` — класи для арифметичних операцій довільної точності;
- `java.net` — класи для роботи в Internet (сокети, протоколи, URL);
- `java.rmi` — класи, зв'язані з RMI (віддалений виклик процедур);
- `java.rmi.dgc` — класи, зв'язані з RMI;
- `java.security` — класи для забезпечення безпеки;
- `java.security.acl` — класи для забезпечення безпеки (`acl`);
- `java.security.interfaces` — класи для забезпечення безпеки (`interfaces`);
- `java.sql` — класи для взаємодії з базами даних;
- `java.text` — класи для забезпечення багатомовної підтримки;
- `java.text.resources` — класи для забезпечення багатомовної підтримки (`resources`);
- `java.util` — різноманітні корисні типи даних (стеки, словники, хеш-таблиці, дати, генератор випадкових чисел);
- `java.util.zip` — класи для забезпечення архівації.

5.5. Побудова аплетів в Internet

Аплети є самим популярним типом програм, що створюються Java-програмістами. Головною причиною цього є те, що їх можна розміщувати на Web-вузлах. Аплети пишуться на мові Java. Вони є частиною Web-сторінки, подібно зображенню чи рядку текста. Аналогічно тому, як браузер забезпечує вивід зображення, на яке в HTML-документі є зсилка, Java-сумісний браузер розпізнає і виконує аплет.

Аплети по суті — це міні-програми, що виконуються в середовищі Java-сумісного браузера, наприклад Netscape Navigator, Microsoft Internet Explorer або HotJava. Унікальність аплета полягає в його можливості підключатися до інформаційної супермагістралі. Насправді ж аплети є першим кроком до того, щоб зробити Internet справжнім розширенням локальної файлової системи. При прогляданні Web-сторінки, яка містить аплети, ці аплети можуть поступати з будь-якого кінця WWW.

Між браузером, який відображає аплет, і системою, яка цей аплет надає, існують відносини типу "клієнт-сервер". Клієнт — це комп'ютер, який звертається до служб іншої системи; сервер — це комп'ютер, що забезпечує роботу таких служб. В випадку Java-аплетів клієнтом являється комп'ютер, який відображає HTML-документ, що містить зсилку на деякий аплет, а сервер передає аплет клієнту і дозволяє тим самим клієнту використовувати цей аплет.

Може виникнути запитання: "А як же щодо безпеки використання аплетів? Якщо мережа Internet наповнена аплетами, в будь-який момент можуть виникнути проблеми з вірусами". Але боятися цього не слід, бо Java-аплети — безпечний спосіб розповсюдження програм через Internet. Це пояснюється тим, що інтерпретатор Java не запустить аплет до тих пір, доки не впевниться в тому, що байткоди аплета не пошкоджені чи не модифіковані.

Більш того, інтерпретатор визначає, чи відповідає байткодове представлення аплета всім правилам мови Java. Наприклад, Java-аплет ніколи не може використовувати показчик для доступу до закритої для нього комп'ютерної пам'яті. І нарешті, Java-аплети не тільки захищені, вони практично не в змозі пошкодити систему.

Оскільки браузери Web першопочатково розроблялись для відображення HTML-документів, для втілення аплетів Java в середовище браузера необхідний HTML-тег, який викликає ці аплети. Нижче наведено приклад подібного HTML-тега `<APPLET>`:

```
<applet code=TextEdit.class width=575 height=350></applet>.
```

Цей приклад містить основні необхідні атрибути. Однак є ще й інші атрибути і особливості тега `<APPLET>`, що потребують додаткового розгляду.

5.5.1. Цикл завантаження аплетів

Оскільки посилання на аплети Java містяться всередині HTML-документів і виконуються Web-браузером, то не повинен викликати здивування той факт, що аплети розташовуються на сервері, як і самі HTML-документи. Існує особлива послідовність подій, що виникають в тому випадку, коли Java-сумісний браузер завантажує HTML-документ і знаходить в ньому тег `<APPLET>`. Послідовність операцій при цьому наступна:

1. Завантажується HTML-файл.
2. Виявляється тег `<APPLET>`.
3. Файл класу, вказаного в тегу `<APPLET>`, завантажується з сервера.
4. Розпізнаються і завантажуються класи, на які зсилається клас `Applet`.
5. Клас `Applet` викликає методи `init()` і `start()`.
6. Якщо все проходить нормально, аплет відображається у вікні браузера (або поза ним, якщо він використовує власний кадр — `frame`).

Після цього код аплета (файли класу), що виконується, потрапляє на комп'ютер, що звернувся до Web-сервера. Ці файли класів завантажуються і виконуються після того, як користувач підключився до Web-вузла і викликав HTML-документ, що містить зсилку на аплет (тег <APPLET>).

Наприклад, щоб підключити до Web-сторінки аплет KubikRubika, потрібно включити в HTML-документ наступний тег:

```
<applet
  codebase=MyApplets
  code=KubikRubika.class
  width=120
  height=180>
  <b>Якщо б у Вас був Java-сумісний браузер, то Ви
     могли б проглянути аплет.
  </b>
</applet> .
```

Показаний тег повідомляє браузеру, що файл KubikRubika.class (відкомпільовані Java-аплети мають розширення *.class) розташований в каталозі MyApplets. Width — висота аплета, а height — його ширина. Безпосередньо перед закриваючим тегом </applet> знаходиться альтернативний текст. Він буде відображатися, коли браузер, несумісний з Java, не зможе відобразити аплет.

Мова програмування Java і бібліотеки дозволяють створювати будь-які аплети — прості і складні. Найпростіший Java-аплет може складатися всього з декількох рядків, і може бути написаний у вигляді:

```
import java.applet.*;
public class MyApplet extends Applet
{
} .
```

Перший рядок коду повідомить компілятору Java про те, що даний аплет буде використовувати деякі або всі класи, описані в пакеті applet. Всі основні властивості аплета реалізовані в цих класах, тому і можна написати робочий аплет, використовуючи всього кілька рядків.

В другому рядку коду оголошується новий клас MyApplet. Цей новий клас оголошується як public, тому до даного класу можна звертатись, коли аплет буде працювати в середовищі Web-браузера чи в програмі Appletviewer. Якщо клас аплета не оголосити, як public, то код буде компілюватись без помилок, однак аплет не запуститься. Тому всі класи аплетів повинні бути оголошені як public. Для того, щоб запустити аплет, достатньо створити HTML-документ, схожий на наступний:

```
<title>Applet Test Page</title>
<h1>Applet Test Page</h1>
<applet
  code="MyApplet.class"
  width=250
  height=250
  name="MyApplet">
</applet> .
```

Однак, якщо запустити цей аплет, то в браузері нічого відображатись не буде, бо аплет не виконує ніяких корисних дій.

5.5.2. Етапи життєвого циклу аплета

Кожен аплет успадковує від класу `Applet` деякі властивості, задані по замовчуванню. Частіше за все по замовчуванню не виконується ніяких дій, якщо не перевизначити деякі методи класу `Applet` для того, щоб розширити базові функціональні можливості аплета. Хоча може здатися, що простий аплет не робить нічого особливого, багато операцій виконуються непомітно для користувача. Деякі з цих операцій важливі для розуміння роботи аплета, а на деякі можна не звертати уваги.

Будь-який аплет має життєвий цикл, який складається з чотирьох етапів; кожен етап має відповідний метод, який можна перевизначити і отримати доступ до конкретного етапу життєвого циклу. Розглянемо ці етапи:

1. **Етап ініціалізації** (*initialization stage*). На цьому етапі створюється і завантажується об'єкт аплета. В цей момент зручно створювати об'єкти для аплета, а також ініціалізувати значення, необхідні при роботі аплета. На протязі життєвого циклу ініціалізація виконується тільки один раз. Можна втрутитись в процес ініціалізації, перевизначивши метод `init()` класу `Applet`.
2. **Етап запуску** (*start stage*). На цьому етапі система починає виконання аплета. Етап запуску може слідувати одразу ж після етапу ініціалізації чи після повторного запуску аплета. Як правило, це відбувається тоді, коли користувач, працюючи з Web-браузером, повертається до сторінки, що містить аплет, після перегляду якої-небудь іншої сторінки. На відміну від етапу ініціалізації, етап запуску на протязі життєвого циклу аплета може виконуватися багаторазово. Для того, щоб виконувався власний код запуску, необхідно перевизначити метод `start()` класу `Applet`.
3. **Етап зупинки** (*stop stage*). Етап зупинки є протилежністю етапу запуску. Інтерпретатор виконує фазу зупинки, коли аплет більше не відображається на екрані, наприклад, коли користувач звертається до іншої Web-сторінки. По замовчуванню, на цьому етапі аплет продовжує роботу в фоновому режимі. Якщо на етапі зупинки потрібно виконати інші дії, то слід перевизначити метод `stop()` класу `Applet`.
4. **Етап знищення** (*destroy stage*). Цей етап за призначенням протилежний етапу ініціалізації і починається тоді, коли система збирається видалити аплет з пам'яті. Подібно фазі ініціалізації, етап знищення виконується тільки один раз. Якщо аплет використовував ресурси, які перед знищенням аплета необхідно звільнити, то це потрібно робити на етапі знищення. Цю фазу можна змінити, перевизначивши метод `destroy()` класу `Applet`.

Клас `Applet`, від якого породжувались класи аплетів, визначав методи життєвого циклу стандартним чином, перевизначеним в системі Java. Нижче показаний невеликий аплет, в якому всі методи перевизначені (включаючи метод `paint()`, який перевизначається, коли потрібно, щоб аплет мав свою область відображення, і виконується кожен раз, коли область відображення аплета повинна проектуватись

на екран, а також кожен раз, коли область аплета змінюється чи відновлюється); це дає можливість користувачеві виконати необхідні йому дії на кожному етапі життєвого циклу аплета.

```
import java.Applet.*;
import java.Avt.*;
public class MyApplet2 extends Applet
{
  public void init( )
  {
    // Тут міститься код етапу ініціалізації
  }
  public void start( )
  {
    // Тут міститься код циклу етапу запуску
  }
  public void paint(Graphics g)
  {
    // Тут міститься код етапу перемальовування
  }
  public void stop( )
  {
    // Тут міститься код етапу зупинки
  }
  public void destroy( )
  {
    // Тут міститься код етапу знищенння
  }
} .
```

Слід звернути увагу на те, що для перевизначення методу `paint()` необхідно імпортувати бібліотеки `java.avt.*`, що містять інформацію про клас `Graphics`. Клас `Graphics` дозволяє виводити інформацію і графіку в області відображення аплета (чи на так званому полотні).

Також аплети мають свої параметри, які дозволяють користувачу, який не має доступу до коду, змінювати аплет, якщо це пердбачено програмістом.

5.5.3. Зображення

Переглядаючи різноманітні аплети, що існують в WWW, не можна не звернути уваги на розвинуту графіку. Оскільки мова Java орієнтована на те, щоб максимально спростити створення аплета, то класи Java справляються майже з усіма труднощами, пов'язаними з виводом зображень та відтворенням звуків. Мова Java може завантажувати графічні файлові формати `.GIF` і `.JPRG`.

Першим кроком при відображенні графічної картинки в аплеті є завантаження зображення з диску. Для цього необхідно створити об'єкт на базі Java-класу `Image`, попередньо створивши URL-об'єкт, що вказує на місцезнаходження графічного файлу. Можна просто вписати URL-адресу зображення в вихідний текст програми, тоді

доведеться змінювати і перекомпільовувати апплет кожен раз, коли буде переміщуватись цей графічний файл в інший каталог на диску. Більш зручний спосіб створення URL-адреси зображення — викликати метод `getDocumentBase()` або `getCodeBase()`.

Перший метод повертає URL-адресу каталогу, з якого був завантажений поточний HTML-файл, а другий дозволяє отримати URL-адресу каталогу, з якого був завантажений аплет. Якщо зображення зберігаються в тому ж каталозі (чи підкаталозі цього каталогу), де знаходяться HTML-файли, то для отримання URL-адреси зображення краще скористатись методом `getDocumentBase()`.

Наприклад, якщо HTML-документи знаходяться в каталозі `C:\APPLETS`, а необхідний файл `IMAGE.GIF` зберігається в підкаталозі `IMAGES` каталогу `APPLETS`, то викликавши метод `getDocumentBase()`, можна отримати відповідну базову URL-адресу. Цей виклик має наступний вигляд:

```
URL url = getDocumentBase( );.
```

Як буде показано нижче, після отримання URL-адреси, можна завантажити файл, використовуючи цю адресу разом з відносною адресою зображення, яке в даному випадку буде знаходитися в підкаталозі `IMAGES\IMAGE.GIF`. Тоді повна URL-адреса буде такою: `FILE:\C:\APPLETS\IMAGES\IMAGE.GIF`. Якщо було прийняте рішення перемістити загальні файли в каталог `MYHOMEPAGE`, то виклик метода `getDocumentBase()` дасть URL-адресу цього нового каталогу, при цьому змінювати початковий текст аплета не доведеться. Оскільки була включена відносна URL-адреса файлу зображення, то нова URL-адреса буде наступною: `FILE:\C:\MYHOMEPAGE\IMAGES\IMAGE.GIF`.

Метод `getCodeBase()` працює аналогічно методу `getDocumentBase()`, за винятком того, що він повертає URL-адресу каталогу, з якого завантажувався аплет.

Коли є базова URL-адреса зображення, то можна завантажувати його і створювати об'єкт класу `Image`. Можна виконати ці задачі одночасно, викликавши метод `getImage()` аплета наступним чином:

```
Image image = getImage(baseURL, relLocation);.
```

Для виводу зображення достатньо викликати метод `drawImage()` об'єкта класу `Graphics`:

```
g.drawImage(myImage, x, y, width, height, this);.
```

Аргументи цього метода: виведений об'єкт зображення, координати X та Y зображення на екрані, ширина і висота зображення та посилання на аплет `this`.

Може виникнути запитання про те, як визначити ширину і висоту зображення. Клас `Image` має два методи: `getWidth()` і `getHeight()`, що повертають ширину і висоту зображення відповідно. Тому повний код для виводу зображення може мати наступний вигляд:

```
int width = image.getWidth(this);
int height = image.getHeight(this);
g.deawImage(image, x, y, width, height, this);.
```

Як видно, методи `getWidth()` і `getHeight()` приймають єдиний аркумент — посилання на аплет `this`.

Тепер можна створити аплет для виводу зображення на екран. Нижче представлений вихідний текст аплета ImageApplet, який виводить невелике зображення і використовує рішення, описані вище.

```
import java.awt.*;
import java.Applet.*;
import java.net.*;
public class ImageApplet extends Applet
{
  Image snace;
  public void init( )
  {
    URL codeBase = getCodeBase( );
    snake = getImage(codeBase, "snake.gif");
    resize(250,  250);
  }
  public void paint(Graphics g)
  {
    int width = snake.getWidth(this);
    int height = snake.getHeight(this);
    g.drawRect(snake, 57, 57, width, height, this);
  }
} .
```

Слід звернути увагу на те, що аплет імпортує класи з пакету net — саме там розміщений клас URL. Якщо не включити цей рядок на початку програми, то клас URL не буде знайдений, і компіляція аплета не завершиться.

5.5.4. Відтворення звуку

Подібно тому, як існує багато типів графічних файлів, є велика кількість форматів звукових файлів. Але якщо говорити про аплети, то потрібно знати тільки про один тип аудіофайлів — тих, що мають розширення файлу .AU. Такі звукві файли були розповсюджені на UNIX-машинах і в даний час є єдиним типом аудіофайлів, які Java може завантажувати і відтворювати.

Якщо потрібно відтворити звуковий файл, то достатньо за допомогою методів getDocumentBase() або getCodeBase() отримати URL-адресу, а потім викликати метод play() для програвання звуку. Виклик метода play() має наступний вигляд:

```
play(baseURL, relLocation);.
```

Метод play() має два параметри: URL-адресу, отриману за допомогою метода getDocumentBase() або getCodeBase(), і відносну адресу звукового файлу.

Тепер можна написати аплет для відтворення звукового файлу. Цей аплет, SoundApplet, має наступний вигляд:

```
import java.awt.*;
import java.Applet.*;
import java.net.*;
public class SoundApplet extends Applet
```

```
{
  Button button;
  public void init( )
  {
    BorderLayout layout = new BorderLayout( );
    SetLayout(layout);
    Font font = new Font("TimesRoman", Font.BOLD, 32);
    SetFont(font);
    Button = new Button("Play Sound");
    add("Center", Button);
    resize(250, 250);
  }
  public boolean action(Event evt, Object arg)
  {
    if (evt.target instanceof Button)
    {
      URL codeBase = getCodeBase( );
      play(codeBase, "spacemusic.au");
    }
  }
  return true;
} .
```

Хоча простіше за все завантажувати і відтворювати звуки за допомогою метода `play()`, цей метод має мало можливостей: він може лише відтворювати звуковий файл від початку до кінця. Якщо необхідно хоч як-небудь керувати звуком, то можна створити об'єкт калсу `AudioClip` і використовувати методи цього об'єкта. Нажаль, навіть клас `AudoiClip` має недостатньо можливостей, хоча він дозволяє програвати, зупиняти звуковий файл і відтворювати його циклічно.

Для створення об'єкта класу `AudioClip` можна викликати метод `getAudioClip()`:

```
AudioClip audioClip = getAudioClip(baseURL, relLocation);.
```

Параметрами цього метода є базова URL-адреса і відносна адреса звукового файлу.

Після того, як об'єкт класу `AudioClip` створений і завантажений, можна використовувати його методи `Play()`, `Stop()` і `Loop()` для керування звуком. Метод `Play()` одноразово програє звуковий файл від початку до кінця, метод `Stop()` в будь-який момент зупиняє відтворення, а метод `Loop()` повторює звук до тих пір, доки його не зупинять.

5.5.5. Події

Багато компонентів Java, наприклад кнопки, створюють події, які можна обробляти за допомогою методу `action()`. Події ж, які генеруються мишею, можна аналізувати, використовуючи методи `mouseDown()` та `mouseMove()`.

Крім цих методів, при написанні аплетів можна використати наступні методи:

- **mouseEnter()** — метод викликається Java-інтерпретатором тоді, коли покажчик миші потрапляє у вікно аплета;
- **mouseExit()** — метод викликається, коли покажчик миші виходить з вікна аплета;
- **mouseDrag()** — викликається тоді, коли користувач переміщує мишу, утримуючи натиснутою ліву кнопку.

Нижче наведений приклад коду аплета, який виводить на екран повідомлення "Mouse entered", коли покажчик миші попадає у вікно аплета, і повідомлення "Mouse exited" при виході пока;чика миші з вікна аплета:

```
import java.awt.*;
import java.Applet.*;
public class MouseApplet extends Applet
{
  String msgStr;
  public void init( )
  {
    msgStr = "";
    Font font = new Font("TimesRoman", Font.BOLD, 32);
    SetFont(font);
    resize(400, 300);
  }
  public void paint(Graphics g)
  {
    g.drawString(msgStr, 40, 120);
  }
  public boolean mouseEnter(Event evt, int x, int y)
  {
    msgStr = "Mouse entered";
    repaint( );
    return true;
  }
  public boolean mouseExit(Event evt, int x, int y)
  {
    msgStr = "Mouse exited";
    repaint( );
    return true;
  }
} .
```

Події клавіатури можна обробляти, описавши в аплеті методи `keyDown()` та `keyUp()`. Метод `keyDown()` викликається кожен раз, коли користувач натискає якунебудь клавішу на клавіатурі, а метод `keyUp()` — коли ця клавіша відпускається. Обидва ці методи отримують в якості параметра код натиснутої чи відпущеної клавіші.

5.5.6. Висновок

Отже, як бачимо, написання аплетів для WWW здійснюється на мові Java і, завдячуючи її об'єктно-орієнтованій структурі, є нескладною процедурою. Мова Java дозволяє користувачеві використовувати вже створені класи, навіть не знаючи того, як вони побудовані. Завдяки цьому найпростіший аплет на мові Java складається всього-навсього з двох рядків (не враховуючи фігурних дужок). Також, використовуючи аплети, можна відтворювати звук та виводити на екран зображення.

5.6. Сокети в Internet (WinSockets API)

Для забезпечення мережевих комунікацій використовуються сокети (*sockets*). ***Сокет*** — це кінцева точка мережевих комунікацій. Кожен сокет, що використовується, має тип і асоційований з ним процес. Сокети існують всередині комунікаційних доменів. Домени — це абстракції, що мають конкретну структуру адресації і множину протоколів, які визначають різні типи сокетів всередині домену. Прикладами комунікаційних доменів можуть бути: UNIX-домен, Internet-домен і т. ін. Надалі будемо розглядати тільки Internet-домен.

В Internet-домені ***сокет*** — це комбінація IP-адреси і номера порту, що однозначно визначає окремий мережевий процес у всій глобальній мережі Internet. Два сокети — один для комп'ютера-серверу, інший для комп'ютера-клієнта — визначають з'єднання для протоколів, орієнтованих на встановлення зв'язку, таких, як TCP.

При роботі з сокетом здійснюється відправка іншому комп'ютеру послідовності символів. Застосовуючи цей метод, можна посилати як прості повідомлення, так і файли. До того ж контролювати безпомилковість передачі немає потреби — це гарантується роботою сокетів.

Роботу з сокетами розглянемо на прикладі реалізації інтерфейса до WinSock ets API в Delphi. Він складається з двох частин: серверного та клієнтського сокетів.

5.6.1. Серверний сокет (клас `TServerSocket`)

Що ж дозволяє робити сокетний сервер? За яким принципом він працює? Сервер, заснований на сокетному протоколі, дозволяє обслуговувати відразу декількох клієнтів. Для кожного підключеного клієнта сервер відкриває окремий сокет, по якому можна обмінюватися даними з клієнтом. Також можливе створення сокету для кожного підключення окремого процесу (*thread*).

Схема роботи сокетного серверу в Delphi-програмах наведена на рис. 5.2:

1. **Визначення властивостей `Port` і `ServerType`** — щоб до серверу могли нормально підключатися клієнти, потрібно, щоб порт, що використовується сервером, точно збігався з портом, що використовується клієнтом (і навпаки). Властивість `ServerType` визначає тип підключення.
2. **Відчинення сокету** — відкриття сокету і визначення порту. Тут виконується автоматичний початок очікування приєднання клієнтів (виклик методу `Listen`).
3. **Приєднання клієнта та обмін даними з ним** — тут підключається клієнт і йде обмін даними з ним.

4. **Від'єднання клієнта** — тут клієнт відключається і закривається його сокетне з'єднання із сервером.
5. **Закриття серверу і сокету** — по команді адміністратора сервер завершує свою роботу, закриваючи усі відкриті сокетні канали і припиняючи очікування підключень клієнтів.

Рис. 5.2. Схема роботи сокетного серверу в Delphi-програмах

Варто помітити, що пункти 3–4 повторюються багаторазово, тобто ці пункти виконуються для кожного нового підключення клієнта.

Опис компонента `TServerSocket`

Розглянемо основні властивості, методи і події компонента `TServerSocket`.

Властивості:

- **`Socket`** (тип `TServerWinSocket`) — сокет, через який є доступ до відкритих сокетних каналів.
- **`ServerType`** (тип `TServerType`) — тип серверу. Може приймати одне з двох значень:
  - `stNonBlocking` — синхронна робота з клієнтськими сокетами. При такому типі серверу можна працювати з клієнтами через події `OnClientRead` і `OnClientWrite`;
  - `stThreadBlocking` — асинхронний тип. Для кожного клієнтського сокетного каналу створюється окремий процес (thread).
- **`ThreadCacheSize`** (тип `Integer`) — кількість клієнтських процесів, що будуть кешуватись сервером. Тут вказується середнє значення в залежності від завантаження серверу. Кешування відбувається для того, щоб не створювати щоразу окремий процес і не знищувати закритий сокет, а залишити їх для подальшого використання.
- **`Active`** (тип `Boolean`) — показник того, активний в даний момент сервер, чи ні. Тобто, значення `True` вказує на те, що сервер працює і готовий до прийому клієнтів, а `False` — що сервер виключений. Щоб запустити сервер, потрібно просто присвоїти цій властивості значення `True`.
- **`Port`** (тип `Integer`) — номер порту для встановлення з'єднань. Порти у сервера і клієнтів повинні бути однаковими. Рекомендовані значення від 1025 до 65535, тому що значення від 1 до 1024 можуть бути зайняті системою.

- **Service** (тип String) — рядок, що визначає службу (ftp, http, pop і т. ін.), порт якої буде використаний. Це своєрідний довідник відповідності номерів портів різним стандартним протоколам.

Методи:

- **Open** — запускає сервер. По суті, цей метод ідентичний присвоєнню значення True властивості Active;
- **Close** — зупиняє сервер. Ця команда ідентична присвоєнню значення False властивості Active.

Події:

- **OnClientConnect** — виникає, коли клієнт встановив сокетне з'єднання і чекає відповіді сервера (OnAccept);
- **OnClientDisconnect** — виникає, коли клієнт від'єднався від сокетного каналу;
- **OnClientError** — виникає, коли поточна операція завершилась невдало, тобто відбулась помилка;
- **OnClientRead** — виникає, коли клієнт передав серверу які-небудь дані; доступ до цих даних можна одержати через переданий параметр Socket: TCustomWinSocket;
- **OnClientWrite** — виникає, коли сервер може відправляти дані клієнту по сокету;
- **OnGetSocket** — в процедурі обробки цієї події можна відредагувати параметр ClientSocket;
- **OnGetThread** — в процедурі обробки цієї події можна визначити унікальний процес для кожного окремого клієнтського каналу, присвоївши параметру SocketThread потрібну підзадачу TServerClientThread;
- **OnThreadStart** — виникає, коли підзадача запускається;
- **OnThreadEnd** — виникає, коли підзадача зупиняється;
- **OnAccept** — виникає, коли сервер приймає клієнта чи відмовляє йому в з'єднанні;
- **OnListen** — виникає, коли сервер переходить у режим очікування приєднання клієнтів.

Властивість TServerSocket.Socket

Отже, як же сервер може відсилати дані клієнту або приймати їх? В основному, якщо працюють через події OnClientRead і OnClientWrite, то спілкуються з клієнтом через властивість ClientSocket (TCustomWinSocket). Про роботу з цим класом можна прочитати в пункті про клієнтські сокети, тому що відправка/посилка даних через цей клас аналогічна — методи Send/Receive.

Оскільки ми розглядаємо сервер, то варто виділити деякі корисні властивості і методи:

- **ActiveConnections** (тип Integer) — кількість підключених клієнтів;
- **ActiveThreads** (тип Integer) — кількість працюючих процесів;
- **Connections** (тип array) — масив, що складається з окремих класів TClientWinSocket для кожного підключеного клієнта, наприклад, команда

```
ServerSocket1.Socket.Connections[0].SendText('Hello!');
```

відсилає першому підключеному клієнту повідомлення "Hello!"; команди для роботи з елементами цього масиву — `Send/Receive`;

- **`IdleThreads`** (тип `Integer`) — кількість вільних процесів; такі процеси кешуються сервером (див. `ThreadCacheSize`);
- **`LocalAddress`, `LocalHost`, `LocalPort`** — відповідно локальна IP-адреса, хост-ім'я, порт;
- **`RemoteAddress`**, **`RemoteHost`**, **`RemotePort`** — відповідно віддалена IP-адреса, хост-ім'я, порт;
- Методи **`Lock`** і **`UnLock`** — відповідно, блокування і розблокування сокету.

5.6.2. Клієнтський сокет `TClientSocket`

Схема роботи клієнтського сокету в Delphi-програмах наведена на рис. 5.3.

Рис. 5.3. Схема роботи клієнтського серверу в Delphi-програмах

Розберемо цю схему докладніше:

1. **Визначення властивостей `Host` і `Port`** – щоб успішно встановити з'єднання, потрібно присвоїти властивостям `Host` і `Port` компонента `TClientSocket` необхідні значення. `Host` — це хост-ім'я (наприклад: `nitro.borland.com`) або IP-адреса (наприклад: `192.168.0.88`) комп'ютера, з яким треба з'єднатися. `Port` — номер порту (від 1 до 65535) для встановлення з'єднання. Звичайно номери портів беруться, починаючи з 1001, тому що номери менше 1000 можуть бути зайняті системними службами (наприклад, POP — 110).
2. **Відчинення сокету** — після того, як властивостям `Host` і `Port` призначені відповідні значення, можна приступити безпосередньо до відчинення сокету (сокет тут розглядається як черга, у якій містяться символи, що передаються від одного комп'ютера до іншого). Для цього можна викликати метод `Open` компонента `TClientSocket`, або присвоїти властивості `Active` значення `True`. Тут корисно створити процедуру обробки виняткової ситуації на той випадок, якщо з'єднатися не вдалося.
3. **Авторизація** — цей пункт можна пропустити, якщо сервер не вимагає введення яких-небудь логінів (імен користувача) та/або паролів. На цьому етапі серверу посилається логін і пароль. Але механізм авторизації залежить вже від конкретного серверу.

4. **Посилка/прийом даних** — це, власне і є те, для чого відкривалося сокетное з'єднання. Протокол обміну даними також залежить від серверу.
5. **Закриття сокету** — після всіх виконаних операцій необхідно закрити сокет за допомогою методу `Close` компонента `TClientSocket` (або присвоїти властивості `Active` значення `False`).

Опис компонента `TClientSocket`

Розглянемо основні властивості, методи і події компонента `TClientSocket`.

Властивості:

- **`Active`** (тип `Boolean`) — показує, відкритий сокет чи ні. Відповідно, значення `True` — відкритий, `False` — закритий. Ця властивість доступна для запису.
- **`Host`** (тип `String`) — рядок, що вказує на хост-ім'я комп'ютера, до якого треба підключитись.
- **`Address`** (тип `String`) — рядок, що вказує на IP-адресу комп'ютера, до якого треба підключитись. На відміну від властивості `Host`, тут може знаходитись лише IP-адреса. Відмінність полягає в тому, що в разі призначення властивості Host символьного імені комп'ютера, то IP-адреса, що відповідає цьому імені, буде запитана у DNS.
- **`Port`** (тип `Integer`) — номер порту, до якого треба підключитись. Допустимі значення — від 1 до 65535.
- **`Service`** (тип `String`) — рядок, що визначає службу (ftp, http, pop і т. ін.), до порту якої відбудеться підключення. Це своєрідний довідник відповідності номерів портів різним стандартам протоколів
- **`ClientType`** (тип `TClientType`) — тип з'єднання:
  - `ctNonBlocking` — асинхронна передача даних, тобто посилати і приймати дані по сокету можна за допомогою процедур обробки подій `OnRead` і `OnWrite`;
  - `ctBlocking` — синхронна (одночасна) передача даних. Події `OnRead` і `OnWrite` не працюють. Цей тип з'єднання корисний для організації обміну даними за допомогою потоків (тобто робота із сокетом як з файлом).

Методи:

- **`Open`** — відкриття сокету (аналогічно присвоєнню значення `True` властивості `Active`);
- **`Close`** — закриття сокету (аналогічно присвоєнню значення `True` властивості `Active`).

Події:

- **`OnConnect`** — як випливає з назви, ця подія виникає при вдалому встановленні з'єднання. Тобто в процедурі обробки цієї події вже можна починати авторизацію або прийом/передачу даних.
- **`OnConnecting`** — виникає протягом встановлення з'єднання. Відмінність від `OnConnect` у тім, що з'єднання ще не встановлене. Наприклад, такі проміжні події використовуються для відновлення статусу.

- **OnDisconnect** — виникає при закритті сокету, причому — як з боку клієнтської програми, так і з боку віддаленого комп'ютера (або через збій).
- **OnError** — виникає при помилці в роботі сокету. Слід зазначити, що ця подія не допоможе відловити помилку в момент відкриття сокету (по виклику методу `Open`). Для того, щоб уникнути видачі стандартного повідомлення про помилку, треба вкласти оператори відкриття сокету в середину блоку `try...except` (обробка виняткових ситуацій).
- **OnLookup** — виникає при спробі одержання від DNS IP-адреси зазначеного хоста.
- **OnRead** — виникає, коли віддалений комп'ютер відправив клієнту які-небудь дані. При виникненні цієї події можлива обробка даних.
- **OnWrite** — виникає, коли клієнту дозволений запис даних у сокет.

Головною властивістю для роботи з сокетами є `TClientSocket.Socket`. Вона надає доступ до таких найважливіших методів відправлення даних по сокетному каналу як `SendText`, `SendBuf` та отримання даних з сокету — `ReceiveText`, `ReceiveBuf`. Наприклад,

```
TClientSocket.Socket.SendText(s);.
```

5.7. Використання технології CGI

При першому ж знайомстві з WWW можна наштовхнутись на документи, що змусять задатись питанням: "Як вони зроблені?". Серед цих документів можуть бути, наприклад, форми зі зворотним зв'язком або інформацією про реєстрацію, малюнки, що дозволяють клацати на різних частинах зображення, сторінки, що показують кількість користувачів, що звернулися до документу, і утиліти, які дозволяють шукати специфічну інформацію в базі даних. У більшості випадків ці ефекти були досягнуті завдяки загальному шлюзовому інтерфейсу, більш відомому як ***CGI*** (Common Gateway Interface).

Одним із секретів інтерфейсу CGI є його простота. Він нескладний в проектуванні, і будь-хто з невеликим досвідом програмування може писати елементарні сценарії.

CGI — це частина Web-серверу, що може зв'язуватись з іншими програмами, які виконуються на сервері. За допомогою CGI Web-сервер може викликати програму, а також передати в неї визначені користувачем дані. Програма обробляє ці дані, і сервер передає відповідь програми на браузер користувачеві через Web.

Інтерфейс CGI не є чимось надзвичайним — це лише програмування з деякими спеціальними типами входу і декількома правилами щодо вихідної інформації програми. Все всередині — це програмування. Звичайно існують спеціальні методи, що є специфічними для CGI.

5.7.1. CGI-програми

CGI перетворює Web з простого сховища статичних гіпермедіа-документів в нове діалогове середовище, в якому користувачі можуть задавати питання й виконува-

ти програми. Розглянемо деякі із можливих програм, що можуть бути розроблені з використанням CGI.

Форми

Одне з найбільш поширених використань CGI — обробка форм. Форми — це частина HTML, що дозволяє користувачу вводити інформацію. Інтерфейс форм робить роботу з WWW інтерактивною для користувача і провайдера інформації.

Для створення форми існує безліч графічних компонентів типу перемикачів, полів вводу тексту, прапорців і списків вибору. Коли користувач завершує введення даних в формі, то для передачі інформації на сервер, що виконує програму, зв'язану зі специфічною формою, яка і "розшифровує" дані, використовується кнопка типу `Submit`.

Взагалі, форми використовуються для двох головних цілей: для збору інформації від користувача і для забезпечення взаємодії в зворотному напрямку. Наприклад, користувачу може бути подана форма, в якій є список різних документів, що доступні на сервері, або, наприклад, вибір розділів, серед яких можна знайти специфічну інформацію. CGI-програма може обробляти цю інформацію та повертати документи, що відповідають критеріям вибору користувача.

Програмні шлюзи

Програмні шлюзи (*gateways*) — це програми або сценарії, які використовуються для доступу до інформації, що не є безпосередньо відкритою для користувача. Наприклад, маємо базу даних Oracle, що містить інформацію про товари компанії, і хотіли б забезпечити доступ до цієї інформації користувачам з WWW. Як це зробити? Звичайно, користувачу неможна дати прямий URL до файлу бази даних, з огляду на те, що той може побачити які-небудь "значущі" дані. CGI забезпечує вирішення цієї проблеми у формі шлюзу. Ви можете використовувати мову типу oraperl або DBI-розширення до Perl, щоб формувати запити SQL, читати інформацію, що міститься в межах бази даних. Після отримання інформації, її можна форматувати і відправляти користувачу. У цьому випадку, CGI-програма виконує роль програмного шлюзу до бази даних Oracle.

Так само можна написати шлюзові програми до будь-якого іншого програмного забезпечення, наприклад, до інформаційних служб Internet — Archie, WAIS і NNTP (новини USENET). Крім того, можна підсилювати можливості шлюзу, використовуючи інтерфейс форм, щоб отримувати запити для пошуку від користувача, показувати динамічну або віртуальну інформацію.

Віртуальні документи

Віртуальне або динамічне створення документу — це серце CGI. Віртуальні документи можуть створюватись "миттєво" у відповідь на інформаційний запит користувача. Можна створити віртуальну HTML-сторінку, простий текст, зображення і навіть звукові документи. Простий приклад віртуального документа може бути дещо тривіальним:

```
Welcome to our server!
```

```
You are visitor #12345 from nic.univ.kiev.ua.
```

У цьому прикладі, є дві частини динамічної інформації: номер відвідувача та ім'я віддаленого комп'ютера.

З іншого боку, дуже складні віртуальні документи можуть бути створені програмами, що використовують комбінацію графічних бібліотек, шлюзів та форм.

5.7.2. Деякі приклади CGI-програм

Наведемо деякі Web-вузли, на яких розташовані найбільш відомі CGI-програми:

- **Lycos**. Розташована за адресою `http://www.lycos.com`. Цей сервер дозволяє користувачу шукати інформацію у Web. Lycos повертає динамічний документ, що містить інформацію, яка відповідає критеріям пошуку користувача.
- **Guestbook**. Заснована на формах програма, що дозволяє користувачам залишати повідомлення, які можуть побачити інші люди. Хоча у WWW є досить велика кількість "гостьових книг" (*guestboks*), це — одна з кращих. Її адреса: `http://uww.cosy.sbg.as.at/rec/guestboock`.
- **Японсько-англійський словник**. Складна CGI-програма, що одержує англійське слово в якості запиту від користувача і повертає віртуальний документ із графічним зображенням еквівалентного японського слова, або навпаки. Адреса: http://enterprise.ic.gc.ca/cgi-bin/j-e

Хоча більшість таких документів — іграшки, вони ілюструють потужні можливості CGI. Шлюзовий інтерфейс дозволяє створювати високоефективні віртуальні документи з використанням форм та інтерфейсів до інших програм.

5.7.3. Внутрішня сторона CGI

Як же працює цей інтерфейс? Більшість серверів передбачає, що CGI-програми та сценарії розташовані у спеціальному каталозі (звичайно `cgi-bin`), або мають деяке спеціальне розширення файлу, ці параметри задаються при конфігуруванні серверу. Коли користувач звертається по URL, зв'язаному з CGI-програмою, клієнтський браузер надсилає запит серверу. В більшості випадків запит на CGI-програму такий же, як і для всіх інших Web-документів. Різниця в тому, що, коли сервер визначає, що необхідна адреса є CGI-програмою, він не повертає користувачу файл. Замість цього, сервер пробує виконати програму. Типовий запит клієнта може мати наступний вигляд:

```
GET /cgi-bin/welcome.pl HTTP/1.0
Accept: www/source
Accept: text/html
Accept: image/gif
User-Agent: Lynx/2.4 libwww/2.14
From: if@univ.kiev.ua.
```

Цей GET-запит визначає файл як `/cgi-bin/welcome.pl`. Оскільки сервер сконфігурований на те, щоб визнавати усі файли в дереві підкаталогів `cgi-bin` як CGI-програми, він розуміє, що потрібно виконати програму замість передачі цього

файлу безпосередньо до браузера клієнта. Рядок `HTTP/1.0` визначає використаний протокол зв'язку.

Запит клієнта також вказує формати даних, які може приймати (`www/source`, `text/html` або image/gif), ідентифікує себе як `Lynx` і посилає інформацію про користувача. Вся ця інформація доступна для CGI-програми, поряд із додатковою інформацією від серверу.

Шлях, яким CGI-програми одержують їхні вхідні дані, залежить від серверу і від операційної системи. На UNIX-системі CGI-програми одержують їхні вхідні дані зі стандартного входу (`STDIN`) і від UNIX-змінних середовища. Ці змінні зберігають таку інформацію, як рядок введення форми, формат вхідних даних, довжина вхідних даних (у байтах), ім'я віддаленого комп'ютера і користувача і іншу інформацію клієнта. Вони також зберігають ім'я серверу, протокол зв'язку і назву програмного забезпечення, що керує сервером.

Як тільки CGI-програма починає виконуватись, вона може або створювати новий документ, або видати URL до існуючого. На UNIX програми посилають свої вихідні дані до стандартного виходу (`STDOUT`). Потік даних складається з двох частин. Перша частина є або повним або частковим HTTP-заголовком, що (як мінімум) описує те, в якому форматі видаються вихідні дані (наприклад, HTML, простий текст, GIF, і т. ін.). Порожній рядок показує кінець заголовку.

Друга частина — тіло, що містить дані у форматі, відповідному до вказаного в заголовку. Тіло не змінюється та не інтерпретується сервером за будь-яких умов. CGI-програма може захотіти посилати тільки що створені дані безпосередньо клієнту або через сервер. Якщо вихідні дані складаються з повного HTTP-заголовку, то вони передаються безпосередньо клієнту без модифікації їх сервером. Або, як це робиться в більшості випадків, вихідні дані передаються серверу. В цьому випадку сервер відповідає за додавання повної інформації до заголовку і використання HTTP-протоколу для передачі даних клієнту.

Розглянемо типові вихідні дані програми, що створює віртуальний HTML-документ, з повним HTTP заголовком:

```
HTTP/1.0 200 OK
Date: Monday, 4-May-98 10:21:00 GMT
Server: NCSA/1.4.2
MIME-version: 1.0
Content-type: text/html
Content-length: 2000
<HTML>
  <HEAD><TITLE>Welcome to where time stands still</TITLE>
  </HEAD>
  <BODY><H1>Welcome to my Home Page!</H1></BODY>
</HTML>.
```

Заголовок містить протокол зв'язку, дату і час відповіді, ім'я сервера і версію, а також назву MIME-типу. Найбільш важливі тут — це назва MIME-типу та кількість символів (еквівалентна числу байтів) даних, що знаходяться у тілі повідомлення. Ось вихід з частковим HTTP-заголовком:

```
Content-type: text/html
```

```
<HTML><HEAD><TITLE>Welcome!</TITLE></H<BODY>
<H1>Welcome!</H1> </BODY></HTML>.
```

У цьому випадку присутній єдиний рядок заголовку, в якому вказується тип даних, що слідують далі. Оскільки виходом є документ формату HTML, то у заголовку вказується `text/html`.

Більшість CGI-програмістів віддає перевагу частковому заголовку. Це набагато простіше — видати формат і дані, ніж сформувати повний HTTP-заголовок, що може бути залишено для виконання серверу. Однак, існують моменти, коли необхідно послати інформацію безпосередньо клієнту — тоді потрібно вказати повний HTTP-заголовок.

5.7.4. Програмування в CGI

Які мови програмування можна використовувати для CGI? Відповідь дуже проста: будь-яку, хоча деякі мови більше задовольняють задачам, що вирішує CGI. Перед вибором мови, необхідно розглянути наступні особливості програмування для CGI:

- простота маніпуляцій з текстом;
- здатність використовувати інші бібліотеки програмного забезпечення;
- здатність доступу до змінних середовища (у UNIX).

Розглянемо кожну з цих особливостей. Більшість CGI-програм так чи інакше пов'язана з обробкою тексту. Наприклад, інформація форми звичайно декодується розбиттям рядку по деяких роздільниках.

Здатність мови на взаємодію з іншим програмним забезпеченням, типу баз даних, пошукових систем тощо є також дуже важливою. Це збільшує потужність Web, дозволяючи писати програмні шлюзи до інших інформаційних джерел, наприклад, баз даних або графічних бібліотек.

Нарешті, під останньою особливістю розуміють невимушеність, із якою мова може мати доступ до змінних середовища. Ці змінні складають вхідні дані CGI-програми, і тому дуже важливі.

Деякими з найбільш популярних мов у програмуванні для CGI є AppleScript, C/C++, C Shell, Perl, Tcl і Visual Basic. Розглянемо, наприклад, мову Perl.

Perl (UNIX, Windows, Macintosh)

Perl — широко вживана мова для програмування CGI. Вона має багато потужних засобів і досить нескладна для програміста-новачка. До переваг Perl можна віднести:

- надзвичайно потужні оператори маніпуляції з даними;
- дуже прості та короткі конструкції;
- виклики команд shell виконуються дуже легко, також є еквіваленти для багатьох системних команд UNIX;
- існує багато розширень та бібліотек, побудованих на базі Perl, для виконання специфічних функцій. Наприклад, існують бібліотеки для комунікації з базами даних, бібліотеки для обробки графічних зображень тощо.

Розглянемо приклад програми, що створює віртуальний документ:

```
!/usr/local/bin/perl
print "Content-type: text/plain", "\n\n";
print "Welcome to my Home Page!", "\n";
$remote_host = $ENV{ REMOTE_HOST };
print "You are visiting from ", $remote_host,".";
exit (0);.
```

Перший рядок програми дуже важливий. Він повідомляє серверу про те, що програму необхідно виконати інтерпретатором Perl, розташованим у каталозі `/usr/local/bin`.

Простий оператор `print` використовується, щоб вивести інформацію до стандартного виходу `STDOUT`. Вихідні дані цієї CGI-програми — частковий HTTP-заголовок (лише рядок "`Content-type`"). Цей сценарій робить простий текст, а не HTML, тому вказується `text/plan`.

Два символи нового рядку (`\n`) додаються після заголовку. Це — вимога стандарту HTTP. У залежності від платформи, може знадобитись вивести на вихід два символи повернення каретки та нового рядку у комбінації `\r\n\r\n`.

Перший оператор `print` після заголовку — це вітання. Другий оператор `print` після заголовку показує ім'я віддаленого комп'ютера користувача, що звертається до серверу. Ця інформація відновлена зі змінної середовища `REMOTE_HOST`.

5.7.5. Вхідна інформація CGI-програми

Коли викликається CGI-програма, інформацію, що доступна для неї, можна "розбити" на три групи:

- інформація щодо клієнта, сервера та користувача;
- дані форми, що заповнює користувач;
- додаткова інформація імені шляху.

Більшість інформації щодо клієнта, сервера та користувача розміщено в CGI-змінних середовища. Дані форми або включені до змінної середовища або містяться в тілі запиту. Додаткова інформація імені шляху розміщена також в змінних середовища.

5.7.6. Використання змінних середовища

Більшість необхідної CGI-програмам інформації є доступною для них через змінні середовища операційної системи UNIX. Програми мають доступ до цієї інформації з тією ж легкістю, як і до будь-якої іншої змінної середовища (наприклад, на мові Perl через асоціативний масив `%ENV`). Перелік змінних середовища, доступних для CGI, наведено в табл. 5.2.

Таблиця 5.2

Назва змінної середовища	Опис
`GATEWAY_INTERFACE`	Огляд CGI, що використовує сервер
`SERVER_NAME`	Ім'я або IP-адреса сервера
`SERVER_SOFTWARE`	Назва та версія ПЗ сервера, що відповідає на запит клієнта

Таблиця 5.2. Закінчення

Назва змінної середовища	Опис
SERVER_PROTOCOL	Назва інформаційного протоколу, що використовується
SERVER_PORT	Номер порту, на якому запущений сервер
REQUEST_METHOD	Метод передачі інформації запиту
PATH_INFO	Додаткова інформація про шлях
PATH_TRANSLATED	Трансльована інформація про шлях
SCRIPT_NAME	Віртуальний каталог, з якого запущений сценарій
DOCUMENT_ROOT	Кореневий каталог для Web-документів
QUERY_STRING	Рядок запиту, що передається після "?"
REMOTE_HOST	Назва віддаленої машини
REMOTE_ADDR	IP-адреса віддаленої машини
AUTH_TYPE	Метод визначення імені користувача
REMOTE_USER	Завірене ім'я користувача
REMOTE_IDENT	Користувач, що робить запит
CONTENT_TYPE	MIME-тип запиту
CONTENT_LENGTH	Довжина даних в байтах
HTTP_FROM	E-mail адреса користувача, що робить запит
HTTP_ACCEPT	Список MIME-типів, які клієнт може прийняти
HTTP_USER_AGENT	Назва клієнтського браузера
HTTP_REFERER	URL документу, який користувач проглядав до того, як здійснив запит

Розглянемо просту програму, що показує різну інформацію про сервер за допомогою змінних середовища.

```
#!/usr/local/bin/perl
print "Content-type: text/html", "\n\n";
print "HTML", "\n"
print "<HEAD><TITLE>About this server</TITLE></HEAD>", "\n"
print "<BODY><H1>About this server</H1>", "\n"
print "<HR><PRE>"
print "Server Name: ",$ENV{'SERVER_NAME'}, "<BR>", "\n"
print "Running on port: ", $ENV{'SERVER_PORT' }, "<BR>",
"\n"
print "Server software: ", $ENV{'SERVER_SOFTWARE'}, "<BR>",
"\n"
print "Server protocol: ", $ENV{'SERVER_PROTOCOL'}, "<BR>",
"\n"
print "<HR><PRE>", "\n"
print "</BODY></HTML>", "\n"
exit (0).
```

Ця програма видає інформацію про сервер та наочно показує, як використовується CGI-інформація, що зберігається у змінних середовища UNIX.

За допомогою такої технології можна створити багато аналогічних програм, що видають інформацію про клієнтську частину, користувача, запит, використані прото-

коли та формати передачі даних. Таким же чином вирішується задача обмеження доступу до конкретних документів, що зберігаються на сервері. Наприклад, бажано показувати деякий документ лише користувачам, IP-адреса машин яких знаходиться серед заданого наперед списку. Якщо IP-адреса знайдена у списку, то відображаємо потрібний документ, інакше відображаємо повідомлення.

5.7.7. Доступ до інформації, що введена у форму користувачем

Форми забезпечують отримання вхідної інформації від користувача і направляють її CGI-програмі. Web-браузер дозволяє користувачу вибирати або друкувати інформацію та потім посилає її на сервер при натисканні кнопки **Submit**. Розглянемо, як же CGI-програми отримують доступ до інформації, що була введена користувачем.

Рядки запитів

Перший спосіб послати дані форми до CGI-програми — додати в кінець URL інформацію форми після знаку питання:

```
http://somewhere.com/cgi-bin/prog.pl?param.
```

До знаку питання (?) — звичайний URL. Це — просто CGI-сценарій з назвою `name.pl`. Інформацію після знаку питання називають ***рядком запиту*** (*query string*). Коли до серверу надходить URL із рядком запиту, він викликає CGI-програму, яка ідентифікована шляхом в першій частині URL (до "?") і потім зберігає частину після "?" у змінній середовища `QUERY_STRING`. Далі CGI-програма аналізує отриману інформацію і використовує її за призначенням.

Проста форма

Наступний приклад — більш реалістична ілюстрація того, як форми працюють із CGI. Замість постачання інформації безпосередньо як частини URL, будемо використовувати для передачі даних від користувача форму (в форматі HTML):

```
<HTML>
<HEAD><TITLE>Simple Form! </TITLE></HEAD>
<BODY>
<H1>Simple Form!</H1>
<HR>
<FORM ACTION="/cgi-bin/unix.pl" METHOD="GET">
Command:<INPUT TYPE="text" NAME="command" SIZE=40>
<p>
<INPUT TYPE="submit" VALUE="Submit Form">
<INPUT TYPE="reset" VALUE="Clear Form">
</FORM>
<HR>
</BODY>.
```

Зовнішній вигляд такої форми залежить від браузера, що використовується. Ця форма складається з одного текстового поля, що називається **Command** та двох кнопок. Кнопка **Submit Form** використовується, щоб відправити інформацію з форми до CGI-програми, зазначеної атрибутом ACTION тегу <FORM> мови HTML. Кнопка **Clear Form** очищує інформацію у всіх полях форми.

Атрибут METHOD=GET тегу <FORM> визначає метод передачі даних серверу. GET — один з двох методів, що використовуються для передачі даних із форми, що заповнена користувачем, на сервер. Якщо припустити, що користувач ввів у поле **Command** текст "fortune" і натиснув кнопку **Submit**, браузер передасть на сервер наступний запит:

```
GET /cgi-bin/unix.pl?command=fortune HTTP/1.0.
.
. (інформація заголовку)
.
```

Сервер виконує сценарій unix.pl із каталогу cgi-bin та встановлює значення змінної середовища QUERY_STRING у command=fortune.

Методи GET та POST

У попередньому прикладі, був використаний метод GET, щоб обробити форму. Однак, є інший метод, що можна використовувати з цією ж метою — POST. При використанні методу POST сервер посилає дані від форми як вхідний потік до програми. Тобто, якщо в попередньому прикладі використати метод POST, то на сервер запит був би направлений у наступній формі:

```
POST /cgi-bin/unix.pl HTTP /1.O
.
. (інформація заголовку)
.
Content-length: 15
command=fortune.
```

Таким чином, щоб отримати доступ до інформації, сценарій повинен визначити кількість байтів, що передані, використовуючи змінну середовища CONTENT_LENGTH, і прочитати з стандартного входу точно таку ж кількість байтів. Наприклад на мові Perl це можна зробити наступною послідовністю команд:

```
$size = $ENV{ 'CONTENT_LENGTH' } ;
read(STDIN, $form_info, $size) ;.
```

Оскільки метод, що використовується, вказується при створенні форми, програміст сам потрібен визначати та правильно обробляти кожен з методів. Змінна середовища REQUEST_METHOD містить інформацію про те, який метод використовується.

Кодування даних

Оскільки при використанні методу GET інформація з форми передається через URL, вона не може містити спеціальних символів, які не дозволяються стандартом для URL. Тому використовується деяке спеціальне кодування.

Припустимо, користувач заповнив форму, одне з полів якої містить його дату народження, наприклад, це — рядок вигляду "15/04/77". В закодованому вигляді кожен спеціальний символ замінюється на його шістнадцятковий код після символу "%". Таким чином, наведений вище рядок буде закодований у вигляді "15%2F04%2F77". Весь тягар по перекодуванню лягає на програміста CGI. На різних мовах програмування ця задача може мати різну ступінь складності. Наприклад, на мові Perl, для перекодування достатньо одного рядка:

```
$form_info =~s/%([\dA-Fa-f][\dA-Fa-f])/pack("C",hex($1))/eq;.
```

Тому мова Perl — одна з найпоширеніших мов для програмування CGI.

Розглянемо більш детально можливості CGI, які зв'язані зі створенням інтерактивних інтерфейсів.

В загальному випадку, інтерактивний інтерфейс користувача являє собою систему, що забезпечує взаємодію користувача і програми. Для WWW, інтерактивний інтерфейс можна визначити як послідовність HTML-документів, що реалізують інтерфейс користувача. Можна також умовно кваліфікувати принципи побудови інтерфейсу по типам формування HTML-документа: статичний або динамічний.

У першому випадку джерелом інтерфейсу є HTML-документ, створений у якомунебудь текстовому чи HTML-орієнтованому редакторі. Отже, даний документ залишається незмінним протягом використання. В другому випадку джерелом інтерфейсу є HTML-документ сгенерированньїй CGI-модулем, т.т. з'являється деяка гнучкість у видозміні інтерфейсу під час використання.

Таким чином, можна ввести поняття інтерактивного інтерфейсу для WWW. Інтерактивний інтерфейс для WWW являє собою послідовність статичних чи динамічно формованих HTML-документів, що реалізують інтерфейс користувача.

Практично будь-яка задача, що вирішує проблему одержання даних від клієнта, зв'язана з побудовою інтерфейсу. Найбільш цікавим є побудова інтерфейсів до різних баз даних, доступ до SQL-сервера, одержання інформації від периферійних пристроїв, створення клієнтських робочих місць.

Задача побудови вищезгаданих інтерфейсів поділяється на дві частини: клієнтську і серверну. Для створення клієнтської частини необхідно створити HTML-документ, в якому реалізований інтерфейс із користувачем. У мові HTML це можливо за допомогою форм. Серверна частина складається з модуля, що виконується, основних задач обробки даних, що надходять від клієнтської частини, формування відповіді у форматі HTML і т. ін. Такий модуль називається ***CGI-модулем***.

5.7.8. Заголовок вихідного потоку

В деяких випадках необхідно уникати обробки сервером результату CGI-модуля і посилати клієнту дані без змін. Для відмінності таких CGI-модулів, CGI вимагає, щоб їх імена починалися на `nph-`. У цьому випадку формування синтаксично правильної відповіді клієнту CGI-модуль бере на себе.

Заголовки із синтаксичним розбором

Результат CGI-модуля повинен починатись з заголовку, що містить визначені рядки, і завершуватись двома символами CR (0x10). Будь-які рядки не є директивами сервера. На даний момент, CGI-специфікація визначає три директиви сервера: `Contetn-type`, `Location` і `Status`.

Content-type

Визначає MIME чи тип документа, що повертається.

Наприклад: `Content-type: text/html <CR><CR>` повідомляє серверу, що наступні за цим повідомленням дані — це документ у форматі HTML.

Location

Вказує серверу, що повертається не сам документ, а посилання на нього. Якщо аргументом є URL, то сервер передасть вказівку клієнту на перенаправлення запиту. Якщо аргумент являє собою віртуальний шлях, то сервер поверне клієнту заданий цим шляхом документ, так якби клієнт запитував цей документ безпосередньо.

Наприклад: `Location: http://host/file.txt` приведе до того, що Web-сервер видасть `file.txt`, так, якби він був викликаний клієнтом. Якщо CGI-модуль повертає посилання на Gopher-сервер (наприклад на `gopher://gopher.ncsa.uiuc.edu/`), то результат буде наступний:

```
Location: gopher://gopher.ncsa.uiuc.edu/.
```

Status

Задає серверу HTTP/1.0 рядок-статус, що буде відправлений клієнту у форматі: `nnn xxxxx`, де: `nnn` — код статусу з трьох цифр, `xxxxx` — рядок причини.

5.7.9. Стандартний вхідний потік

При використанні методу запиту POST дані передаються як вміст HTTP-запиту, і будуть послані в стандартний вхідний потік.

Дані передаються CGI-модулю в наступній формі:

```
name=value&name1=value1&... &nameN=valueN,
```

де `name` — ім'я змінної, `value` — значення змінної, `N` — кількість змінних.

На файловий дескриптор стандартного потоку вводу посилається байт `CONTENT_LENGTH`. Так само сервер передає CGI-модулю `CONTENT_TYPE` (тип даних). Сервер не посилає символ кінця файлу після передачі `CONTENT_LENGTH` байт даних чи після того, як CGI-модуль їх прочитає.

Змінні оточення `CONTENT_LENGTH` і `CONTENT_TYPE` встановлюються в той момент, коли сервер виконує CGI-модуль. Таким чином, якщо в результаті виконання форми з аргументом тега `<FORM> METHOD="POST"` сформований рядок даних `firm=MMM&price=100023`, то сервер встановить значення `CONTENT_LENGTH` рівним 21 і `CONTENT_TYPE` у `application/x-www-form-urlencoded`, а в стандартний потік вводу посилається блок даних.

У випадку методу GET, рядок даних передається як частина URL. Наприклад, `http://host/cgi-bin/script?name1=value1&name2=value2`. У цьому випадку змінна оточення `QUERY_STRING` приймає значення `name1=value1&name2=value2`.

Аргументи командного рядка

CGI-модуль у командному рядку від сервера отримує:

- залишок URL після імені CGI-модуля як перший параметр (перший параметр буде порожній, якщо було присутнє тільки ім'я CGI-модуля);
- список ключових слів як залишок командного рядка для сценарію пошуку, чи імена чергових полів, форми з доданим знаком рівності і відповідних значень змінних.

Ключові слова, імена і значення полів форми передаються декодованими (з HTTP URL-формату кодування) і перекодованими відповідно до правил кодування Bourne shell так, що CGI-модуль у командному рядку одержить інформацію без необхідності здійснювати додаткові перетворення.

Послідовність дій для обробки вхідних даних CGI-модуля

Виходячи з різниці методів запитів GET і POST, можна визначити послідовність дій для обробки вхідних даних CGI-модуля для різних типів запитів.

Для методу GET:

1. Одержати значення змінної `QUERY_STRING`.
2. Декодувати імена і їхні значення (з огляду на те, що всі прогалини при декодуванні сервером були замінені символом "+", і всі символи з десятковим кодом більше 128 перетворені в символ "%" і наступним за ним шістнадцятковим кодом символу).
3. Сформувати структуру відповідності "ім'я-значення" для подальшого використання в CGI-модулі

Для методу POST:

1. Одержати зі стандартного вхідного потоку `CONTENT_LENGTH` символів.
2. Декодувати імена і їх значення (з огляду на те, что всі прогалини при декодуванні сервером були замінені символом "+" і всі символи з десятковим кодом більше 128 перетворені в символ "%" і наступним за ним шістнадцятковим кодом символу).
3. Сформувати структуру відповідності "ім'я-значення" для подальшого використання в CGI-модулі.

Вочевидь, що відмінність тільки в джерелі даних. Тому, в принципі, можливе створення єдиного модуля для методів POST і GET. Необхідно тільки додати в початок перевірку значення змінної `REQUEST_METHOD` для визначення методу запиту.

Після формування структури "ім'я-значення" можна приступити до рішення задач, заради яких, власне, створювався CGI-модуль. Зрозуміло, що задачі, які вирішуються CGI-модулем, можуть бути дуже різноманітними (одержання й обробка пошти, доступ до баз даних, гостьова книга і т.ін.).

Наступним важливим моментом є динамічне формування CGI-модулем HTML-документа (оформлення результату роботи модуля). Наприклад, таблиці вибірки з бази даних. Для цього CGI-модуль повинен видати в стандартний вихідний потік заголовок, що складається з рядка `Content-type: text/html` і порожнього рядка

(двох символів CR). Після цього заголовка можна давати будь-який текст у форматі HTML.

5.7.10. Приклади CGI-модулів

В якості прикладу, розглянемо роботу тестових програм, що поставляються разом із ПЗ сервера HTTP стандарту NCSA.

Для тестування роботи форм поставляються програми:

- **post-query** — для тестування роботи форм із методом запиту POST;
- **query** — для тестування роботи форм із методом запиту GET;
- **util.c** — опис функцій для обробки вхідного потоку (використовується `query` і `post-query`).

Розглянемо простий приклад форми мовою HTML і програму, що використовує `query`:

```
<HTML>
<HEAD>
<TITLE>Приклад використання CGI</TITLE>
</HEAD>
<BODY>
<FORM ACTION="http://iceman.cnit.nsu.ru/cgi-bin/post-query"
      METHOD="POST">
<B>Введіть своє ім'я<I>(Прізвище Ім'я По батькові)</I>:</B>
<INPUT name=RealName type=text size=40 maxlength=60
       value="Петров Іван Сидорович"><P>
Стать: <INPUT name=Sex type=Radio
              value="Чоловічий" CHECKED>- чоловічий
<INPUT name=Sex type=Radio value="Жіночий">-жіночий<P>
<INPUT name=Submit type=submit value="Надіслати запит"><BR>
<INPUT name=Reset type=reset value="Очистити">
</FORM></BODY></HTML>.
```

Після ініціації форми шляхом натискання кнопки **Надіслати запит** Web-сервер обробляє потік даних від форми (заміняє всі прогалини в іменах і значеннях на символ "+", заміняє всі символи з десятковим кодом великим 128 на символ "%" і наступним за ним шестнадцятковим кодом символу (наприклад "И" у `%38`))

Вихідний потік буде мати наступний вигляд:

```
RealName=0/oCF%E5%F2%FO%EE%E2+%C8%E2%EO%ED+%Dl%E8%E4%EE%F0
%EE%E2%E8%F7&Sex=%CC%F3%E6%Fl%EA%EE%E9&Submit=%CF%EE%F1%EB
%EO%F2%FC+%E7%EO%EF%FO%EE%F1.
```

У момент передачі керування модулю `post-query` сервер привласнює значення змінних оточення й аргументам командного рядка:

```
argc = 0. argv =
SERVER_SOFTWARE = NCSA/1.5.1
SERVER_NAME = iceman.cnit.nsu.ru
GATEWAY_INTERFACE = CGI/1.1
SERVER_PROTOCOL = HTTP/1.0
```

```
SERVER_PORT = 80
REQUEST_METHOD = POST
HTTP_ACCEPT = image/gif, image/x-xbitmap,
              image/jpeg, image/pjpeg,*/*
PATH_INFO =
PATH_TRANSLATED =
SCRIPT_NAME = /cgi-bin/test-cgi
QUERY_STRING =
REMOTE_HOST = fwa.cnit.nsu.ru
REMOTE_ADDR= 193.124.209.74
REMOTE_USER =
AUTH_TYPE =
CONTENT_TYPE = application/x-www-form-urlencoded
CONTENT_LENGTH = 142.
```

Результат роботи `post-query`:

```
<H1>Query Results</H1>
You submitted the following name/value pairs:<p>
<ul><li><code>RealName = Петров Іван Сидорович</code>
    <li><code>Sex = Чоловічий</code>
    <li><code>Submit = Надіслати запит </code></ul>.
```

І на екрані браузера:

```
Query Results
You submitted the following name/value pairs:
RealName = Петров Іван Сидорович
Sex = Чоловічий
Submit = Надіслати запит.
```

Нижче наведений вихідний текст програми `post-query` на мові С:

```
#include <stdio.h>
#ifndef NO_STDLIB_H
#include <stdlib.h>
#elsechar *getenv();
#endif
#defme MAX_ENTRTES 10000

typedef struct {
char *name;
char *val;
} entry;

char *makeword (char *line, char stop);
char *finakeword (FILE *f, char stop, int *len);
char x2c (char *what);
void unescape_url (char *url);
void plustospace (char *str);
```

```
main (int argc, char *argv[])
{
entry entries [MAX_ENTRIES];
register int x,m=0;
int cl;
printf ("Content-type: text/html%c%c", 10, 10);
if (strcmp(getenv("REQUEST_METHOD"),"POST"))
{ printf ("This script should be referenced
          with a METHOD of POST.\n");
printf ("If you don't understand this, see this ");
printf ("<A HREF=\"http://www.ncsa. uiuc.edu/SDG/Software/
         Mosaic/ Docs/ffil-out-forms/overview.html\">
forms overview</A>.%c", 10);
exit (1);
}
if (strcmp(getenv("CONTENT_TYPE"),
    "application/x-www-form-urlencoded"))
{printf ("This script can only be used to
         decode form results. \n"); exit(1);  }
cl = atoi (getenv ("CONTENT_LENGTH"));
for (x=0; cl && (!feof (stdin)); x++)
{m=x; entries [x].val = fmakeword (stdin,'&',&cl);
plustospace (entries[x].val);
unescape_url (entries[x]. val);
entries [x].name = makeword (entries[x].val, '=');  }
printf ("<H1>Query Results</H1>");
printf ("You submitted the following
        name/value pairs:<p>%c", 10);
printf ("<ul>%c",10);
for (x=0; x <= m; x++)
printf ("<H> <code>%s = %s</code>%c",
        entries [x] .name, entries [x]. val, 10);
prmtfr </ul>%c",10);  }.
```

Треба відзначити, що `post-query` не обробляє імена, тому в прикладі вони дані англійською мовою. Якщо треба використати російські або українські назви імен, то слід обробити імена також, як і значення, тобто замінити всі символи "+" на прогалини і перетворити шістнадцяткові коди кириличних символів у сам символ. Приведемо також вихідний текст функцій, використаних `post-query`.

```
char *makeword (char *line, char stop) {
/* Призначена для виділення частини рядка,
   обмеженої "стоп-символами" */
int x = 0,y;
char *word = (char *) malloc(sizeof (char) *
             (strlen (line) + 1));
for (x=0;((line[x]) && (line [x] != stop)); x++)
```

```
word [x] = linefx];
word [x] = '\0';
if (line [x]) -H-x;
y=0;
while (line[y++] return word;  }
char *fmakeword (FILE *f, char stop, int *cl) {
/* Призначена для виділення рядка, обмеженої
 "стоп-символом" stop, з потоку f довжиною cl.*/
int wsize;
char *word;
int ll;
wsize = 102400;
ll = 0;
word = (char *) malloc (sizeof (char) * (wsize + 1));
while(1)  {
word [ll] = (char)fgetc (f);
If (ll==wsize) { word [ll+1] = '\0';
wsize+=102400;
word = (char *)realloc(word,sizeof(char)*(wsize+1));  }
-- (*cl);
if (word[ll] stop) || (feof(f)) || (!(*cl))) {
if (word[ll] != stop) ll++;
word [ll] = -\0';
return word;  } ++ll; } }
char x2c (char *what) {
/* Призначена для перетворення шістнадцяткового
  коду символу в код символу */
register char digit;
digit = (whatf [0] >= 'A' ?
       ((what[0] & Oxdf) - 'A')+10 : (what[0] - '0'));
digit *= 16;
digit += (what[1] >= 'A' ?
        ((whatf [1] & Oxdf) - 'A')+10 : (what [1] - '0'));
return(digit);  }
void unescape_url (char *url) { register int x,y;
for (x=0, y=0; url [y]; ++x,++y) {
if ((url [x] = url [y]) ='%')
{ url [x] =x2c (&url [y+1]);
y+=2; }  }
url[x] = '\0';  }
void plustospace(char *str) {
/*заміна символів "+" на символ "прогалина"*/
register int x;
for(x=0;str[x];x++) if(str[x] == '+') str[x]  }.
```

Для демонстрації реалізації форми з методом запиту GET скористаємося тією ж самою формою, що і для методу POST, і програмою `query`. Для цього змінимо значення атрибутів `ACTION` і `METHOD` у тезі `<FORM>`:

```
FORM action="http://iceman.cnit.nsu.ru/cgi-bin/query" METHOD=GET>
```

Після ініціації форми сервер установить наступні значення для змінних оточення й аргументів командного рядка:

```
argc = 0. argv is =
SERVER_SOFTWARE = NCSA/1.5.1
SERVER_NAME = iceman.cnit.nsu.ru
GATEWAY_INTERFACE = CGI/1.1
SERVER_PROTOCOL = HTTP/1.0
SERVER_PORT = 80
REQUEST_METHOD = GET
HTTP_ACCEPT = image/gif, image/x-xbitmap,
              image/jpeg, image/pjpeg, */*
PATH_INFO =
PATH TRANSLATED =
SCRIPT_NAME = /cgi-bin/test-cgi
QUERY_STRING =
RealName=%CF%E5%F2%F0%EE%E2+%C8%E2%E0%ED+%Dl%E8%E4%EE%F0%EE
%E2%E8%F7&Sex=%CC%F3%E6?/oFl%EA%EE%E9&Submit=%CF%EE%Fl%EB%E0
%F2%FC+%E7%E0%EF%F0%EE%Fl
REMOTE_HOST = fwa.cnit.nsu.ru
REMOTE_ADDR= 193.124.209.74
REMOTE_USER =
AUTH_TYPE =
CONTENT_TYPE =
CONTENT_LENGTH =.
```

Як бачимо, вихідний потік від форми з'явився в значенні змінної `QUERY_STRING`. Результат роботи `query` цілком збігається з результатом роботи `post-query`.

5.7.11. Конструкції мови HTML для побудови форм

Тег <FORM>

Призначений для одержання інформації від клієнта і визначає початок і кінець форми.

Атрибути:

- **ACTION** — визначає URI CGI-сценарію;
- **METHOD** — визначає метод передачі інформації сценарію; можливі значення `GET` чи `POST`;
- **ENCTYPE** — визначає тип MIME декодування інформації (значення цього атрибута за замовчуванням — `application/x-www-form-urlencoded`);
- **SCRIPT** — використовується для передачі URI сценарію. Мова сценарію й інтерфейс користувача при цьому не є частиною специфікації HTML 3.0.

Тег <INPUT>

Призначений для створення різних по своїй функціональності полів вводу.
Атрибути:

- **TYPE** — визначає тип поля форми. Припустимі значення:
  - TEXT — дозволяє символьне введення;
  - PASSWORD — призначений для "схованого" введення символів (символи, що вводяться, не відображаються);
  - CHECKBOX — поле, що дозволяє два стани ("так", "ні"). Повинно застосовуватися з атрибутами NAME і VALUE;
  - RADIO — поле, що дозволяє вибір "один із всіх";
  - SUBMIT — кнопка, що ініціює передачу інформації з форми сценарію обробки, визначеному в ACTION відповідно до методу, визначеному атрибутом METHOD;
  - RESET — кнопка, що скидає усі введені раніше значення;
  - IMAGE — поле що дозволяє відтворити подію SUBMIT за допомогою зображення, при цьому повертається два значення: name.x = координатаX та name.y = координата Y, де X і Y — координати положення покажчика миші на зображенні в момент натискання кнопки миші;
  - HIDDEN — поле значення, що не відображається;
  - RANGE — визначає поле, що дозволяє ввести цифрове значення з визначеними припустимими верхньою і нижньою межами; використовується разом з атрибутами MAX і MIN, що визначають область припустимих значень (наприклад: TYPE=RANGE MIN=1 MAX=10);
- **NAME** — значення цього атрибута визначає ідентифікатор поля;
- **VALUE** — значення цього атрибута визначає, що буде передано в якості значення по замовчуванню для даного поля при ініціалізації форми;
- **SRC** — визначає URI файлу зображення. Використовується тільки з типом поля IMAGE;
- **SIZE** — визначає розмір поля;
- **CHECKED** — дозволяє установити початкове значення поля типу CHECKBOX;
- **MAXLENGTH** — визначає максимальну кількість символів, припустимої для ведення в поле;
- **ALIGN** — дозволяє позиціонування; припустимі значення:
  - TOP — вертикальне вирівнювання по верхньому краю (тільки з TYPE=IMAGE);
  - MIDDLE — вертикальне вирівнювання по середині (тільки з TYPE=IMAGE);
  - BOTTOM — вертикальне вирівнювання по нижньому краю (тільки з TYPE=IMAGE).
  - LEFT — горизонтальне вирівнювання по лівому краю;
  - RIGHT — вирівнювання праворуч;
- **DISABLED** — визначає поле як "read only" — тільки для читання; значення в полі не може бути змінено користувачем;
- **ERROR** — визначає повідомлення про помилку, що пояснює, чому введене значення в поле невірне.

Тег <TEXTAREA>

Призначений для визначення області вводу тексту. Розмір поля визначається атрибутами:

- **NAME** — значення цього атрибута визначає ідентифікатор поля; повертається при ініціації форми;
- **ROWS** — визначає кількість рядків у текстовій області;
- **COLS** — визначає кількість стовпчиків у текстовій області;
- **VALUE** — задає значення за замовчуванням;
- **DISABLED** — визначає поле як "read only" — тільки для читання; значення в полі не може бути змінено користувачем;
- **ERROR** — визначає повідомлення про помилку, що пояснює, чому введене значення в полі невірне.

Тег <SELECT>

Призначений для визначення області вибору з декількох значень (меню). Синтаксис:

<SELECT атрибути><OPTION>значення ... <OPTION> значення</SELECT>.

Атрибути:

- **NAME** — значення цього атрибута визначає ідентифікатор поля; повертається при ініціації форми;
- **SIZE** — визначає кількість видимих можливих значень;
- **MULTIPLE** — визначає можливість множинного вибору;
- **DISABLED** — визначає меню как "read only" — тільки для читання; значення в меню не може бути обрано користувачем і відображається сірим кольором.

Тег <OPTION>

Використовується тільки в тезі <SELECT> для визначення пунктів меню. Атрибути:

- **SELECTED** — визначає значення за замовчуванням;
- **VALUE** - визначає значення, що повертається.

Розділ 6

Корпоративна мережа Intranet

"Усі слова уже були чиїмось"
(Народна мудрість)

6.1. Структура мережі Intranet

Історія Intranet почалася восени 1994 року, коли Стів Теллін (компанія Amdahl) запропонував термін ***Intranet*** для визначення корпоративних інформаційних систем, побудованих на принципах Internet.

Intranet — це територіально обмежена локально-обчислювана мережа (LAN), в якій застосовується технологія побудови і служби представлення інформації Internet. Тому базовим протоколом управління передачі інформації є TCP/IP. Для повноцінного функціонування Intranet необхідна підтримка наступних протоколів:

- HTTP — протокол передачі гіпертекстів;
- SMPT, POP3 — протоколи електронної пошти;
- FTP — протокол передачі файлів;
- NNTP — протокол групи новин Usenet.

Якщо немає можливості перейти на TCP/IP, то можна використовувати спеціальні шлюзи TCP/IP компаній Firelor чи Perfomance Technologies. Шлюзи перетворюють формати пакетів між протоколами TCP/IP і IPX (міжмережевий обмін пакетами) фірми Novell або NetBIOS, спрощуючи проблему переносу TCP/IP в корпоративну LAN.

Intranet дозволяє значно підвищити ефективність праці за рахунок сумісного використання інформаційних ресурсів всередині компанії (фірми, корпорації тощо) і за рахунок використання технологічних концепцій Internet (Web-технології).

В Intranet задачі зв`язку, виробничі задачі і процеси (наприклад, робочі взаємовідношення, інфраструктури, проекти, бюджет і графіки) визначаються в режимі "on-line" через єдиний інтерфейс. Таким чином, Intranet —це інтелектуальний потенціал організації, де кожний окремий комп`ютер використовується з максимальною результативністю, мінімальними грошовими і трудовими видатками та оперативно.

Відомо, що традиційна архітектура корпоративної мережі, яка існувала до 1994 року, базувалася на сукупності сегментів LAN, пов'язаних між собою через маршрутизатор (рис. 6.1). Продуктивність мережі у такому випадку обмежується пропускною здатністю мережі і продуктивністю магістрального маршрутизатора.

В Intranet ситуація змінюється на інше. Архітектура Intranet представлена на рис. 6.2. Розглянемо деякі особливості такої архітектури.

- **Зміна потоків трафіку**. В традиційній моделі мережі "клієнт-сервер" 80% приходилось на обмін даними з локальним сервером і 20% —на міжмережевий обмін даними. В моделі "клієнт-сервер" на основі Intranet вже не діє старе правило "80/20", тому що неможливо передбачити звертання

користувачів до Web-серверів, які розподілені по всій організації. Це потребує більш якісного планування Intranet.

- **Зміна інтенсивності трафіку**. Розповсюдження мережеорієнтованих, заснованих на Web, програм викликає експоненціальне збільшення мережевого трафіку, що вимагає підвищення вимог до швидкості і надійності пристроїв у мережі.
- **Зміна структури трафіку**. Зростання кількості критично важливих програм, що використовують мультимедія-інформацію, на основі Intranet збільшує потребу в появі служби якості обслуговування, підтримки групових програм з інтенсивним трафіком тощо.

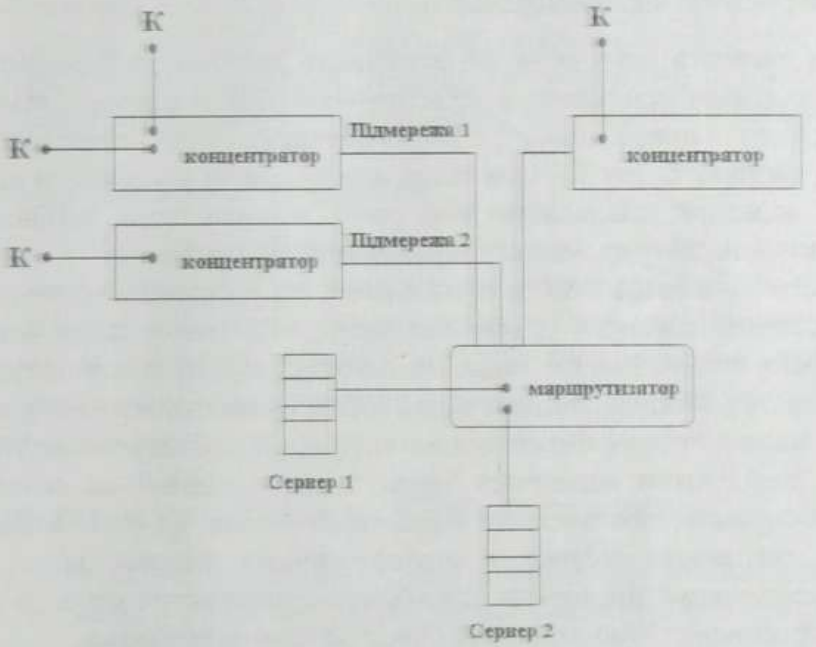


Рис. 6.1. Традиційна архітектура корпоративної мережі (К — ком'ютер)

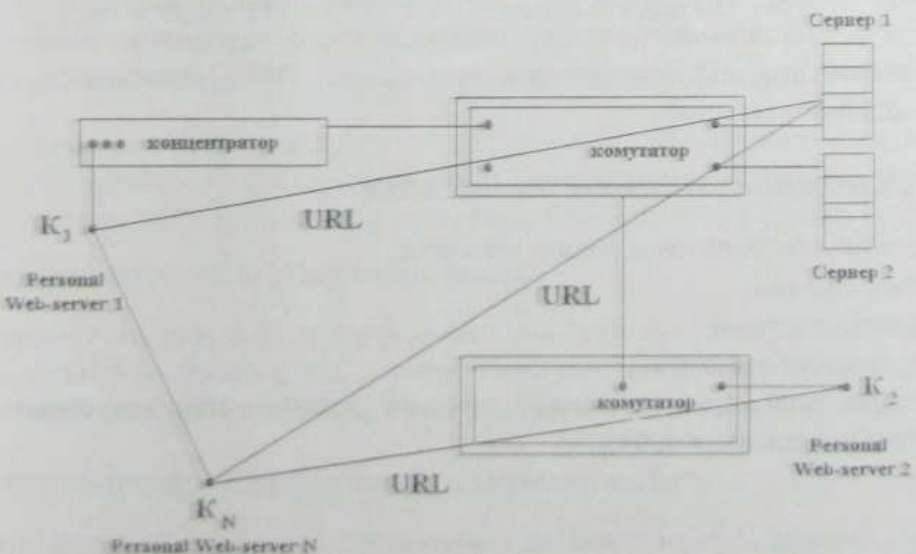


Рис. 6.2. Потоки трафіку в Intranet

Тобто, з впровадженням технології Intranet в корпоративній мережі все більш актуальною, як бачимо, стає потреба в зміні архітектури існуючих корпоративних чи локальних мереж.

Корпоративні мережі часто використовуються для наступних цілей:

- для підтримки оперативних (кожного дня) ділових функцій, наприклад, ведення складського обліку, діловодство тощо;
- для представлення співробітникам компанії доступу до корпоративних документів, баз даних, для друку необхідної інформації;
- для ефективної підтримки внутрішньокорпоративних інформаційних служб;
- для органічної вбудови внутрішньої інформаційної структури в загальну мережу Internet (при необхідності).

Головна перевага Intranet — це можливість доступе до будь-якої інформації, прикладної програми, програмного забезпечення (ПЗ) у одному і тому ж "вікні" за рахунок єдиного універсального інструментарію. Це технології WAN/LAN, Client/Server, систем PC, UNIX, Apple тощо, але на єдиній концепції та пинципах.

Intranet дозволяє створювати Web-сайти відділів, груп фахівців і окремих спеціалістів для інтелектуального корпоративного спілкування.

Відмінність Intranet від LAN полягає в тому, що корпоративна мережа дозволяє вирішити проблему сумісності різних технологій і ПЗ різних фірм (наприклад, HP, IBM, SUN, Apple, Novell, Banyan тощо, які використовуються для побудови LAN) на рівні одного об`єкту. Модель Intranet являє собою універсальну надбудову над LAN.

Сучасні мережі Intranet базуються на внутрішніх корпоративних Web-серверах, які доступні працівникам корпорації через будь-які мережі на основі протокола TCP/IP. Web-сервери дають різну інформацію через єдиний інтерфейс (Web-браузер) за допомогою зв`язків з корпоративними базами даних, файловими серверами і сховищами документів. Web-браузер використовується для доступу до множини корпоративних Web-сторінок з гіпертекстовими зв`язками.

Перевага Intranet від платформи групових технологій (groupware) (наприклад Lotus Notes, Novell Groupwise) — це простота, менша вартість, відсутність необхідності у спеціальній підготовці обслуговуючого персоналу, універсальність (тобто незалежність від конкретного розробника ПЗ), незалежність від ОС, відкритість архітектури тощо.

6.1.1. Основні компоненти Intranet

Основними компонентами мережі Intranet є:

- сервер системи;
- навігатор системи;
- гіпертекстові редактори;
- інструментарії для організації сумісної роботи, обслуговування архівів, організації документообігу.

Сервер системи

Використовується для розподілу ресурсів інформаційної системи. Він зчитує необхідні файли з дисків, запускає програми і передає клієнтським програмам (навігаторам) гіпертекстовий документ. Для цього сервер системи використовує URL.

Для зв`язку системи з базою даних сервер використовує спеціальну програму перетворення формату бази даних в формати мови HTML за допомогою шлюзового інтерфейсу Web — CGI (розглядається в попередньому розділі).

Крім універсального серверу використовуються спеціалізовані сервери. Так, наприклад, всі основні виробники баз даних випустили свої власні Web-сервери, які можуть без CGI звертатись до бази даних. Такі сервери ефективніше використовують можливості обладнання.

Сервер системи має ще одну важливу функціональну особливість: він слідкує за правом (паролем) доступа до документів, тобто забезпечує простий і надійний контроль за діями користувачів, що підвищує безпеку і надійність Intranet.

Навігатор системи чи браузер

Підтримує інтерфейс користувача з системою. Навігатор отримує з різних серверів документи з графікою, які представлені у форматі HTML, і видає їх на екран чи принтер. Навігатор дозволяє використовувати різні протоколи для зв`язку з серверами, наприклад, HTTP, FTP, NNTP, SMTP. За допомогою навігатора просто посилати повідомлення ЕП, запускати програми для перегляду визначених документів різних форматів.

Гіпертекстові редактори чи HTML-редактори

Використовуються для створення нових документів. Вважається, що сервер, навігатор і редактор створюють ядро Web-технології, без якого неможливо побудувати справжню Intranet.

Інструментарій узгодження Intranet з вже існуючим ПЗ

Інструментарій для організації спільної (сумісної) роботи

Це програми, які дозволяють створювати умови для сумісної роботи службовців, організовувати дискусії тощо, комфортно і візуально через сервер системи. Наприклад, Web Crossing, Workgroup Web Forum, netThread тощо.

Інструментарій обслуговування архівів

Використовується для пошуку інформації у великих базах даних корпорації (DWH — Data Ware House), для перетворення (при необхідності) форматів архівних документів в HTML, для створення каталогів документів і пошука по ним.

Інструментарій організації документообігу

Використовується для спрощення документообігу об'єкта за рахунок використання стандартних процедур обробки документів, які дозволяють

користувачу заповнювати лише стандартні спеціальні форми, а весь подальший процес оформлення документа буде виконувати програма Intranet. Наприклад, InterNotes Web Publisher, Basis Document Manager, Infobase Web Publisher, Dyna Web тощо.

6.2. Захист корпоративних мереж

6.2.1. Організація керування каналом доступу

Сьогодні багатьом компаніям потрібні послуги, які надає всесвітня комп'ютерна мережа Internet. Однак їх керівників лякає пряме підключення до глобальної мережі, і це цілком вірно. Internet об'єднує велику кількість людей. Фактично це — не просто комп'ютерна мережа, а зовсім нове суспільство, яке називають кіберпростором.

В Internet, як і в іншому суспільстві, можуть зустрітися комп'ютерні хулігани, шпигуни або так звані хакери (комп'ютерні бандити). Часто буває важко встановити особу користувача, який намагається отримати віддалений доступ до комп'ютера.

Різноманітні протоколи, що використовуються в сегментах Internet, можуть бути погано захищені від прослухування або підробки, і тому недоброзичливець може нелегально використовувати ресурси глобальної мережі.

При підключенні до Internet дуже важливим є організація керування каналом доступу. Здійснювати контроль над цим каналом можна різним чином: перетинати тільки небезпечні дії користувачів або підключитися до мережі через спеціальну захисну систему, яка обмежить такі дії. Це дуже важливий вибір ,від якого залежить майбутня безпека всієї корпоративної мережі.

При використанні спеціальної системи, яка буде контролювати канал доступу до зовнішнього світу (так званий захисний екран), з'являється можливість дозволяти лише необхідні для внутрішньої мережі зв'язки, а інші блокувати. Такий захист дозволяє запобігати не тільки нападу на корпоративну мережу, але й його підготовчому етапу — дослідженню мережі. Крім того, інколи достатньо встановити захист тільки на канал доступу — це спрощує проблему, що розглядається.

Багато компаній та обчислюваних центрів дотримуються означених принципів та правил безпеки. Захисний екран — найбільш адекватний інструмент для їх втілення. Його використовують для захисту від зовнішнього нападу, і якщо корпоративна мережа може бути атакована через модем, то екран у змозі ефективно контролювати телефонні зв'язки.

Більш складна екрануюча система блокує доступ зовнішніх користувачів у корпоративну мережу, але дозволяє вільний доступ до Internet для внутрішніх користувачів. В загальному випадку захисний екран можна налаштувати на захист від будь-якого нападу, однак для цього необхідно правильно його запрограмувати, тобто скласти для нього політику безпеки.

6.2.2. Поняття захисного екрану

Екрануюча система включає два механізми: дозволяючої та забороняючої дії. Дозволяюча частина екрана забезпечує зв'язок між корпоративною та відкритими мережами, а забороняюча блокує небезпечні та незаконні дії користувачів.

Дозволяючий механізм складається з програм, що підтримують сервіси Internet: Telnet, ftp,WWW,Gopher та інші. Такі програми, які називаються сервісними агентами, працюють так, щоб виконувати лише визначені, безпечні команди. Можна побудувати екрануючу систему, цілком складену з сервісних агентів. Принцип побудови такого екрану, який називається ***екрануючим шлюзом***, можна сформулювати наступним чином: "заборонено все, крім необхідного".

Забороняючий механізм захищає корпоративну мережу від небезпечних дій користувачів. Для блокування зв'язку використовуються різноманітні критерії фільтрації. Найбільш відомі з них:

- адреса і маска підмережі відправника;
- адреса і маска підмережі одержувача;
- порт відправника;
- порт одержувача;
- сервіс;
- час використання сервісу.

Фільтрація по адресі та масці підмережі відправника необхідна у тому випадку, коли необхідно блокувати зв'язок з означеним комп'ютером або мережею, а по адресі одержувача — якщо потрібно закрити доступ зовнішньому користувачу у деяку заборонену область.

Якщо користувач не повинен працювати з деяким конкретним сервісом, то екран може блокувати пакети, які використовуються цим сервісом і належать даному користувачу. Іноді фільтрують пакети інформації на основі часу їх відправки. Це необхідно, наприклад, для тимчасового допуску клієнтів у корпоративну мережу. Скомбінувавши основні критерії, можна означити достатньо точні правила.

Як правило, захисний екран складається з набору агентів для кількох сервісів (дозволяючий механізм), правил доступу зовнішніх користувачів до цих агентів і механізму блокування всіх інших дій. Крім цих компонентів екрануюча система може мати додаткові блоки, наприклад, для аналізу дій користувачів та вияву можливих нападів або для створення віртуальної корпоративної мережі.

Захисний екран — це послідовність фільтрів, які виконують означені дії з потоком інформації. Фільтри мають різноманітні відомості про дані, що передаються через них, і тому можуть по різному обробляти потоки інформації. Загальними для всіх фільтрів є наступні дії: пропустити потік інформації через фільтр, блокувати його, подати сигнал тривоги або записати означене повідомлення в системний журнал.

Трьома основними типами фільтрів є :

- пакетний фільтр, або екрануючий маршрутизатор, який фільтрує інформаційні пакети;
- транспортний фільтр, який керує сеансами зв'язку;
- екрануючий шлюз, який керує окремими сервісами.

З цих трьох фільтрів можна побудувати мережевий екран довільної конфігурації. Фільтри найкраще встановлювати послідовно, причому чим більше вони відрізняються, тим надійніший їх загальний захист.

Задача ***екрануючого маршрутизатора*** — створити зону умовної статистичної маршрутизації, тобто екрануючий маршрутизатор передає пакети інформації за адресою, встановленою системним адміністратором, а не по вказаній в пакеті. Наприклад, якщо в корпоративній мережі є безпечний поштовий сервер, то екрануючий маршрутизатор весь потік електронної пошти передає на цей сервер. Існують захисні екрани, які цілком складаються з екрануючих маршрутизаторів, однак важко побудувати необхідний "лабіринт" статистичної маршрутизації.

Екрануючі маршрутизатори є простими, прозорими для користувачів і мають високу швидкодію. Екран, який складається з маршрутизаторів, не затримує передачі інформації. Крім того, фільтри цього типу оперують найбільш повною інформацією про топологію мережі і напрямки передачі інформації.

Недолік цього екрана в тому, що маршрутизатор не враховує кількість інформації, що передається, контекст пакетів та інші аналогічні критерії. Крім того, якщо хакер зламав такий фільтр, то визначити це важко.

Транспортний фільтр керує сеансами зв'язку. Фільтр цього типу під час відкриття інформаційного каналу та обміну повідомленнями може загубити відповідність адрес, записувати контрольну інформацію в системний журнал, контролювати кількість інформації, що передається, і виконувати інші дії транспортного рівня.

Такий фільтр збирає інформацію, котра може попередити можливі напади або знайти помилки керування та конфігурації екранів. Він відслідковує, наприклад, підробку адрес та перевіряє ім'я комп'ютерів, блокує у випадку небезпеки виклик відповідної служби.

Неможливість управління UDP-сеансами — основний недолік транспортного фільтру. Також він не може визначити "законність" сполучень, які контролює.

Екрануючий шлюз забезпечує безпечне обслуговування мережевих служб. Він проводить аунтентифікацію користувачів і тільки після цього дозволяє або заперечує їм зв'язок з внутрішньою мережею. Шлюз відкриває два різних сеанси зв'язку з користувачами різних мереж, тому зовнішнім користувачам здається, що вони встановили зв'язок з самим шлюзом. Такий фільтр фактично є представником корпоративної мережі в Internet.

Шлюз має справу вже з користувачами, а не з пакетами або з сеансами. Тому він містить більш повні дані і реалізує захисний механізм, недоступні для інших фільтрів, наприклад, перевіряє істинність кожного користувача, приховує адреси внутрішніх комп'ютерів та виконує інші аналогічні дії. Системний журнал, який ведеться цим фільтром, найбільш місткий та корисний. Екрануючий шлюз — самий надійний захисний механізм корпоративної системи.

Недоліки цього фільтра: його складність, вузька спеціалізація та невелика швидкість передачі інформації. Необхідність установки нових агентів для кожного нового сервіса зменшує гнучкість системи.

Згадані фільтри — основа сучасної технології екрануючих систем. Однак є змішані екрани, які неможливо віднести до жодного з типів. Ці екрани об'єднують всі види типів в одній системі. Змішані екрани мають повну інформацію про встановлений зв'язок, на підставі якої вони більш точніше можуть визначати можливий напад.

Розглянемо приклади для кожного з типів фільтрації та можливості фільтрації маршрутизаторів. Проаналізуємо транспортний фільтр — програму TCP Wrapper, яка контролює TCP-сеанси, дозволяючи встановити пастки на деякі небезпечні команди, а також про набір сервісних агентів TIS Toolkit, розроблених фірмою Trusted Information System, який є прикладом екрануючого шлюзу. В якості прикладу екрану змішаного типу розглянемо комерційний продукт FIREWALL-1.

6.2.3. Маршрутизатори

Фільтрація пакетів, що обробляються, по означеним полям протоколів дозволяє маршрутизаторам використовувати до потоку інформації декотрі правила обмеженості доступу.

Фільтр задається шаблоном, який містить правила фільтрації. Вони складаються з трьох компонентів: поля, діапазону та дії.

Поле фільтрації — це частина заголовка, яка перевіряється для кожного вхідного пакета. Наприклад, якщо для фільтрації на канальному рівні використовується діапазон значень MAC-адреси відправника, то поле MAC-адреси буде перевірятися у кожного отриманого пакета.

Поля різні для різних протоколів і мають один або декілька діапазонів дозволених значень. Для деяких протоколів системний адміністратор може означити свої поля.

Діапазон. Для кожного поля фільтрації повинен бути вказаний принаймні один діапазон значень, який складається з одного або декількох чисел — мінімального та максимального значень. Наприклад, якщо фільтрувати по полю MAC-адреси відправника, то можна встановити значення **0x0000A2000001** як мінімальне та **0x000OA2000003** як максимальне. У всіх вхідних пакетах будуть перевірятись MAC-адреса відправника, і якщо він знаходиться в означеному діапазоні, то до цього пакету буде застосована встановлена у правилі дія.

Дія в свою чергу визначає реакцію маршрутизатора на пакет, який відповідає правилу. Дії ці різні, але дві з них застосовуються до всіх пакетів. Це:

- блокування — встановлює, що пакет, відповідаючий правилу, не буде передаватися далі;
- запис у системний журнал — визначає повідомлення, яке буде записано у системний журнал. Ця дія може бути об'єднана з довільною іншою дією. Однак вона повинна використовуватись тільки у критичних випадках, інакше системний журнал буде дуже швидко заповнюватись повідомленнями і тому стане непотрібним.

Інші дії різноманітні для кожного протокола, який підтримується маршрутизатором. Наприклад, маршрутизатори Bay Networks в змозі фільтрувати наступні протоколи: Bridge (канальний рівень), IP, DECnet, IV, VINES, Source Routing, IPX, XNS, OSI, DLS та інші.

6.2.4. TCP Wrapper

Ідея TCP Wrapper полягає в наступному: запускати сервісні програми тільки після того, як перевірені всі параметри TCP-сеансу. При спробі відкрити сеанс TCP

Wrapper записує ім'я віддалених комп'ютерів, перевіряє їх, а потім передає управління першочерговим сервісним програмам.

Наприклад, якщо зовнішній користувач намагається встановити зв'язок з програмою Telnet, то TCP Wrapper записує в системний журнал ім'я віддаленого комп'ютера, а потім передає управління дійсному Telnet-серверу і далі у зв'язку участі не приймає.

Крім того, у TCP Wrapper є простий механізм контроля доступу. Якщо зв'язок намагаються встановити з забороненого комп'ютера, то TCP Wrapper блокує його.

Іноді необхідно знати, хто і звідки намагається відкрити заборонений сеанс. Для цього поряд з TCP Wrapper використовується спеціальний finger-сервер, який посилає автоматичні запроси на ім'я відправника пакета. TCP Wrapper можна сконфігурувати таким чином, щоб можна було визначити його відповідну реакцію на деякі підозрілі запити.

6.2.5. TIS Toolkit

TIS Toolkit — набір інструментів для створення екрануючого шлюзу. Всі сервісні агенти та інструменти управління доступом використовують єдиний файл конфігурації з однаковим для всіх компонентів синтаксисом. Це дозволяє централізовано налагодити різноманітні інструменти TIS Toolkit.

Правила конфігурації розроблені таким чином, щоб забезпечити просте налагодження всього комплексу інструментів. Файл конфігурації перевіряється зверху вниз і зліва направо. В правилах можуть використовуватись імена комп'ютерів або їх IP-адреси. Рекомендовано користуватись IP-адресами, оскільки хакер може підробити ім'я DNS.

Програми TIS Toolkit можуть управляти процесами запису у системний журнал, електронною поштою, сервісом імен, tfp-, Telnet-, TCP-сеансами тощо.

Системний журнал

Програма Syslogd записує в системний журнал всю необхідну контрольну інформацію про стан комп'ютера та сполучень з ним. Ця програма, що входить до складу TIS Toolkit має одну особливість: при записі повідомлень в системний журнал може запускатись означена програма. Це використовується, наприклад, для повідомлення системному адміністратору про можливу небезпеку. Оскільки Syslogd виконує декотрі команди, то її конфігураційний файл потрібно захищати від незаконної зміни.

Електронна пошта

Для обслуговування електронної пошти TIS Toolkit містить програму Smap, яка реалізує тільки основні функції SMTP-протоколу, обслуговує SMTP-порт комп'ютера і має невеликий розмір. Цей сервісний агент приймає повідомлення, що надходять, та записує їх на диск до спеціального каталогу. Smap має обмежені повноваження і може змінювати тільки каталог, де зберігаються поштові повідомлення.

Інші програми переглядають поштові повідомлення і передають їх реальному sendmail для подальшої обробки. Smap виконує усі команди SMTP, крім безпосеред-

ньо пов'язаних з передачею пошти: HELO, FROM, RCPT, DATA, QUIT. А команди типу VRFY та EXPN передають відправнику шанобливе повідомлення про помилку.

Smap зберігає функціональні можливості sendmail, захищаючи користувача від прямого зв'язку з відправником листа. Поряд із Smap можна використовувати програми, які виявляють в поштових повідомленнях потенціальну небезпеку. Для захисту від атаки за допомогою пошти її необхідно розподіляти так, щоб було можливо обробляти отримані повідомлення.

FTP

Сервісний агент FTP — єдина програма, яка передає дані між двома мережами в інтерактивному режимі. Шлюз дозволяє використовувати FTP лише для перевірених користувачів. Оскільки служба FTP не виконує ніяких активних дій, крім роботи з диском, то для неї можна легко сформувати достатньо безпечну конфігурацію. Але важливо, щоб шлюз FTP мав невеликі повноваження. У файлі конфігурації вказані заборонені та записані в системний журнал команди FTP, а також перелік FTP-серверів, яким дозволений доступ у систему.

Telnet

Агент Telnet — це програма, яка виконує команди протоколу Telnet. Вона перевіряє повноваження користувача до того, як дозволяє йому доступ у внутрішню мережу. Єдиний доступний файл для Telnet — це файл конфігурації, який читається при встановленні зв'язку. Відразу після того, як файл конфігурації прочитаний, користувач має доступ до дозволених для йому каталогів.

Файл конфігурації Telnet-агента дозволяє вказати, в якій системі чи мережі можна використовувати цей сервіс та з якими системами чи мережами дозволяється зв'язок. В разі "мовчання" він конфігурується так, щоб дозволити внутрішнім системам використовувати Telnet в Internet, але не навпаки. Усі сеанси та час їх відкриття записуються в системний журнал.

UDP-сервіси

Для UDP-потоку немає надійних сервісних агентів, тому доцільно блокувати UDP-пакети. Однак багато незамінних UDP-сервісів типа NTP та DNS можна підтримувати через захисний екран, конфігуруючи їх так, щоб вони забезпечували потреби корпоративної мережі.

Програмне забезпечення служби DNS для UNIX працює з базою даних, яка доступна тільки для читання. Але щоб виключити підробку имен, захисний екран не повинен довіряти їй при роботі з важливою інформацією. Програмне забезпечення сервера имен потрібне для роботи поштової системи, і це єдине програмне забезпечення, яке "довіряє" DNS. Назви машин також використовуються в звітах, але завжди разом з адресами мереж.

TCP-доступ та його використання

Існує програмне забезпечення, яке дозволяє системному адміністратору встановити ім'я користувача, який намагається зв'язатися з системою. TIS Toolkit містить

транспортний фільтр Netacl, що забезпечує підтримку для всіх сервісів на основі TCP. Цей фільтр не має загального використання для всього пакета інструментів TIS Toolkit.

Сервер широкого профілю

Plug-gw — багатоцільовий сервісний агент, який контролює два сервіси незалежно. Він призначений для підтримки новин Usenet, але у випадку необхідності може використовуватись і для інших сервісів. Plug-gw конфігурується так, як і інші агенти TIS Toolkit, і працює як простий передавач даних — не звертається до диску і не запускає ніяких додаткових програм. Всі свої дії він записує в системний журнал.

TCP-з'єднання через захисний екран загального призначення повинні використовуватись з великою обережністю, оскільки Plug-wg не перевіряє правильність повідомлень і не аутентифікує користувачів, крім як на основі їх адрес.

Аутентифікація

Аутентифікація — це перевірка дійсності ім'я користувача або комп'ютера. Сервер аутентифікації Authsrv забезпечує перевірку повноважень користувачів. Використовувати його необов'язково — він необхідний лише для агентів FTP та Telnet, встановлених на захисному екрані. Authsrv використовує різноманітні засоби аутентифікації, а адміністратору необхідно тільки вказати алгоритм перевірки повноможень для конкретних користувачів. Це дозволяє організації, яка вже використовує для аутентифікації користувачів спеціальні пристрої, використовувати їх для роботи з захисними екранами. Підтримується декілька алгоритмів аутентифікації — запит/відповідь, системи одноразових або простих паролів тощо.

База даних Authsrv підтримує групову аутентифікацію, тобто дозволяє визначити групу користувачів. Authsrv автоматично зберігає інформацію про те, коли користувач останній раз звертався до серверу аутентифікації і скільки було невдалих спроб. Можна автоматично заперечити або блокувати доступ в систему тим користувачам, які досить часто помиляються. Authsrv записує всі дані в системний журнал. База даних Authsrv повинна бути захищена від зчитування та змін, тому її рекомендують встановлювати на захищеному комп'ютері.

6.2.6. FIREWALL-1

Захисна система FIREWALL-1 (екран змішаного типу) містить два основних компоненти:

- модуль пакетної фільтрації;
- модуль контролю.

Контрольний модуль формує політику безпеки FIREWALL-1. Він програмує і управляє фільтруючими модулями по правилам, заданим системним адміністратором.

Фільтруючий модуль міститься в тих точках корпоративної мережі, де необхідний контроль маршрутизації, наприклад на шлюзовому комп'ютері. Для надійності фільтрації інформаційних пакетів модуль контролює низькорівневі протоколи. Шлюз

пропускає пакет, якщо виконується деяке дозволяюче правило, інакше інформація може бути загубленою.

Фільтруючий модуль має характерні особливості:

- коректно обслуговує такі протоколи, як UDP, RPC та ін.;
- перевіряє цілісність інформаційного пакету;
- для фільтрації використовує поля: сервіс, адресу відправника та одержувача, час відправки пакета;
- в небезпечних випадках подає сигнал небезпеки;
- інформація про пакети, що проходять через модуль, міститься в System Status Monitor, і її можна отримати за допомогою протоколу SNMP;
- інформація про кожне невдале намагання встановити зв'язок через фільтруючий модуль записується у системний журнал (стандартні дані про сеанс — адреси відправника та отримувача інформації, сервіс, протокол, дата та час відправлення пакета, порт відправника, стан зв'язку — підтвердження або відмова, тип сигналу небезпеки або пароль фільтруючого модуля).

Додаткові компоненти та блоки

Контрольна станція управляє діями фільтруючих модулів і передає системному адміністратору інформацію про них, забезпечуючи таким чином перевірку роботи захисної системи. Контрольний модуль включає блок мережевих об'єктів, блок контролю сервісу, блок створення або реалізації правил, переліки дозволеного доступу, блок управління системою, візуалізатор системного журналу.

Блок об'єктів мережі зберігає інформацію про об'єкти даної корпоративної мережі. До таких об'єктів відносяться:

- мережі та підмережі;
- сервери та робочі станції;
- комп'ютери FIREWALL-1 та шлюзи;
- маршрутизатори;
- область Internet.

Параметри кожного об'єкта визначаються його атрибутами (мережева адреса, підмережева маска та ін.) та зберігаються в спеціальній базі даних. Інформацію про комп'ютери, маршрутизатори та шлюзи можна отримати за допомогою SNMP. Об'єкти мережі можуть об'єднуватись в групи, створюючи таким чином ієрархічні структури.

Блок контролю сервісу управляє сервісами і записує виконані дії. Блок підтримує наступні сервіси:

- стандартний ARPA-сервіс: Telnet, FTP, SMTP і т. ін.;
- берклеївський r-сервіс: rlogin, rsh і т. ін.;
- різні протоколи Internet, такі як HTTP, Gopher, Archie та ін.;
- IP-сервіс, Internet Control Message Protocol (ICMP), Routing Internet Protocol (RIP), SNMP і т. ін.

Для того щоб налаштувати модуль на новий сервіс, необхідно визначити, яким низькорівневим протоколом він підтримується, а потім коректно установити атрибути даного сервісу. Тут маються на увазі Transmission Control Protocol (TCP), User Data-

gram Protocol (UDP), Remote Procedure Call (RPC) та інші протоколи, які не відповідають стандартам. Останні легко визначити, використовуючи прості вирази та макроси. Сервіси можуть об'єднуватися у групи, створюючи ієрархічну структуру.

Блок створення та реалізації правил встановлює параметри (комунікаційна пара, стан каналу, стан маршруту та виконувач правила), по яких фільтруючі модулі FIREWALL-1 будуть блокувати повідомлення, що надходять. Комунікаційна пара — адреса відправника та отримувача повідомлення, а також назва сервісу, що використовується. Інформаційний канал може бути або відкритим, коли система FIREWALL-1 "прозора" для пакетів, або закритим, якщо при намаганні зв'язку захист надсилає відмову або просто ігнорує всі пакети.

Інформація про стан шляху може включати в себе наступні пункти:

- непідтверджений пароль;
- сигнал тривоги;
- повідомлення про отримання пошти;
- генерація елементарної пастки;
- виконання програми.

Кожне правило може виконуватись на шлюзах, серверах або комп'ютерах.

При намаганні доступу в систему база правил перевіряється зверху вниз та справа наліво. По замовчуванню останнє правило — блокувати будь-який зв'язок. Таким чином, принцип захисту — "заборонено все, що не дозволено". База правил компілюється в інструкції, які описують об'єкти та правила, що до них застосовуються.

Переліки дозволеного доступу перераховують об'єкти, що доступні для даного користувача, після його аутентифікації. Ці переліки надсилаються кожному маршрутизатору, і після цього користувачі отримують доступ до потрібних об'єктів уже без додаткової аутентифікації.

Блок управління системою демонструє адміністратору екрану інформацію про стан кожного фільтруючого модуля та статистичні зведення про пакети, що проходять через нього. Цю інформацію можна отримати і через SNMP.

Візуалізатор системного журналу дозволяє адміністратору екрану аналізувати системний журнал, в якому міститься вся необхідна інформація про перевірки паролів користувачів, намагання зв'язку, розривах зв'язку, тощо. Для кожного запису системного журналу вказується дата, точний час, відправник і отримувач інформації, сервіс, виконана дія, тип сигналу тривоги, інші дані. Можливо здійснювати пошук іншої інформації. Отриманий компактний текст можна продивитись , зберегти у форматі ASCII чи роздрукувати.

Робота візуалізатора здійснюється у реальному часі, що дозволяє системному адміністратору швидко реагувати на ситуацію, яка трапляється. Контроль FIREWALL-1 також можна проводити за допомогою SNMP.

6.2.7. Конфігурації захисних екранів

Як правило, екрануючі системи будуються із наступних основних елементів:

- екрануючий маршрутизатор;
- комп'ютер-бастіон;
- екрануючий шлюз.

Про екрануючий маршрутизатор та шлюз було сказано вище, а комп'ютер-бастіон — це найбільш захищена обчислювальна машина, яка встановлюється в потенційно небезпечному місці корпоративної мережі.

Ком'ютер-бастіон являє собою екрануючий шлюз з доповнюючими захисними управляючими та аналітичними програмами. Цей комп'ютер — основа комплексного захисту корпоративної мережі.

Мережевий екран — конфігурація екрануючої системи, яка цілком складається з екрануючих маршрутизаторів, що містяться між окремими мережами. Такий екран забезпечує прямий зв'язок між комп'ютерами різних мереж.

На кордонах, що розділяють локальні мережі, звичайно вже є маршрутизатор. Тому для створення мережевого екрану достатньо задіяти його екрануючі механізми. Однак правила фільтрації важко скласти так, щоб вони були простими та ефективними. Крім того, навіть неможливо з'ясувати наскільки уразливий мережевий екран, хоча і існують спеціальні прилади, за допомогою яких можна перевірити стандартні помилки безпеки. Маршрутизатори, як правило, не записують ніякої контрольної інформації і тому хакер легко може "замісти сліди".

Шлюзовий екран — екрануюча система, яка складається з єдиного захищеного комп'ютера-бастіона, що повинен блокувати пряму передачу IP-пакетів між окремими мережами.

Адміністратор екрану може встановлювати на шлюзовий комп'ютер сервісні агенти або дозволяти користувачам доступ до шлюзового комп'ютера. Оскільки шлюзовий екран складається з набору сервісних агентів, то він може самостійно проводити аутентифікацію зовнішніх користувачів і відкривати канали доступу з зовнішніми комп'ютерами. Однак використання агентів звичайно ускладнює роботу користувачів, створюючи додаткові труднощі через необхідність навчати виконавців поводженню з новим програмним забезпеченням.

А якщо користувач має свої ресурси на бастіоні і робить їх легкодоступними (наприклад, визначить пароль, що легко розпізнається), то це послабить його захист. Крім того, нападник може використати загальновідомі методи нападу, оскільки бастіон стає універсальним комп'ютером. Однак при жорсткій перевірці захисту та зміні конфігурації комп'ютера-бастіона "зламати" таку систему не так просто.

Часто комп'ютер-бастіон замінює адреси внутрішніх машин своїм особистим, і зовнішнім користувачам буде здаватись, що вони підтримують зв'язок лише із захисним екраном. Це використовується для захисту від прослухування внутрішніх потоків інформації корпоративних мереж. Таким чином, шлюзовий комп'ютер є представником корпоративної мережі в Internet.

Більш надійна екрануюча система — ***захищений комп'ютерний шлюз***. Він створюється з комп'ютера-бастіона, який знаходиться в корпоративній мережі, та керуємого ним комп'ютерного маршрутизатора, встановленого між зовнішньою та внутрішньою мережами. Маршрутизатор налаштовується так, щоб весь потік інформації проходив лише через комп'ютер-бастіон. В такій конфігурації бастіон захищений екрануючим маршрутизатором, оскільки немає, як такої, взаємодії з зовнішньою мережею.

Ця конфігурація поєднує переваги екрануючого маршрутизатора та шлюза. Конфігурація маршрутизатора проста, оскільки весь потік інформації (крім забороненого) направляється в комп'ютер-бастіон. В той же час маршрутизатор блокує не-

безпечні пакети, не допускаючи їх навіть до бастіона, що прискорює роботу екрануючого шлюзу. Зовнішній користувач може отримати доступ тільки до двох приладів — комп'ютера-бастіона та екрануючого маршрутизатора.

Для зруйнування захищеного комп'ютерного шлюзу нападник повинен змінити параметри трьох приладів: комп'ютера-бастіона, довільного комп'ютера корпоративної мережі і тільки потім довільного маршрутизатора мережі. Причому це необхідно зробити обережно, щоб не розірвати свій зв'язок з системою і не зчинити галас.

Захищена підмережа — спеціальна ізольована підмережа, яка знаходиться між зовнішніми та внутрішніми мережами, що побудована так, щоб обидві мережі мали доступ до її комп'ютерів, але пряма передача повідомлень між ними при цьому блокувалась. Така підмережа може містити всі три компоненти захисту: комп'ютери-бастіони, екрануючі маршрутизатори та шлюзи. Звичайно доступ в екрануючу підмережу зовні неможливий, а тільки через бастіонний комп'ютер, що зменшує ризик нападів: зовнішній користувач може отримати доступ лише до одного або декількох комп'ютерів-бастіонів та маршрутизаторів.

Захищену підмережу звичайно використовують для оборони розгалужених корпоративних мереж з великим обсягом інформації, що передається через екран, та декількома каналами зв'язку.

Змішаний шлюз — тип захищеного екрану, який не підходить ні до одного вище приведеного опису. Нетривіальна конфігурація екрану або нестандартне обладнання, яке використовується для його створення, затримує хакера на деякий час і дозволяє адміністратору системи вчасно помітити атаку.

В майбутньому очікується бурхливий розвиток глобальних і розподілених корпоративних мереж і тому створюються нові, не описані тут, системи захисту. Але основні принципи побудови екранів, певно, залишаться незмінними.

Розділ 7

Правові основи Internet

"Говори або стисло, або зрозуміло"
(Народна мудрість)

Незаперечним фактом сьогодні є прояв масового інтересу до Internet. Значимість суспільних відносин, що виникають у зв'язку з її використанням, не викликає сумніву. Тому природно, що і юристи не залишають без уваги глобальну мережу. З'являється об'єктивна потреба в правовому врегулюванні нової галузі. Відповідно виникає нова юридична спеціалізація — Internet-право.

Розвиток законодавства в сфері сучасних інформаційних технологій набрав обертів в усіх провідних країнах світу. Що ж являє собою Internet-право, у чому виявляється його юридична природа?

Проблеми врегулювання суспільних відносин, пов'язаних з використанням Internet, дуже актуальні для України. Сьогодні в Україні існує цілий ряд нормативних актів, що регулюють відносини в сфері інформації і відповідно торкаються питань використання глобальної мережі. Один з таких документів — Указ Президента "Про заходи щодо розвитку національної складової глобальної інформаційної мережі Internet та забезпечення широкого доступу до цієї мережі в Україні" № 928/2000 від 31 липня 2000 року.

На відміну від України, деякі країни мають набагато більший досвід правотворення в розробці і прийнятті правових норм для врегулювання відносин пов'язаних з роботою в Internet. Так, у Німеччині і США ще в середині 90-х років був прийнятий закон "Про електронно-цифровий підпис". Поширення забороненої інформації через мережу привело до того, що в Австралії в ті ж роки були прийняті закони, спрямовані на врегулювання змісту інформації в глобальній мережі, у Німеччині працює закон "Про відповідальність провайдера".

При прийнятті Україною власних законів необхідно врахувати позитивний досвід цих держав. Займаючись правотворенням у цій галузі, варто враховувати її специфіку і звичайно ж наявні розробки і досягнення юридичної науки.

Характерним для сучасного юридичного порядку є поділ норм права на дві великі групи: на право публічне і право приватне. Незважаючи на традиційність цього поділу, з наукової точки зору до цих пір остаточно не з'ясовано, де знаходиться грань, що розмежовує публічне і приватне право. Критерієм розмежування цих галузей виступає метод правового регулювання. У відносинах публічно-правового характеру усе підпорядковано волі держави, застосовується метод влади і підпорядкування. У сфері відносин приватного характеру застосовується метод автономії, іншими словами, учасники приватноправових відносин мають можливість визначити їхній зміст самі.

Для врегулювання проблем глобальних мереж розроблені правові норми Internet-права, які можна віднести і до публічної галузі, і до приватного права, тому що зачіпаються однаковою мірою інтереси як держави, так і окремого індивіда.

В Internet-праві інтереси держави і суспільства порушуються, насамперед, коли відбувається розміщення в мережі несанкціонованої інформації. Інтереси приватні переважають при проведенні реєстрації доменних імен, захисту інтелектуальної власності тощо. Дуже динамічно розвивається електронна комерція, що містить величезний масив питань. Основними з них є порядок проведення розрахунків між продавцем і покупцем, питання оподатковування, криптографічного захисту і застосування електронно-цифрового підпису.

Тобто, при правовому врегулюванні відносин, зв'язаних з використанням Internet, варто застосовувати спільно метод влади-підпорядкування і метод автономії.

7.1. Правове регулювання у сфері інформаційних відносин

Світовий досвід свідчить про те, що на рівні законів регулюється три групи правовідносин:

- право громадян на доступ до інформації;
- захист інформації;
- охорона виключних прав.

Принципово не має значення, в якому середовищі, в яких умовах реалізуються конкретні суспільні відносини — якщо відбувається порушення прав та інтересів, то повинен бути забезпечений їх захист. Таким чином, права та інтереси повинні захищатись, в тому числі і в сфері Internet, а певна їх сукупність — і владою закону. При вирішенні питання, яке законодавство застосовувати: традиційне чи специфічно нове, — слід мати на увазі, що суспільні відносини, які мають правову специфіку завдяки використання Internet, потребують створення нових правових норм, а всі інші — забезпечуються нормами діючого законодавства.

Сьогодні відомо не так багато країн, де створюється національна система законодавчого регулювання відносин в глобальному інформаційному просторі. Міжнародні угоди з цих проблем ще тільки розробляються, судова практика ще тільки формується. Однак нинішня ситуація несе у собі своєрідну загрозу для ряду країн, які не приділяють належної уваги цим питанням. Загроза полягає у можливій юридичній експансії, яка призведе до спроб поширення юрисдикції національної системи законодавства однієї чи кількох країн на відношення у всій мережі Internet.

Перед юридичною системою країни стоїть дуже важливе завдання розробити систему національного законодавства у сфері відносин глобального інформаційного простору, яке повинно бути максимально гармонізованим з міжнародним законодавством, враховувати тенденції та напрямки його розвитку. Це нормативно-правові акти, які встановлюють юридичний статус електронного документу, цифрового підпису, ЗМІ в Internet, такі, які регулюють особливості авторського права, економічних взаємовідносин в умовах мережевих технологій, такі, які встановлюють відповідальність за комп'ютерні злочини та інше.

У 70-х роках з'явились перші дослідження щодо правового режиму машинної інформації, а саме — програм для ЕОМ, а на початку 90-х років видатний російський цивіліст Е. А. Суханов доводив, що найбільш загальними, принциповими проблема-

ми правової регламентації відносин у цій сфері є правова охорона програм для ЕОМ та економічно-правовий режим інформаційних ресурсів. На думку вченого, вирішення цих ключових питань повинно стати основою всієї наступної законодавчої регламентації інформаційних відносин, тому що саме вони визначають і характер цієї регламентації, і її місце в існуючому правопорядку, і можливі форми законодавчого розвитку.

Від цього, наприклад, залежить, чи буде законодавство про інформатику розвиватись в якості складової частини цивільного або господарського права, а, можливо, буде поступово відокремлюватись у самостійну законодавчу галузь чи підгалузь комплексного характеру, єдність якої повинна забезпечуватись окремим законодавчим актом кодифікаційного типу.

Якщо правовий режим програм для ЕОМ та інших інформаційних ресурсів буде будуватись за одним (або навіть декількома) із відомих традиційній правовій системі напрямками (наприклад, по типу авторсько-правового захисту, як це відбувається в наш час у багатьох розвинених країнах), а їх обіг в якості товарного продукту включатись у сферу майнових відносин, регламентованих цивільно-правовими договорами та іншими інститутами цивільного права, то мова буде йти лише про модифікацію цивільного законодавства, певному його пристосуванні до специфіки умов "інформаційного суспільства". Якщо ж правова регламентація цих інформаційних відносин вийде за вказані межі, то вже можна буде говорити про наявність поряд зі спеціальним законодавством окремого "права інформатики" як комплексного правового утворення.

Безумовно, відповідь на це запитання цілком залежатиме від реального стану інформаційних процесів у суспільстві, а також від перспектив їх розвитку. З урахуванням наявних обставин обидва ці підходи (як і можливі перехідні позиції) мають право на життя.

Відомі цивілісти Ю. К. Толстой і А. П. Сергєєв зазначають в своїх роботах, що у сучасному світі інформація виступає в якості особливого об'єкту договірних відносин, пов'язаних з її збиранням, збереженням і пошуком, переробкою, поширенням і використанням у різних сферах людської діяльності. Особливе місце при цьому відводиться машинній інформації, що циркулює в комп'ютерному середовищі, яка зафіксована на фізичному носії у формі, доступній для сприйняття ЕОМ, або така, що передається по телекомунікаційних каналах.

На думку вчених інформація, як об'єкт цивільних прав повинна мати наступні ознаки:

- інформація є ідеальним компонентом буття, тобто благом нематеріальним;
- інформація — благо неспоживче, яке піддається лише моральному, але не фізичному старінню;
- інформація має можливість необмеженого тиражування, поширення і перетворення форм її фіксації.

Крім того, не слід закріплювати за будь-ким монополії на володіння і використання інформації, за винятком тієї, що є одночасно об'єктом інтелектуальної власності або цивільних прав.

Автори А. Г. Калпін та А. І. Масляєв вказують на множинність об'єктів цивільних правовідносин, серед яких є інформація та результати інтелектуальної діяльності.

Іншими словами, використовуючи термін "інформація", юристи мають на увазі фактично новий об'єкт цивільних прав.

Сучасний період розвитку інформаційних процесів характеризується розробкою законодавства, покликаного регулювати різні аспекти цього складного явища. Під інформацією розуміють відомості про осіб, предмети, факти, події, явища та процеси незалежно від форми їх представлення. Окремі документи й окремі масиви документів, документи і масиви документів в інформаційних системах, у свою чергу, становлять інформаційні ресурси.

Визначення інформації закріплене у статті 1 Закону України "Про інформацію" від 2 жовтня 1992 року, згідно якому під ***інформацією*** слід розуміти документовані або публічно оголошені відомості про події та явища, що відбуваються у суспільстві, державі та навколишньому природному середовищі. У статті 18 названого закону перераховані основні види інформації: статистична, масова, інформація про діяльність державних органів влади та органів місцевого та регіонального самоврядування, правова, інформація про особу, довідково-енциклопедичного характеру, соціологічна тощо.

У ряді законодавчих актів наведені визначення інших видів інформації, зокрема, науково-технічної як документовані або публічно оголошувані відомості про вітчизняні та зарубіжні досягнення науки, техніки і виробництва, одержані в ході науково-дослідної, дослідно-конструкторської, проектно-технологічної, виробничої та громадської діяльності. Науково-технічна інформація поділяється на два види: відкриту та з обмеженим доступом. З урахуванням розвитку інформаційних відносин, наведений перелік видів інформації не вважається вичерпним.

Крім того, стаття 28 Закону України "Про інформацію" дає поняття відкритої інформації та інформації з обмеженим доступом, покладаючи в основу такого поділу передбачений нормами права порядок одержання, використання, поширення і зберігання інформації. Контроль за режимом доступу до інформації здійснюється державою і покладається на спеціальні органи, які визначають Верховна Рада України та Кабінет Міністрів України.

Одним із об'єктивних показників зростання ролі права в регулюванні суспільних відносин у сфері інформаційних процесів є прийняття в Україні законів "Про інформацію" 1992 року і "Про науково-технічну інформацію" 1992 року, в яких інформація розглядається як об'єкт права власності. Ці закони передбачають створення правової бази для одержання і використання інформації у сфері товарно-грошових відносин. Фізичні та юридичні особи, які реалізують інформаційний продукт, вважаються товаровиробниками з усіма правами та зобов'язаннями.

Що стосується інтелектуальної суті машинної інформації, то питання її правової охорони зводиться до того, щоб нетиповій (машинній) інформації надати типовий правовий режим, розв'язавши комплекс суперечливих питань. Питання щодо наявності (або відсутності) творчих зусиль при створенні машинної інформації як інтелектуального продукту є достатньо складним і дебатується не один рік.

В свою чергу, Закон України "Про обмеження монополізму та недопущення недобросовісної конкуренції у підприємницькій діяльності" від 18 лютого 1992 року визначає ***інформацію*** як відомості у будь-якій формі та вигляді, на будь-яких носіях (у тому числі листування, книги, помітки, ілюстрації (карти, діаграми, органіграми, малюнки, схеми тощо), фотографії, голограми, кіно-, відеофільми, мікрофільми, звукові

записи, бази даних комп'ютерних систем або повне чи часткове відтворення їх елементів, пояснення осіб та будь-які інші публічно оголошені чи документовані відомості. "Конвенция о формировании информационного пространства СНГ", затверджена Рішенням Ради голів держав СНД від 18 жовтня 1996 року під інформацією розуміє відомості про осіб, предмети, факти, явища та процеси незалежно від форми їх представлення.

Таким чином, інформація — важлива складова частина єдиного поняття "інформаційна система", що є абсолютно логічним, коли мова йде про інформаційні системи як об'єкт цивільно-правового регулювання для з'ясування в зв'язку з цим місця та ролі інформації.

7.2. Теоретична концепція розвитку інформаційного законодавства

7.2.1. Суть концепції

Специфічним предметом правового регулювання інформаційного законодавства пропонується визначити суспільні відносини, що складаються в процесі інформаційної діяльності суспільства, держави, реалізації права на інформацію в таких основних галузях, як:

- пошук, одержання, споживання інформації, її виробництво, переробка для формування інформаційних ресурсів;
- надання інформації, створення і застосування інформаційних систем, інформаційних технологій;
- створення механізмів захисту інформації від несанкціонованого доступу (інформаційна безпека).

Окреслені напрямки визначають склад, структурну схему законодавства України у відповідній сфері, що складають спеціальні закони, підзаконні акти (укази Президента України, постанови Кабінету Міністрів України та ін.), окремі норми інших галузей законодавства.

Крім того, процес законотворення триває. Це дає правознавцям підстави говорити про існування в національній правовій системі окремої самостійної комплексної галузі — інформаційного законодавства.

Необхідно мати на увазі, що цивільне законодавство регулює майнові відносини, які зумовлені використанням товарно-грошової форми в суспільстві і пов'язані з ними особисті немайнові відносини.

Отже, предметом цивільно-правового регулювання є:

- майнові відносини, зумовлені використанням товарно-грошової форми в суспільстві;
- особисті немайнові відносини.

Майнові відносини, як відомо з теорії цивільного права, поділяються на три групи: відносини власності, відносини в галузі товарообігу і майново-організаційні. Саме перші дві групи регулюються цивільним правом.

Особисті немайнові відносини виникають у зв'язку із здійсненням особистих прав і поділяються на:

- особисті немайнові відносини, пов'язані з майновими, що має відношення до нашого об'єкту — інформаційних систем, бо маються на увазі особисті права авторів творів в галузі науки, літератури, наукових відкриттів і т. ін.;
- особисті немайнові відносини, які виникають та існують незалежно від майнових, тобто не пов'язані з ними.

Таким чином, визначаючи предмет правового регулювання в сфері створення і використання програм, баз даних та інформаційних систем, слід говорити про суспільні відносини, що складаються у процесі діяльності суспільства, держави, реалізації права на інформацію в таких галузях як пошук, одержання, використання інформації; виробництво, переробка для формування інформаційних ресурсів; надання інформації, створення і застосування інформаційних технологій. Тобто, за змістом предмет правового регулювання, створення і використання інформаційних систем співпадає з загальноінформаційним. Цей висновок відповідає і місцю інформаційних систем в інформаційному законодавстві України.

Одною із складних теоретичних і практичних проблем є проблема не тільки одночасного впливу права на весь об'єкт вцілому, але й проблема ефективного поєднання та синхронної дії норм, які відносяться до різних галузей права в регулюванні єдиного процесу інформаційної діяльності суспільства, співвідношення диференційованого та інтегрованого підходів в такому регулюванні.

Для нормативного правового масиву в інформаційній галузі характерним є велика ступінь диференціації правового впливу, яке породжує відсутність поєднання (сумісності) норм різних галузей права, а іноді і їх протиріччя, що призводить до розриву між окремими ланками механізму правового регулювання та не забезпечує необхідного правонаступництва норм в його комплексному регулюванні.

Існує об'єктивна закономірність між предметом правового регулювання, методом і правовими засобами, які притаманні даному методу. Якщо ж будь-який вид діяльності, в даному випадку діяльність у галузі використання інформаційних систем, поєднує у собі елементи, характерні для різних видів суспільних відносин, тоді й правове регулювання повинно трансформувати в собі особливості цих видів відносин або їх окремих елементів. При цьому, однак, повинні враховуватись й існуючі об'єктивні закономірності в поєднанні та взаємозв'язку норм різних галузей права, що застосовуються в сукупності для регулювання всього обсягу суспільних відносин в даній галузі.

Реалізація на практиці названих принципів дозволить забезпечити рівновагу у використанні норм різних галузей права, зробить правове регулювання інтегрованим.

7.2.2. Законодавство про інформаційні відносини у сфері авторського права

Аналізуючи законодавство у сфері створення та використання інформаційних систем, неможливо обминути проблему, вирішення якої знаходиться в сфері авторського права.

Як вже підкреслювалось, інформаційну систему можна віднести до творів в галузі науки, а саме до таких, які перераховані у статті 5 Закону України "Про авторське право і суміжні права" 1994 року. Важливим завданням на сьогодні є необхідність забезпечення авторсько-правовою охороною інформаційні системи як "збірники творів, збірники звичайних даних, включаючи бази даних, інші складові твори за умови, що вони є результатом творчої праці по добору, координації або упорядкуванню змісту без заподіяння шкоди охороні творів, що входять до них".

Попередньо проведені дослідження дають можливість з упевненістю вести розмову про виникнення нового об'єкту авторсько-правового регулювання, про відповідність його ознак правовим ознакам, що притаманні творам науки.

У зв'язку з цим, доцільно було б доповнити статтю 4 Закону терміном "інформаційна система" у такому тлумаченні: "Інформаційна система — сукупність інформаційних ресурсів і програмного забезпечення у формі, доступній для сприйняття машиною, функції якої забезпечуються інформаційними технологіями і інформаційними процесами". Зрозуміло, що поняття "інформаційна система" не поширюється на поняття "комп'ютерна програма" і "база даних", значення яких наведено в Законі.

Позитивним є те, що Законом охоплюється більшість об'єктів, про які йшла мова під час розгляду складових інформаційної системи. До творів, що охороняються, відносяться комп'ютерні програми, збірники звичайних даних, бази даних, операційні системи; а стаття 18 повністю присвячена питанню вільного відтворення програм. Без цих досліджень було б складно сьогодні вести мову про інформаційні системи.

Існує необхідність, можливо, дати визначення поняттям "інформаційні ресурси" і "програмне забезпечення", які існують в інших законодавчих актах, а також найбільш сучасному виду інформаційних систем — експертним системам. Разом з тим, може виникнути необхідність ввести поняття спеціальної (і, особливо, експертної) системи.

З урахуванням ряду особливостей нового об'єкту авторсько-правового регулювання найбільш повно розкрити всі аспекти можна за допомогою спеціального закону про правову охорону інформаційних систем, включивши до нього наступні розділи та положення:

- загальні положення, до яких слід включити:
  - основні поняття;
  - відносини, які регулюються законом і пов'язані зі створенням, правовою охороною та використанням інформаційних систем, наданням інформаційній системі правової охорони як збірникам;
  - об'єкт правової охорони;
  - умови визнання авторського права;
  - авторське право на інформаційні системи, як на збірники;
  - строк дії авторського права;
  - сфера дії даного закону;
- виключні авторські права:
  - авторство;
  - особисті права;
  - майнові права;
  - передача майнових прав;

- майнові права на інформаційну систему, створену в порядку виконання службових обов'язків;
- право на реєстрацію;

- використання інформаційних систем:
  - використання інформаційних систем за договором з правоволодільцем;
  - вільне відтворення;
  - вільний перепродаж екземпляру інформаційної системи;
- захист прав:
  - порушення авторського права;
  - захист прав на інформаційні системи;
  - інші форми відповідальності.

Такий нормативний акт має розкрити суть практично всіх перерахованих положень, хоча для прийняття закону необхідні додаткові дослідження.

Стосовно всіх ускладнюючих питань, про які йшлося в попередніх параграфах розділу, висновків, зроблених про можливість комплексного регулювання розглядуваного об'єкту за допомогою і авторського, і винахідницького права — то на нинішньому етапі доцільно вести мову про авторсько-правову охорону з подальшим вивченням проблеми. Адже пункт 2 Закону "Про авторське право" вказує на те, що охорона авторським правом не поширюється на будь-які ідею, процедуру, метод, процес, концепцію, відкриття, винахід, корисну модель, промисловий зразок, знак для товарів і послуг, раціоналізаторську пропозицію, звичайні дані, навіть якщо вони виражені, описані, пояснені, проілюстровані у творі тощо.

У даному переліку не наведені системи, і це не заважає говорити про системи (інформаційні та експертні зокрема) як про об'єкт і авторського і винахідницького права. Можна розглядати ті ж інформаційні системи з точки зору інформаційних технологій, в тому числі з використанням засобів обчислювальної техніки і зв'язку, що реалізують інформаційні процеси. І тоді це виявляється скоріше об'єктом винахідництва, у випадку, якщо наявні всі ознаки: новизна або нове технічне рішення і т. ін.

Комплексне вирішення цього питання в межах нового інституту — єдине вірне рішення. І це далеко не останній об'єкт в інформатиці, проблеми і протиріччя якого можливо вирішити повноцінно, тільки ідучи шляхом комплексної охорони об'єктів інтелектуальної власності.

Слід зазначити, що твори науки, літератури, мистецтва, відкриття, винаходи, промислові зразки та інші результати інтелектуальної праці віднесені сьогодні до об'єктів права власності, тобто не лише результати творчої діяльності закріплені як об'єкти права інтелектуальної власності, а й сама ця діяльність. Отже, законодавець робить коло об'єктів права власності надзвичайно широким.

В результаті широкої дискусії були висловлені різні точки зору з цього приводу. Перша — породжена умовами, в яких автор визначається товароволодільцем, який відчужує за гроші свій твір, а право власності проголошується "природним правом людини". Друга базується на тому, що, не заперечуючи практичної необхідності таких підходів, підкреслює, що навіть коли ми говоримо про ринкові відносини, немає достатніх обставин для спроб підкреслити абсолютний характер даних прав на нематеріальні об'єкти розумової праці, ототожнюючи їх з правом приватної власності на речі.

Однак, як вже підкреслювалось, найбільш прийнятною є позиція, яка стверджує, що результати творчої праці, що охороняються правом, мають свою специфіку і не можуть безпосередньо кваліфікуватися як звичайні об'єкти права власності, як це відбувається у випадку аналогічного права на матеріальні продукти людської діяльності або природні об'єкти.

7.3. Правовий статус учасників у сфері інформаційних відносин

7.3.1. Автор програм, баз даних та інформаційних систем

Більшість цивілістів виходять з розуміння автора як особи, творчою працею якої створюються твори. За законом носіями авторських прав в більшості країн Європи, США та Японії, перш за все, є громадяни — автори творів.

З цього витікає, що першою категорією суб'єктів в процесі створення програм, баз даних та інформаційних систем є їх творці, які є авторами результату такої творчої праці. Необхідно у зв'язку з цим вказати на те, що поняття "автор твору" і "суб'єкт авторського права" не є тотожними як за значенням, так і за змістом. Це означає, що суб'єктом творчої діяльності може бути тільки людина або група (колектив) людей за умови їх персонального творчого внеску, а суб'єктами авторських відносин — як автор твору, так й інші фізичні та юридичні особи, для яких право може виникати в силу закону, договору або успадкування.

На визнання творця суб'єктом авторського права його вік не впливає. Автори у віці від 15 до 18 років здійснюють свої авторські права самостійно, а за недієздатних авторські права здійснюють від їх імені батьки, усиновителі або опікуни. Це означає, що теоретично створити нову інформаційну систему може один автор, навіть малолітній, тобто дієздатність суб'єктів цих відносин може бути дещо більшою за загальну. Але практично важко уявити, щоб таке складне явище як інформаційна система стало результатом творчості дитини, або зусиль одного автора.

Важливим положенням є те, що громадянин є суб'єктом авторського права з моменту створення твору. Для визнання особи суб'єктом авторського права в сфері інформаційних відносин непотрібна будь-яка реєстрація твору. На визнання творця суб'єктом авторського права не впливає й те, що твір створено ним в порядку виконання службового завдання. В цьому випадку може бути встановлений інший порядок використання твору й обмежені права автора на винагороду.

7.3.2. Роль роботодавця

Свого часу союзне законодавство про інтелектуальну діяльність істотно підняло роль роботодавця, який надав творцеві (або творцям) роботу, як суб'єкта права на результат творчої праці. Якщо твір виконувався в порядку службового завдання, авторське право належало громадянину — його творцеві. Організації в даному випадку не належало будь-яких правочинностей. Однак вона могла використати твір, і порядок такого використання встановлювався законодавством бувшого СРСР і постано-

вами Ради Міністрів союзних республік. Створення цих творів в більшості випадків вважалось оплаченим за рахунок платні, яку автор отримував за місцем роботи. В деяких випадках припускалась виплата гонорару крім заробітної плати.

На сьогодні ситуація змінилась. Законодавство в сфері авторських прав і стаття 435 Цивільного кодексу України передбачають наступне: права на об'єкт права інтелектуальної власності повинен мати її творець — автор. І тільки він може змінити це за своїм бажанням, уклавши відповідний договір.

7.3.3. Суб'єкти авторського права

Суб'єктом авторського права — творцями програм, баз даних та інформаційних систем — може бути як громадянин України, так й іноземець. За громадянами України авторське право визнається незалежно від того, чи з'явився їх твір вперше на території України або за кордоном.

Слід вести розмову про можливість охорони інформаційних систем на території України у випадках, якщо ці твори вперше вийшли в світ на території України або не випущені в світ, але знаходяться на території України в будь-якій об'єктивній формі і якщо вони охороняються відповідно двосторонніх угод, так само, як передбачено для програм і баз даних. Подібні угоди слід укласти між Україною й іншими країнами. Інформаційні системи авторів цих країн повинні охоронятись в Україні в межах встановлених термінів охорони незалежно від часу та місця їх появи, а використовуватися повинні тільки за згодою авторів або їх правонаступників.

7.3.4. Суб'єкти винахідницького права у сфері інформаційних відносин

Якщо при створенні інформаційної системи з'являються елементи технічної творчості, то можливо говорити про суб'єктів винахідницького права. Не зупиняючись детально на питанні, відмітимо, що суб'єктами винахідницького права можуть бути їх автори, а також інші суб'єкти цивільного права в силу закону чи договору. Їх склад аналогічний з тим, який зустрічається в авторському праві.

Особа стає суб'єктом права на програми, бази даних та інформаційні системи як автор, як правило, за умови:

- творчого технічного рішення;
- кваліфікації цього рішення у встановленому порядку з боку держави у формі видачі авторського свідоцтва, патенту на винахід.

Особа, яка створила інформаційну систему, не може бути суб'єктом названого права до відповідного визнання з боку компетентного органу. Це, однак, не заважає вважати його автором відповідної ідеї, яка отримала зовнішнє вираження в описі і без визнання прав. Але подібне авторство ще не надає авторові всіх прав. Тільки винахідники, які отримали авторське свідоцтво, мають право на винагороду. А інші права і пільги, передбачені Законом про винахідництво, також належать лише авторам, які отримали зазначені документи.

Складність охорони винахідницького права є основною завадою для застосування її до об'єктів інформаційних відносин, які потребують такої охорони відразу після їх появи. Тому розмова йде про суб'єктів винахідницького права лише у плані

перспективного вирішення питання про створення ефективного механізму охорони інформаційних систем як за допомогою методів авторського, так і винахідницького права.

7.3.5. Створення програм, баз даних та інформаційних систем у співавторстві

Тепер проаналізуємо більш реальний і поширений випадок, коли програму, базу даних, інформаційну систему створено не одним автором, а спільною творчою працею кількох співавторів. Тоді складаються відносини співавторства, що є характерним і для авторського і для винахідницького права. Але це повинно стати не наданням простої технічної допомоги, а справді співпрацею творчого характеру. Такі відносини між повноправними авторами, безперечно, краще попередньо закріпити в угоді між ними, передбачаючи при цьому рівність прав всіх співавторів.

Це більше відповідає створенню такого об'єкту, як інформаційна система, однак угоду про участь тієї чи іншої особи в роботі над твором можна укласти на будь-якому етапі творчого процесу, втому числі й після завершення роботи, коли виникає необхідність внести до твору зміни творчого характеру.

Сумісна творча праця по створенню колективного твору за загальним правилом ґрунтується на попередній угоді співавторів. Але в ряді випадків відносини співавторства можуть виникнути й за відсутності такої згоди. Так, співавторство може бути встановленим в судовому порядку, якщо і не було попередньої угоди.

У випадку створення серйозних розробок для державних потреб інформатизації доцільним буде виникнення відносин співавторства в результаті сумісної праці творчої групи, спеціально сформованої для розробки наукової проблеми.

Згідно з законодавством співавтори — це особи, які творчо приймали участь у створенні твору. Таким чином, під твором, створеним у співавторстві, слід розуміти твір, створений у співпраці двох або більше авторів, при якому вклад кожного автора є невід'ємним від вкладу іншого автора або авторів. У таких випадках твір охороняється в особі всіх його творців.

У результаті сумісної творчої діяльності створюється інформаційна система, яка характеризується єдністю форми і змісту, або необхідним внутрішнім зв'язком двох або більше форм, обумовленим єдиним змістом (інформаційні ресурси та програмне забезпечення). Співавторство, як відомо, має місце як у випадку, коли твір являє собою одне нерозривне ціле, так і в тому випадку, коли воно складається з частин, кожна з яких має самостійне значення. В першому випадку має місце нероздільне співавторство, в другому — роздільне. Останній більш притаманний інформаційній системі як об'єкту авторського права.

Особи, які визнаються співавторами, сумісно реалізують право на твір. Особи, які надавали технічну допомогу, не визнаються співавторами. Зрозуміло, що як і раніше, при використанні ЕОМ для роботи інформаційної системи, а тим більше для такої, що працює в єдиній мережі (районній, міжрайонній, регіональній), необхідні спеціалісти для інформаційного аналізу матеріалу і запитів, механічного або традиційного відтворення їх змісту, для підтримки технічної працездатності системи. Це означає, що крім творчих осіб, авторів таких проектів, створення сучасної інформаційної системи неможливо уявити собі без інженерів, операторів та іншого технічного

персоналу, який допомагає автору. Тому якість створеної інформаційної системи буде залежати від кваліфікації спеціалістів, їх оперативності і доброчинності.

Протилежні твердження базуються на занадто високій оцінці можливостей технічних засобів і нехтуванні незалежною роллю спеціалістів, які приймають участь у створенні і експлуатації інформаційних систем. Але висока оцінка діяльності таких спеціалістів не є підставою для її ототожнення з творчою працею автора.

Не має творчого характеру і не вважається творчістю запозичення, перенесення і використання чужого, навіть позитивного досвіду або використання за ліцензією. І з цього приводу також можна зробити певний висновок: творчий характер діяльності означає передусім певну індивідуальність, тобто те, що відрізняє цю діяльність від інших видів діяльності.

Слід зазначити, що для заповнення бази знань, складання програми потрібен також спеціаліст, який вміє узагальнити знання експерта і перевести їх на доступну ЕОМ мову — когнітолог. В цьому випадку висловлюється думка про те, що необхідне визнання сумісної однакової творчої праці двох повноправних суб'єктів, і тому обидва вони повинні розглядатись як співавтори авторського права колективного твору, що охороняється, — експертної системи, тому відносини між співавторами слід визначити за договором (домовленістю, угодою) і встановити порядок розподілу винагороди між співавторами за створення експертної системи, порядок і способи визначення її назви як за авторський твір. І експерту і когнітологу має належати весь комплекс прав на створену експертну систему.

Від співавторства слід відрізняти співпрацю. Наприклад, об'єднання творів різних авторів і в той же час новий твір як результат творчої роботи по добору і систематизації матеріалу як єдиного твору. В цій творчій роботі окремі автори участі не приймають, і по відношенню до такого твору вони є лише співробітниками, а авторське право на єдиний твір належить відповідній організації.

7.3.6. Інші суб'єкти права на програму, бази даних та інформаційні системи

В теорії права поняття "суб'єкт права", пов'язане з творчою діяльністю (зі створенням програми, бази даних, інформаційної системи), стосується як автора, так і його правонаступників — будь-яких фізичних чи юридичних осіб, яким автор передав своє суб'єктивне майнове право на результат своєї творчої праці. Такими правонаступниками можуть виступати, наприклад, спадкоємці автора, юридичні особи або держава.

При цьому необхідно пам'ятати, що суб'єктами зазначених прав можуть бути будь-які фізичні та юридичні особи, до яких суб'єктивні права авторів перейшли за договором або за заповітом, чи в силу закону. Саме в цьому випадку авторське право автора називають первинним, а авторське право правонаступників — похідним.

Слід звернути увагу, що до спадкоємців переходить не весь обсяг авторських правомочностей, а лише право на опублікування, відтворення і поширення твору, а також право на винагороду за використання твору іншими особами в межах, встановлених законом.

Автор може в заповіті вказати на конкретну особу, на яку він покладає охорону недоторканності творів після його смерті. Ця особа здійснює повноваження охорони

довічно. Якщо таке розпорядження автором не зроблено, то правомочність охорони недоторканності твору виникає у спадкоємця одночасно з переходом до нього прав успадкування. Якщо ж спадкоємців немає або авторське право припинилось, то охорону здійснюють державні організації, на які законом покладена охорона авторських прав.

При житті автора суб'єктами похідного авторського права на інформаційні системи слід визнати організації. Похідне авторське — право організацій — виникає лише на підставі авторського договору, про що йшлося вище, крім випадків, зазначених в законі.

Якщо автор й укладає з організацією договір, у неї виникають не лише обов'язки перед автором, але і права, наприклад, право вимагати від автора своєчасної передачі твору для використання: протягом строку дії авторського договору не передавати твір іншим організаціям.

7.3.7. Позначення авторських прав

З метою охорони твору в усіх країнах, учасницях Всесвітньої конвенції про авторське право 1952 року, організація наділена правом на виданнях творів ставити особливий знак (©), наявність якого означає, що без дозволу першовидавця ніхто не має права перевидавати твір. Слід таким правом користуватись і для позначення такого твору як інформаційна система. В законі можуть бути встановлені права організації по використанню творів, створених в порядку виконання службових завдань.

7.3.8. Юридичні особи як суб'єкти авторського права

Слід відмітити, що носіями авторського права при створенні програм, баз даних, інформаційних систем можуть бути юридичні особи. Однак юридичні особи, враховуючи їх спеціальну правоздатність, визнаються суб'єктами авторського права не на загальних підставах, а лише у випадках, встановлених законодавством.

Можна передбачити, що авторське право на інформаційну систему буде належати відповідним організаціям, які здійснюють роботи по формуванню випуску в світ цих творів, як збірник, який є об'єднанням за відповідною системою (принципом) творів різних авторів. Автори творів, включених в такі видання, не отримують авторських прав на інформаційну систему в цілому, а організація не набуває авторського права на твори цих авторів.

У всіх вказаних випадках авторське право, яке належить організації, виникає на самостійний об'єкт як результат її творчої діяльності. При цьому юридична особа не привласнює чиє-небудь авторське право. Ці твори є результатом творчої діяльності організації як відповідним чином організованого колективу людей, тому що він виконав творчу роботу, необхідну для об'єднання самостійних творів окремих авторів в цілісний твір. Тому й авторське право на нього визнається за юридичною особою.

Сучасний період розвитку інформаційних систем характеризується необхідністю комплексного підходу до вирішення проблем охорони даного об'єкту авторським правом і правом промислової власності.

З урахуванням цієї особливості закономірно виникає потреба розглянути в схематичному вигляді суб'єктний склад відносин, які є результатом творчої діяльності по розробці інформаційних систем.

Перш за все, обидва правові інститути регулюють відносини, які складаються у зв'язку з творчою діяльністю людей, тому і в авторському праві, і в патентному законодавстві підкреслюється, що творцем може бути тільки людина, тобто суб'єктом творчої діяльності в сфері створення інформаційних систем може бути лише людина або група (колектив) людей, а не організації (юридичні особи). Проте суб'єктами авторських відносин і відносин промислової власності (прав і обов'язків) можуть бути як громадяни, так і юридичні особи.

Спільним є те, що надання будь-якої технічної допомоги автору інформаційної системи чи винахіднику не має творчого характеру і не відповідає поняттю "творчість", тому допомога такого роду не може вважатись за творче співавторство; а запозичення, перенесення і використання чужого досвіду не вважають за творчість.

Однаковими виявляються підходи обох інститутів в питаннях захисту авторських прав громадян України на території іноземних держав, визначення дієздатності суб'єктів творчого процесу, можливість держави бути суб'єктом зазначених прав.

Але при вирішенні питання про доцільність застосування норм права промислової власності до відносин у сфері створення і використання інформаційних систем не слід забувати й про значні відмінності цих інститутів, про які йшлось при дослідженні інформаційних систем як об'єкту цивільно-правового регулювання. Значна кількість відмінностей знаходиться у сфері прав і обов'язків суб'єктів цих відносин, у процедурі оформлення прав на винаходи, яка докорінно відрізняється від авторсько-правової, а не в самому суб'єктному складі.

7.3.9. Держава як суб'єкт авторських прав

Слід зазначити, що держава може стати суб'єктом зазначених прав на програми, бази даних, інформаційні системи у чітко визначених законом випадках. Держава є суб'єктом первісного права у випадку примусового викупу авторського права у автора та його спадкоємців. Такий викуп припускається у випадку суспільного інтересу до використання твору, його охорони, що не забезпечує або не може забезпечити сам автор (автори) або його спадкоємці. Для викупу в кожному випадку приймається постанова Уряду.

Особливе право держава набуває на авторські твори, які проголошено її здобутком. Це право не може бути віднесене до первісних, оскільки твір не тільки створений заздалегідь до оголошення його здобутком держави (інакше він міг бути лише викупленим у автора чи його спадкоємців), але й термін дії авторського права по відношенню до нього закінчився. Це право не може бути віднесеним і до похідного права, оскільки тут немає ніякого правонаступництва. Зміст проголошення твору здобутком держави полягає в тому, що цим підкреслюється віднесення твору до переліку видатних, епохальних творів людського генію. Органи держави забезпечують особливу охорону їх недоторканності, видання та відтворення. Крім того, держава може визначати долю доходів, що отримуються від їх використання.

7.3.10. Користувач або споживач інформації

Не слід забувати, що сьогодні йдеться про те, що інформаційна система є не лише продуктом інтелектуальної діяльності, але і продуктом матеріального виробництва і може безпосередньо використовуватись в процесі ринкових відносин.

Особливе місце серед учасників відносин у сфері використання інформаційних систем слід відвести користувачу або споживачу інформації, який звертається до інформаційної системи чи до посередника з метою отримання необхідної йому інформації і користується нею, а також користувачу інформаційною системою, технології та засобів її забезпечення.

Користувачами інформаційної системи, технології і засобів їх забезпечення можуть виступати фізичні та юридичні особи, держава. Користувачем інформаційної системи сьогодні виступає і державне підприємство, що здійснює право користування як елемент права повного господарського відання, і державна установа через оперативне право управління.

На підставі викладеного вище можна зробити наступні висновки:

- в умовах розвитку ринкових відносин в Україні з'явилась і продовжує розвиватись тенденція відокремлення інформаційних суспільних відносин, які на теперішньому етапі потребують детального правового регулювання;
- суб'єкти авторського права у сфері використання програм, баз даних, інформаційних систем розрізняються як суб'єкти первісного й похідного права; до перших відносяться автори і співавтори, причому співавторство є більш характерним при створенні інформаційних систем і повинно ґрунтуватись на угоді між співавторами; до других — ті, хто отримують право на твір за волевиявленням авторів;
- охорона інформаційних систем авторським правом потребує розробити спеціальний закон, в якому слід врегулювати всі питання, пов'язані з суб'єктами авторського права на інформаційні системи, їх правами і обов'язками;
- програми, бази даних, інформаційна система як продукти матеріального виробництва можуть безпосередньо використовуватися у процесі ринкових відносин. Всі положення, які характеризують цей бік проблеми слід об'єднати в єдиний закон, визначивши суб'єктний склад відносин власності, володіння і користування у сфері використання інформаційних систем без урахування відносин, які повинні регулюватися Законом України "Про авторське право і суміжні права".

7.4. Права та обов'язки учасників інформаційних відносин

7.4.1. Зміст інформаційних відносин

Досліджуючи суб'єктний склад відносин у сфері використання програм, баз даних, інформаційних систем, зверталась увага на те, що наука цивільного права з метою індивідуалізації окремих цивільно-правових відносин окреслює їх елементи: суб'єкти і об'єкти; суб'єктивне цивільне право і суб'єктивний цивільний обов'язок. Таким

чином, змістом інформаційних, як і будь-яких інших правовідносин, є взаємні права і обов'язки їх учасників.

Не слід забувати, що у цивільно-правових відносинах повинні брати участь принаймні дві особи, бо це відносини між людьми. Особи, які є учасниками цивільно-правових відносин у сфері використання програм, баз даних та інформаційних систем, мають суб'єктивні права і обов'язки.

Важливо зазначити, що суб'єкт цивільно-правових відносин, якому належить право, називається ***активним суб'єктом***, або суб'єктом права, а суб'єкт цивільних відносин, на якого покладено обов'язок, називається ***пасивним суб'єктом***, або суб'єктом обов'язку.

Загальна теорія цивільного права визначає, що таких цивільно-правових відносин, в яких існує лише суб'єкт права і лише суб'єкт обов'язку, дуже мало і в цивільно-правовому відношенні, як правило, кожен з учасників має суб'єктивні права і несе суб'єктивні обов'язки.

7.4.2. Підстави виникнення цивільних правовідносин у сфері інформаційних систем

Цивільно-правові норми самі по собі не породжують цивільних правовідносин, цивільних прав і обов'язків, які входять до її складу. Необхідним є виникнення конкретних суспільних відносин за таких обставин, з якими закон пов'язує настання правових наслідків.

У законодавстві дається перелік основних юридичних фактів, з якими цивільне законодавство пов'язує виникнення цивільних прав і обов'язків, а юридичними фактами називаються обставини, з настанням яких норми права пов'язують виникнення, зміну або припинення правовідносин.

До них, крім багатьох інших, відносяться і створення творів науки, літератури і мистецтва, винаходів та ін., що має безпосереднє відношення до об'єкту, що досліджується.

7.4.3. Суб'єктивне авторське право у сфері інформаційних відносин

З моменту створення інформаційної системи її автор або автори набувають суб'єктивні немайнові та майнові права. Поняття й сутність суб'єктивних авторських прав базуються на єдиному розумінні суб'єктивного права як передбаченої законом можливості певної поведінки уповноваженого суб'єкта.

Відповідно до цього суб'єктивне авторське право характеризується як забезпечена нормами права можливість самому суб'єкту здійснювати різноманітні дії, вимагати відповідної поведінки від зобов'язаних осіб і звертатись у відповідні органи за захистом порушених прав і інтересів. Тобто, суб'єктивне авторське право — це комплекс немайнових і майнових прав, які мають відносно самостійне призначення, хоча і тісно пов'язані між собою. Зобов'язані особи не повинні перешкоджати здійсненню прав авторів.

Все перелічене повною мірою можна віднести й до суб'єктів відносин у сфері створення і використання програм, баз даних та інформаційних систем.

7.4.4. Умови визнання авторського права і сповіщення про авторське право

Для виникнення, визнання і здійснення авторського права на програми, бази даних та інформаційні системи не вимагається виконання будь-яких формальностей, включаючи депонування або реєстрацію.

Особа, яка має авторське право, для сповіщення про свої права може, починаючи з першого випуску в світ програми, бази даних чи інформаційної системи повинна використати знак охорони авторського права, який розташовується на кожному примірнику інформаційної системи і складається з трьох елементів: © — літери С у колі, імені особи, яка має авторське право і року першого випуску інформаційної системи в світ.

Необхідно передбачити варіант, коли інформаційна система складається з матеріалів, які не є об'єктами авторського права. В цьому випадку авторське право на таку інформаційну систему належить особам, які її створили.

Авторське право на програму, базу даних, інформаційну систему:

- визнається за умови дотримання авторського права на кожен з творів, що включено до інформаційної системи;
- зберігається на кожен з творів, які включено до інформаційної системи за умови можливого використання незалежно від такої інформаційної системи.

Дуже важливим є надання можливості забезпечення неперешкоджання іншим особам здійснювати самостійний добір, організацію творів і матеріалів, які входять до інформаційних систем, навіть коли на цю інформаційну систему існує авторське право.

Особі, яка має авторське право, або будь-яку виключну правомочність на програму, базу даних чи інформаційну систему, необхідно надати можливість зареєструвати це право або правомочність в офіційних державних реєстрах самостійно або через представників протягом терміну охорони авторського права.

Державній реєстрації можуть підлягати свідчення про авторство на обнародуваний чи необнародуваний твір, факт і дата опублікування твору та договори, які зачіпають права автора на твір.

У Цивільному кодексі України передбачається, що орган, який забезпечує державну реєстрацію і положення про державну реєстрацію прав автора на витвори науки, літератури і мистецтва затверджує Кабінет Міністрів України. Для інформаційної системи таким органом може бути Українське агентство з правової охорони програм для ЕОМ, баз даних, топологій інтегральних мікросхем та інформаційних систем.

Про реєстрацію прав автора інформаційної системи необхідно видавати свідоцтво, а при виникненні суперечки державна реєстрація має визнаватися судом як юридична презумпція авторства, тобто має вважатись дійсною, якщо в судовому порядку не буде доведено іншого.

Поширення на інформаційні системи принципів авторсько-правової охорони означає визнання за автором (авторами) особистих немайнових і майнових прав.

7.4.5. Особисті немайнові права автора

Закон відносить до особистих (немайнових) прав автора право авторства, право на авторське ім'я, право на недоторканність твору та право на обнародування твору.

Конкретизуючи раніше наведені положення, зазначимо, що автор є суб'єктом абсолютних правовідносин, за якими всі оточуючі особи зобов'язані не порушувати його право. В свою чергу, вступаючи у правовідносини по використанню інформаційних систем, автор (автори) набувають відносні майнові права.

Авторське право більшості зарубіжних країн поділяється на ряд повноважень майнового і особистого характеру. Першими за їх виникненням слід називати немайнові, або моральні права: право авторства, право на ім'я, на назву твору і його недоторканність. Однак основне значення мають виключні права на використання твору і право на гонорар. У зв'язку з цим підкреслимо, що в Україні сьогодні особисті (немайнові) права автора складаються з прав авторства, права на авторське ім'я, право на недоторканність твору і права на обнародування твору.

Право авторства на програми, бази даних та інформаційні системи

Право авторства на програми, бази даних та інформаційні системи, як і на будь-який інший об'єкт авторського права, є забезпечена законом можливість особи вважатись автором даного твору і вимагати від інших осіб виконувати вимоги авторського права (визнавати права авторства) відповідно до існуючого законодавства.

Назване право може належати лише творцю програми, бази даних та інформаційної системи, а не будь-кому іншому і закріплює той факт, що дана особа своєю творчою працею створила конкретний твір. Його виникнення не залежить від того, чи опубліковано твір і чи є підстави для отримання винагороди. Право авторства не відчужується і не передається за договором, воно не переходить і у спадщину, тобто є безстроковим. Воно не підлягає майновій оцінці і є невід'ємним від автора правом.

Не слід забувати, що право авторства характеризується як основа для виникнення всіх інших правомочностей автора.

Право співавторства на програми, бази даних та інформаційні системи

Для створення програми, бази даних та інформаційної системи на сьогодні більш характерним є наявність співавторів, яким авторське право повинно належати сумісно, а взаємодія співавторів з приводу створеної інформаційної системи може визначатись договором між ними. Кожний із співавторів зберігає авторське право на створену ним частину, якщо вона має самостійне значення і вправі розпорядитись такою частиною за власним розсудом (наприклад, творці операційної системи, опису програм, допоміжних матеріалів, баз знань) і повинні користуватись всім комплексом відповідних правомочностей самостійно і незалежно один від одного.

Більш складним може виявитись вирішення питання про авторство, якщо одна інформаційна система — назвемо її базовою — надає можливість для створення цілого покоління інформаційних систем з широким діапазоном використання.

Творці вихідних інформаційних систем можуть докладати значних зусиль стосовно адаптації базової інформаційної системи, пристосування її для вирішення комплексу нових проблем і виконання нових функцій.

Зрозуміло, що в цих умовах не можна позбавити творців базової інформаційної системи відповідних прав, оскільки нові інформаційні системи суттєво залежать від першої.

Право на авторське ім'я (назву)

Зазначення імені автора при використанні твору обов'язкове в усіх випадках. Існують винятки, хоча до програми, бази даних та інформаційної системи вони навряд чи можуть мати відношення.

З правом авторства нерозривно пов'язане право на авторське ім'я, яке по суті є конкретизацією права авторства і представляє собою забезпечену законом можливість опублікувати твір під своїм ім'ям, під умовним ім'ям (псевдонімом) або без зазначення імені (анонімно). Автор, який створив програму, базу даних, інформаційну систему, здійснює право на ім'я, випускаючи її у світ.

У нашому випадку без згоди автора ніхто не може вносити до програми, бази даних, інформаційної системи будь-які зміни у позначенні його імені. Можливим, на наш погляд, є варіант, коли група співавторів домовиться між собою випускати інформаційну систему без зазначення імен авторів, використовуючи спеціальну назву.

На основі переліченого, право на ім'я автора програми, бази даних, інформаційної системи можна тлумачити як право визначати форму зазначення імені автора: під своїм ім'ям, під умовним ім'ям (псевдонімом) або анонімно.

Право на недоторканність твору і контроль за його дотриманням

Право на недоторканність твору визначається як право протидіяти будь-якому перекрученню, спотворенню чи іншій зміні твору або будь-якому іншому посяганню на твір, що може зашкодити честі та репутації автора. Це означає, що при виданні або будь-якому іншому використанні твору, забороняється без дозволу автора (авторів) та його правонаступників вносити будь-які зміни як до самого твору, так і до його назви та позначення імені автора.

Не слід допускати без дозволу автора супроводження програми, бази даних, інформаційної системи передмовами, післямовами, коментарями, будь-якими іншими поясненнями, скороченнями чи доповненнями твору.

Таким чином, право на недоторканність (цілісність) програми, бази даних, інформаційної системи слід визначити як право на захист як самих цих понять, так і їх назви від всілякого роду перекручень, спотворень або інших посягань, що може завдавати шкоду честі та репутації автора. Наведене підтверджується пунктом 4 статті 13 Закону "Про авторське право і суміжні права".

Слід зазначити, що право на недоторканність програми, бази даних, інформаційної системи згідно закону є особистим немайновим правом, але спадкоємці також наділяються цим правом, а особи, які використовують твори, що стали суспільним надбанням, зобов'язані також дотримуватись вимог недоторканності твору.

За загальним правилом контроль за дотриманням вимог щодо недоторканності твору здійснює Державне агентство України з авторських і суміжних прав, але для комп'ютерного об'єкту, як вже наголошувалось, доцільно створити спеціальне Українське агентство з правової охорони програм для ЕОМ, баз даних, топологій інтегральних мікросхем, інформаційних (і експертних) систем. Необхідність появи такого спеціального агентства є зрозумілою і актуальною, і вона обґрунтовувалась при розгляді питання про суб'єктивні права автора інформаційної системи.

Право автора на недоторканність полягає і в тому, що переклад програми, бази даних, інформаційної системи з однієї мови на іншу з метою випуску у світ слід дозволяти лише за згодою автора або його правонаступників і на підставі договору. При цьому переклад може здійснюватись лише за умови збереження цілісності та змісту твору.

Право на обнародування твору

За Законом про авторське право ***обнародування твору*** — це дія, що надає твору доступності для публіки, якими б засобами це не досягалось. Інакше це істотне особисте немайнове право автора називається правом ***випуску твору у світ***. Відповідно до Закону програму, базу даних, інформаційну систему можна вважати випущеною у світ (опублікованою, обнародуваною), якщо вони будь-яким способом повідомлені невизначеному колу осіб.

У літературі справедливо відмічається, що твір може бути випущений у світ різними способами, але істотним є те, що його зміст повідомлений невизначеному колу осіб і те, що не вважається випуском у світ інформація про твір з викладенням його короткого змісту або повідомлення вузькому колу осіб. Ті самі положення знайшли своє відображення в Законі "Про авторське право і суміжні права".

Дещо іншу позицію має з цього приводу російське законодавство. Випуск у світ програми для ЕОМ або бази даних, наприклад, відноситься до майнових прав автора. Доцільність такого трактування відносно інформаційних систем необхідно ще детально проаналізувати, але така перспектива є досить обґрунтованою.

Оскільки за діючим законодавством, передусім автору чи іншій особі, що має авторське право, належить виключне право на використання твору в будь-якій формі і будь-яким способом, доцільним буде першочергово звернутись до аналізу самого поняття "виключне право".

Сучасна юридична думка характеризує ***виключне право***, як право, коли жодна особа, крім тієї, якій належить авторське право, не може використовувати твір, не маючи на те відповідного дозволу (ліцензії), за винятком випадків, встановлених Законом про авторське право. Аналогічну позицію в цьому питанні займали й інші автори, які обґрунтовували введення і використання цього поняття.

7.4.6. Виключні права автора та інших осіб, які мають авторське право

У Законі "Про авторське право і суміжні права" 1994 року до виключних прав автора або іншої особи, яка має авторське право, належить право дозволяти або забороняти:

- відтворення творів;

- публічне виконання і публічне сповіщення творів;
- публічний показ;
- будь-яке повторне публічне сповіщення в ефірі чи по дротах вже переданих в ефірі творів, якщо воно здійснюється іншою організацією;
- переклади творів;
- переробки, адаптації, аранжування та інші подібні зміни творів;
- розповсюдження творів шляхом продажу, відчуження іншим способом або шляхом здачі в найм чи у прокат та іншої передачі до першого продажу примірників твору;
- здачу в найм після першого продажу, відчуження іншим способом примірників аудіовізуальних творів, музичних творів у нотній формі, а також творів, зафіксованих на фонограмі або у формі, яку читає машина;
- імпорт примірників твору.

Більшою або меншою мірою всі перераховані виключні майнові права належать і автору програми, бази даних, інформаційної системи з урахуванням особливостей їх як об'єктів. Крім того, цей перелік не є вичерпним і може доповнюватись.

Найбільш суттєві для інформаційних систем виключні майнові права слід сформулювати наступним чином:

- відтворення інформаційної системи (повне або часткове) в будь-якій формі і будь-якими способами;
- розповсюдження інформаційної системи;
- модифікація інформаційної системи, в тому числі переведення інформаційної системи з однієї мови на іншу;
- право відтворення програми, бази даних, інформаційної системи.

Відтворення інформаційної системи — це виготовлення одного або більше примірників інформаційної системи в будь-якій матеріальній формі, а також запис інформаційної системи для тимчасового чи постійного зберігання в електронній (включаючи цифрову), оптичній або іншій формі, яку читає машина, де примірник — це результат будь-якого відтворення твору.

Здійсненню права публічного сповіщення у сфері створення інформаційної системи більше відповідає така передача в ефір або по дротах зображень і (або) звуків інформаційної системи, коли зазначені зображення чи звуки можуть бути сприйнятими невизначеним колом осіб.

Автор (автори) оригіналу інформаційних систем може сам перекласти свій твір іншою мовою, тобто здійснити авторський переклад. При цьому можливо збереження загального правила про те, що за наявності авторського перекладу ніхто інший не може перекладати цей твір тією ж мовою. Можливі тут й авторизовані переклади.

Модифікація програм, баз даних, інформаційних систем

Досить очевидним є необхідність віднести будь-яку модифікацію програми, бази даних, інформаційних систем (переробку твору) до виключних майнових прав автора. Така модифікація може мати місце лише за згодою автора чи його правонаступників.

Розповсюдження програм, баз даних та інформаційних систем

Особливо гостро постає питання щодо розповсюдження таких творів як програма, база даних, інформаційна система шляхом продажу, відчуження в інший спосіб або шляхом здавання в найом чи у прокат, іншої передачі їх до першого продажу.

Відповідно до Закону "Про авторське право і суміжні права" розповсюджувати чи будь-яким іншим способом поширювати до першого продажу свій твір може лише автор або його правонаступник чи інші особи, але тільки з дозволу автора чи його правонаступників. Проте, після першого продажу програми, бази даних, інформаційної системи згідно з даним правилом право автора чи його правонаступників на розповсюдження твору вичерпується. Щодо творів, зафіксованих у формі, яку читає машина, це право зберігається за автором чи їх правонаступниками також після першого продажу чи відчуження в інший спосіб.

Практичну значимість це набуває у зв'язку з тим, що програму, базу даних, інформаційну систему після її першого продажу може здавати в найм власник твору.

Як передбачається Законом "Про авторське право та суміжні права", зазначений перелік не є вичерпним, тому автор програми, бази даних, інформаційної системи має право дозволяти або забороняти використовувати свій твір в інші способи.

7.4.7. Передача майнових прав за договором

У законодавстві, присвяченому правовій охороні прав на програму, базу даних, інформаційну систему, так само як для програм та баз даних, передбачена можливість передачі майнових прав на інформаційні системи повністю або частково іншим фізичним або юридичним особам за договором.

Майнові права можуть відчужуватись автором та іншими суб'єктами авторського права (ліцензіаром). Ліцензіар може видавати ліцензію іншій особі (ліцензіату) на використання творів відповідно до такої ліцензії. Відчуження авторських майнових прав і видача ліцензії оформляється на підставі авторського договору. Особисті немайнові права автора при цьому не можуть передаватись іншим особам.

Договір повинен укладатись в письмовій формі, і має встановлювати наступні суттєві умови:

- обсяг і способи використання інформаційної системи;
- порядок виплати і розмір винагороди;
- термін дії договору.

Важливим є той факт, що майнові права на інформаційну систему можуть переходити у спадок у встановленому законом порядку.

Що стосується майнових прав на інформаційні системи, створені в порядку виконання службових обов'язків або за завданням роботодавця, то вони будуть належати роботодавцю, якщо в договорі між ним і автором не передбачено іншого. Порядок оплати і розмір винагороди встановлюється договором між автором і роботодавцем.

7.4.8. Вільне використання програм, баз даних та інформаційних систем

Разом з тим, велику практичну значимість має те, що Закон в деяких випадках обмежує виключне право автора на використання твору, а використання твору без згоди автора отримало назву ***вільного використання***. Можливість і доцільність вільного використання творів без згоди автора і без виплати йому авторської винагороди або з виплатою йому авторської винагороди з урахуванням певних особливостей слід передбачити й для інформаційних систем.

Зразком і підставою для цього може слугувати детально регламентоване вільне відтворення комп'ютерних програм (стаття 18 Закону "Про авторське право і суміжні права").

Без дозволу автора чи іншої особи, якій належить авторське право на інформаційну систему, якщо інше не визначено в договорі, дозволяється здійснювати наступні дії:

- відтворення одного примірника інформаційної системи, а також адаптацію інформаційних систем, які здійснюються особою, яка є законним власником примірника інформаційної системи;
- виготовлення або доручення виготовляти копії інформаційної системи за умови, що ця інформаційна система призначена лише для архівних цілей та для заміни законно придбаного примірника інформаційної системи у випадку її втрати, пошкодження або непридатності.

Безсумнівно, що визначений примірник інформаційної системи або її адаптація не можуть використовуватись з іншою метою, ніж передбачено, і підлягає знищенню у випадку, коли продовження володіння інформаційною системою втрачає законний характер.

Іншою підставою може бути відтворення в одному примірнику для архівних цілей і заміна законно придбаного примірника інформаційної системи у випадку, коли такий примірник втрачено, пошкоджено або він став непридатним.

Подальший аналіз цих питань дозволяє дійти висновку, що автор або інша особа, якій належить право на використання примірника інформаційної системи, може без отримання дозволу особи, якій належить право власності на інформаційну систему, здійснювати будь-які дії, пов'язані з функціонуванням інформаційної системи відповідно до її призначення, в тому числі виправлення явних помилок. Запис і зберігання в пам'яті ЕОМ допускається по відношенню до однієї ЕОМ або одного користувача в мережі, якщо інше не передбачено договором з автором, або іншою особою, якій належить право власності на інформаційну систему.

7.4.9. Право на авторську винагороду — основне майнове право авторів

Як вже зазначалось, право на авторську винагороду — це основне майнове право автора чи іншої особи, що має авторське право. У зв'язку з цим підставою для винагороди є факт використання твору у будь-який спосіб.

Якщо вести розмову про основні правові форми використання творів, в тому числі й програм, баз даних, інформаційних систем, то їх можна охарактеризувати як виключне право дозволяти або забороняти ті чи інші дії, про які йшлося раніше.

До конкретних юридичних фактів, що породжують у автора чи іншої особи, яка має авторське право, право на винагороду, закон відносить:

- авторський договір;
- факт позадоговірного використання твору, коли не вимагається згода автора, але передбачена виплата авторської винагороди;
- неправомірне використання твору.

Тобто, право на одержання винагороди породжується, як правило, лише фактом використання твору. Таким чином, сам по собі факт наявності твору в об'єктивній формі права на винагороду породжує не завжди.

У літературі звертається увага на те, що винагорода, отримана автором чи іншою особою, яка має авторське право, є по суті винагородою за працю, вкладену у створення твору. Вона може бути як у формі заробітної плати, так й у формі авторського гонорару. Крім того, можливе поєднання цих форм оплати. Не слід забувати, що винагорода має виплачуватись за будь-яке використання твору, окрім згадуваних нами випадків використання твору без згоди автора.

Що ж стосується розміру й порядку обчислення авторської винагороди за створення і використання твору, то він визначається у відповідному авторському договорі і може здійснюватись у вигляді одноразового платежу — одноразова винагорода, або у формі відрахувань — відсотків за кожний проданий примірник чи кожне використання твору, або складатись зі змішаних платежів.

Все викладене повною мірою належить і до інформаційної системи як до об'єкту авторсько-правової охорони.

7.4.10. Договір між автором і користувачем

Велику практичну значимість при використанні програми, бази даних, інформаційних систем відіграє договір між автором (його спадкоємцем, а також будь-якою фізичною або юридичною особою, яка має виключні майнові права на програму, базу даних, інформаційну систему, отримані в силу закону або договору) і третіми особами (користувачами). Таке використання здійснюється на підставі договору, за виключенням випадків вільного перепродажу екземпляру інформаційної системи, який слід розглянути окремо.

Вказаний договір повинен укладатися в письмовій формі. При продажу і наданні масовим користувачам доступу до інформаційної системи необхідно передбачити особливий порядок укладання договорів, наприклад, шляхом викладення типових умов договору на екземплярах інформаційних систем, які передаються.

Разом з тим, слід передбачити вільний перепродаж екземпляру інформаційної системи, а саме допустити перепродаж або передачу іншим способом власності або інших речових прав на екземпляр інформаційної системи після першого продажу або іншої передачі права власності на цей екземпляр без згоди автора і без виплати йому додаткової винагороди.

7.5. Висновки щодо інформаційних систем

Таким чином, можна зробити певні висновки щодо інформаційних систем, а саме:

- з моменту створення інформаційної системи автор або автори набувають суб'єктивних немайнових і майнових прав; це повинно бути закріплено спеціальним законом, що повинен містити в собі ряд положень, основними з яких є наступні:
  - серед умов визнання авторського права на інформаційні системи можна назвати: авторське право на інформаційні системи, яке виникає у зв'язку з їх створенням (для виникнення, визнання і здійснення авторського права на інформаційні системи не вимагається виконання будь-яких формальностей, включаючи депонування або реєстрацію);
  - особа, яка має авторське право, для сповіщення про свої права може, починаючи з першого випуску у світ інформаційної системи, використати знак охорони авторського права, який має бути розташований на кожному примірнику інформаційної системи і складатися з трьох елементів: літери С у колі, імені автору (авторів) і року першого випуску інформаційної системи у світ;
  - у випадку, коли інформаційна система складається з матеріалів, які не є об'єктами авторського права, авторське право на таку інформаційну систему належить особам, які створили цю інформаційну систему;
  - авторське право на інформаційну систему визнається за умови дотримання авторського права на кожен з творів, включених до цієї інформаційної системи і зберігається на кожен з творів, які включені в інформаційну систему за умови можливості використання незалежно від такої інформаційної системи;
  - необхідно надати можливість і не перешкоджати іншим особам здійснювати самостійний підбір і організацію творів і матеріалів, які входять до інформаційних систем, навіть коли на цю інформаційну систему існує авторське право;
  - особі, яка має авторське право, або будь-яку виключну правомочність на інформаційну систему, необхідно надати можливість зареєструвати її в офіційних державних реєстрах самостійно або через представників протягом терміну охорони авторського права;
- для інформаційних систем слід передбачити створення спеціального органу, який має забезпечувати їх державну реєстрацію; таким органом може стати спеціальне Українське агентство з правової охорони програм для ЕОМ, баз даних, топологій інтегральних мікросхем та інформаційних систем;
- найбільш суттєвими для авторів інформаційних систем є виключні майнові права, які слід сформулювати наступним чином:
  - відтворення інформаційної системи (повне або часткове) в будь-якій формі і будь-якими способами;
  - розповсюдження інформаційної системи;
  - модифікація інформаційної системи, в тому числі переклад інформаційної системи з однієї мови на іншу;

- у законодавстві, яке передбачає правову охорону прав на інформаційну систему, слід передбачити можливість передачі майнових прав на інформаційні системи повністю або частково іншим фізичним або юридичним особам за договором; майнові права можуть відчужуватись автором та іншими суб'єктами авторського права (ліцензіаром); ліцензіар може видавати ліцензію іншій особі (ліцензіату) на використання інформаційної системи відповідно до такої ліцензії;
- практичну значимість при використанні інформаційних систем відіграє договір між автором і третіми особами (користувачами). Таке використання повинне здійснюватись на підставі договору, який слід укладати в письмовій формі. При продажу й наданні масовим користувачам доступу до інформаційної системи необхідно передбачити особливий порядок укладання договорів, наприклад, шляхом викладення типових умов договору на екземплярах інформаційних систем, які передаються.

7.6. Проблеми Internet за кордоном

Слід зазначити, що з моменту виникнення і створення глобальної мережі Internet США більше уваги приділяє проблемам, які стосуються саме Internet. Кількість постійних користувачів мережі у світі перевищило 400 млн. Безумовним лідером є США (біля 154 млн постійних користувачів Internet, або 76 відсотків від усього дорослого населення). Країни Західної Європи прикладають значних зусиль, щоб наздогнати США. Вже сьогодні більше 113 млн. європейців є постійними користувачами Internet, що складає майже 36 відсотків від усього дорослого населення. Загальні темпи приросту кількості хостів в Європі складає 23,8 відсотків.

Однак, не слід забувати, що глобальна мережа, як і інші регіональні, включає в себе велику кількість інформаційних систем, які в поєднанні з засобами передачі складають мережі. Тому всі питання, які вирішуються в Internet слід вирішувати, перш за все, для інформаційних систем.

Регулювання питань, пов'язаних з інформаційним наповненням Internet та правовими аспектами функціонування, — це складна і делікатна справа. Країн, в яких є спеціальне законодавство, що стосується Internet, в Європі небагато. Однією з таких країн є Німеччина. Німецькі законодавці, наприклад, у червні 1997 року прийняли спеціальний закон про мультимедіа. Цим законом, наприклад, передбачається контролювання матеріалів, які розміщуються в Internet.

У липні 1997 року на засіданні Ради міністрів країн Європейського Союзу (ЄС) в Бонні була прийнята Декларація про принципи використання Internet. В ній, зокрема, записано, що Internet-провайдери в цілому не відповідають за інформацію, яку одержують їх клієнти з Internet. Крім того, було схвалено принцип недискримінаційного оподаткування при використанні Internet в електронній торгівлі. Але Internet проголошений "вільною зоною торгівлі", як це пропонувалось США.

Цікавим є те, що офіційними органами США з одного боку підкреслюється необхідність обмеження державного втручання в Internet, чим пояснюється його бурхливий розвиток, а з іншого боку, Конгрес США вже прийняв законопроекти, присвячені Internet (бібліотека Конгресу — `http://thomas.loc.gov`). Закон про захист персо-

нальної інформації в Internet передбачає введення заборони для федеральних відомств на надання доступу до конфіденційних записів про індивідів через Internet.

Закон про сімейний доступ до Internet спрямований на забезпечення батьків засобами контролю за змістом, до якого мають доступ їх діти через Internet. Провайдери послуг Internet повинні надавати програмне забезпечення, яке дозволяє здійснювати такий контроль. Заборона на використання Internet для азартних ігор, укладання парі, проведення лотерей та інших засобів вилучення грошей у населення передбачає закон "Про заборону на азартні ігри в Internet"

7.6.1. Проблеми конфіденційності у сфері інформаційних відносин

Багато уваги США приділяють питанням конфіденційності, яку асоціюють з концепцією особистої свободи. Легкий доступ до інформації в інформаційних системах і в глобальній інформаційній інфраструктурі (GII — Global Information Infrastructure) в цілому, можливість її багаторазового використання і миттєвої передачі можуть призвести до зниження рівня конфіденційності. Тому однією з необхідних умов успішного ведення бізнесу вважається забезпечення конфіденційності.

У червні 1995 року Робоча група з питань конфіденційності у складі Цільової групи з питань інформаційної інфраструктури уряду США (IITE) опублікувала звіт під назвою "Конфіденційність і національна інформаційна інфраструктура: принципи надання і використання особистої інформації". У звіті наведено ряд принципів, які регламентують збір, обробку, зберігання і повторне використання особистих даних. Ці принципи, які базуються на основі розроблених Організацією економічного співробітництва і розвитку "Основних принципів захисту конфіденційності та міжнародного потоку персональних даних", об'єднують принципи доброчинної практики в роботі з інформацією. Крім того, принципи конфіденційності визначають три умови, які регулюють отримання, розголошення і використання конфіденційної інформації, її цілісність і якість.

У квітні 1997 року Комітет з інформаційної політики Цільової групи IITE видав проект документу "Варіанти забезпечення конфіденційності в системі Національної інфраструктури". В цьому документі аналізується інформаційна практика США і громадська думка, які стосуються конфіденційності інформації. Метою є пошук оптимального балансу між розумінням особистої конфіденційності і вільним потоком інформації в демократичному електронному суспільстві.

Доречно буде відмітити, що питання конфіденційності, як вони загалом називаються у США, розглядаються у багатьох країнах, в тому числі й в Україні. Деякі країни прийняли закони, які вводять принцип саморегулювання в промисловості або передбачили адміністративні заходи по захисту прав конфіденційності своїх громадян.

Однак, на думку урядових кіл США, відмінності цих законів можуть призвести до порушення міжнародних потоків даних. Так, ЄС прийняв директиву, яка забороняє передачу особистих даних в країни, які, за його думкою, не приймають належних заходів по захисту конфіденційності громадян країн ЄС.

Центральною проблемою в США вважається безпека і надійність мережі GII. В офіційних джерелах наголошується, що користувачі Internet повинні бути впевнені,

що їх дані захищені від несанкціонованого доступу, інакше вони не будуть користуватись мережею з комерційною метою. Безпека мереж GII, на їх думку, потребує:

- безпечних і надійних мереж телекомунікацій;
- ефективних засобів захисту інформаційних систем, які підключені до цих мереж;
- ефективних засобів ідентифікації і забезпечення конфіденційності електронної інформації для захисту даних від несанкціонованого доступу;
- добре підготовлених користувачів мережі GII, які розуміють як захистити свої системи і свої дані.

Особливо важливою є розробка надійних процедур сертифікації, які підтримують передачу підписів по комп'ютерним каналам, що дозволить користувачам знати, з ким вони спілкуються в мережі.

Для створення електронної комерції у США підтримується розвиток добровільної, ринково-орієнтованої інфраструктури управління криптографічних ключів, які забезпечили б ідентифікацію, цілісність і конфіденційність інформації. Велику надію в США викликає приватний сектор, ведуча позиція якого здатна призвести до стрімкого розвитку інформаційних систем, різнотипових мереж і Internet. Якщо приватний сектор і уряд будуть діяти узгоджено, це відкриє величезні можливості для комерційної діяльності.

Підводячи підсумки, слід звернути увагу на те, що переважною формою правової охорони програмного забезпечення за кордоном є авторсько-правова охорона. Практично у всіх зарубіжних країнах програми патентуються лише у виключних випадках. При цьому можливі варіанти одночасної охорони програм ЕОМ нормами як патентного, так і авторського права. Іншими словами, в охороні цього об'єкту має місце "множинність охорони". У процесі формування правової охорони програмного забезпечення ЕОМ за кордоном умовно можна виділити два етапи:

- перший етап — до середини 70-х років, пов'язаний головним чином з використанням переваг патентного законодавства;
- другий етап —з середини 70-х років — в основному з авторським правом.

У літературі констатується, що самі по собі програми ЕОМ є непатентоздатними. Однак їх комбінація, наприклад, з промисловим способом або з машиною, в якій використаний такий спосіб, може бути визнана патентоздатною. При цьому патентна охорона розповсюджується на суть програми ЕОМ, забезпечує виключне право патентоволодільця на використання і продаж програми протягом певного періоду часу.

Основним способом забезпечення інтересів осіб і організацій, які створюють програмний продукт, виступають конкретні угоди (ліцензійні договори), відповідно до яких передається право на використання об'єкту, що охороняється, на відповідних, чітко встановлених умовах, з дотриманням, як правило, принципу таємності змісту угоди.

За об'єкти правової охорони в більшості високорозвинутих країн (США, Японія, Великобританія та ін.) розглядаються програми для ЕОМ, а в деяких країнах і бази даних. Таким чином, ця тенденція притаманна не лише національному законодавству зарубіжних країн, але й міжнародній практиці правової охорони програмного забезпечення ЕОМ.

7.7. Розвиток законодавства України у сфері інформаційних відносин

7.7.1. Розвиток законодавства України другої половини XX століття у сфері інформатизації

Друга половина XX століття характеризується поширенням масштабів наукових досліджень, утворенням засобів електронно-обчислювальної техніки, прогресивним ростом обсягу нової інформації. Виявилось, що останнє десятиріччя XX століття породило стільки інформації, скільки накопичилось за попередні 2000 років. Цілком зрозуміло, що жодній людині не під силу пізнати всі опубліковані факти, що стосуються тієї чи іншої проблеми. Це може призвести до дублювання досліджень і розробок, втрати часу, невиправданих економічних витрат.

В літературі звертається увага на те, що природні ресурси України чималі, навіть за світовими масштабами. Проте входження країни до загальноєвропейського світового процесу лише за рахунок природних ресурсів сьогодні практично неможливий. Тому необхідно вирішити багато важливих завдань для того, щоб досягти високого соціально-економічного рівня конкурентоздатності інформаційної продукції. Серед найбільш актуальних — усвідомлення значення систематизації та розробки способів економічно вигідної реалізації інформаційних ресурсів, а також форм, методів і засобів регулювання інформаційного потоку, спрямованого з-за кордону і за кордон.

Відсутність координації інформаційних ресурсів і механізму включення до господарського обігу призводить до економічних втрат. Сьогодні є підстави стверджувати, що у світі склався ринок науковомісткої інформації, ефективний обмін якої дає змогу різним країнам вирішувати завдання розвитку на основі міжнародної спеціалізації, купівлі-продажу інформації. Для виходу України на цей ринок і поліпшення структури як експорту, так і імпорту необхідно здійснити ряд заходів як законодавчого, так і централізованого організаційного характеру. В першу чергу це стосується стимулювання інформаційного експорту через митні тарифи і ліцензування, механізм пільг та полегшений його порядок.

7.7.2. Першочергові завдання створення інформаційного ринку в Україні

Не слід очікувати, що інформаційний ринок утвориться самостійно і сам виробить цивілізовані правила взаємовідносин партнерів. Країна, перебуваючи у складному економічному стані, не може дозволити собі і надалі втрачати інформаційний ресурс, який безумовно є головним засобом розвитку будь-якої країни.

Зазначене вище свідчить про необхідність у вирішенні першочергових проблем:

- формування законодавчої і нормативної бази, яка б регламентувала експортно-імпортний інформаційний обмін на комерційній підставі;
- розробки гнучкого механізму цивілізованої реалізації прав фізичних та юридичних осіб на інформаційні ресурси внутрішнього ринку;
- розробки критеріїв та методів оцінки інформаційного ресурсу;
- розробки і реалізації системи організаційних заходів до охорони і захисту інформаційних систем.

Розв'язання зазначених проблем є вирішенням комплексного завдання щодо розробки та реалізації регулювання ринку інформаційних ресурсів.

На думку фахівців, XXI сторіччя — це сторіччя індустрії інформації та активного переходу країн до безпаперової технології. Головним товаром стане інформація. Розвиток інформаційних систем є закономірним наслідком науково-технічного прогресу, розширенням сфери інтелектуальної діяльності людства.

У зв'язку з цим слід зазначити, що традиційний підхід до науково-технічного прогресу взагалі орієнтований на послідовність: наука–техніка–виробництво. Наука породжує ідеї, техніка є їх матеріальним втіленням, а виробництво становить сферу промислової реалізації науково-технічних досягнень в реальний продукт. Таким чином, кожна з цих ланок є відносно самостійною і виконує специфічну роль.

Зрозуміло, що провідна роль в організації творчої діяльності вчених, авторів і винахідників, а також у визначенні напрямів досліджень і забезпеченні науково-технічного прогресу належить державі. Саме на державу покладається завдання щодо створення найсприятливіші умови для розвитку науки, техніки, виробництва шляхом розробки відповідної економічної і правової бази.

Не слід забувати і те, що єдина загальнодержавна система керування науково - технічним прогресом не змогла спрогнозувати заздалегідь соціально-економічного стану, в якому опинилася зараз країна. Між тим, нове знання виникає внаслідок протиріч, головним з яких є конфлікт між теорією і практикою, що означає наявність суперечностей, а це в свою чергу викликає настання кризи. Зрозуміло, що за цих умов потрібна інша система поглядів, новий стиль мислення, інший підхід.

Монополія державних підприємств, яка існувала до 90-х років винищила конкуренцію, а відсутність внутрішнього ринку на промислові вироби призвела до того, що нові та перспективні результати інтелектуальної діяльності, особливо перспективні технології, які вважають самим вигідним і престижним продуктом світу, були позбавлені статусу товару.

Що стосується продажу технологій зарубіжним партнерам, то слід зазначити, що за 30 років Україна продала лише 4 тис. ліцензій, в той час як Японія продає в середньому 2 тис. і одержує понад 50 млн ієн щорічно.

Повільне впровадження ринкових принципів у сфері науки і техніки спричиняє знецінення інтелектуальної творчої праці. На думку більшості авторів, єдиний, поступовий, взаємообумовлений їх розвиток можливий лише тоді, коли саме сфера товарного обміну, тобто ринок, буде мати в них потребу.

7.7.3. Поява нових видів наукової діяльності у сфері інформаційних відносин

Вочевидь, що накопичення знань в інформаційній сфері, поширення спеціалізації породжує новий вид наукової діяльності — інформатику. На думку деяких авторів, на сьогодні обмін інформацією здійснюється переважно за принципом вільного ринку: інформація, науково-технічні дані характеризуються як товар. Однак, в останній час стала помітною тенденція до розгляду інформатики як міжнародного ресурсу. Розробка нових принципів у даній сфері могла б стати важливим внеском до створення нового комплексу правових норм.

Інформатика виконує роль засобу забезпечення суспільних потреб. Проте специфічність її теоретичних основ, що застосовуються на принципі єдиного підходу до різноманітних процесів суміжних наук, правові аспекти у досягненні мети гарантованого забезпечення розвитку інформаційних відносин в інтересах суспільних потреб, на думку ряду зарубіжних авторів дозволяє вже сьогодні говорити про нову наукову дисципліну — право інформатики.

Разом з тим, розвиток будь-якої наукової дисципліни передбачає її автономність, самостійність функціонування. Самостійності має відповідати ряд цілком певних вимог, найважливіші з яких — реальність предмету і вчення, практична потреба, вияв властивих закономірностей тощо. У зв'язку з цим, можна зробити наступні висновки:

- сучасна діяльність у галузі правової інформатики повинна забезпечити вирішення наступних важливих питань:
  - утворення правової інформатики як науки, яка б органічно і гармонійно відповідала соціально-економічному стану країни на певному історичному етапі;
  - науковий розвиток правової інформатики в інтересах розвитку творчості людей і економічних інтересів суспільства.

  Якщо перше завдання якоюсь мірою вирішується з використанням емпіричних підходів, то практичні завдання можуть бути вирішені успішно лише на основі науково-теоретичного підходу, враховуючи різноманітні точки зору науковців, численні соціально-економічні фактори шляхом глибокого їх вивчення, співвідношення та узагальнення;
- у праві інформатики не можна лише накопичувати факти — факти, взяті самі по собі, ще не є наукою, і лише за допомогою мислення, відволікаючись від штучних зв'язків, можна з'ясувати глибинну послідовність і закономірності; на цій основі слід будувати нову правову дисципліну — право інформатики;
- друге завдання слід вирішувати, виходячи з того, що будь-яка теорія спочатку має характер чисто описовий, тобто вивчає існуюче. Після цього етапу слід переходити до порівняння, і вже потім теорія набуває ознак пояснювальної.

Право інформатики — не виняток. На цей час право інформатики володіє безліччю фактів, які є важливими не тільки у формі інформації, чи порівняння окремих явищ, головне — їх узагальнення мусить переслідувати мету знаходження подальших перспектив, оптимальної побудови системи права інформатики як в цілому, так і її невід'ємних елементів.

7.7.4. Можливість розвитку правової інформатики в XXI столітті

З початком нового тисячоліття стає можливим започаткування нової галузі — правової інформатики, невід'ємною складовою якої буде розробка питань у сфері використання інформаційних систем.

Серед найважливіших теоретичних питань, як зазначають фахівці, сьогодні є питання місцезнаходження відповідного нормативного масиву в системі законодавства України і його внутрішня будова. Сьогодні можна говорити про існування у національ-

ній правовій системі окремої самостійної галузі — інформаційного законодавства. Стає можливим вести розмову про офіційне визнання вищеозначеної галузі у випадку прийняття на державному рівні блоку нормативних актів, про які йшлося вище.

Ще один напрямок включає питання, які безпосередньо пов'язані не лише із створенням, а й з використанням інформаційних систем, відносин, які при цьому виникають. Його успішне розв'язання дає змогу налагодити договірні стосунки між суб'єктами даних відносин згідно з нормами цивільного законодавства.

При цьому треба враховувати суперечність між користувачами інформації, що прагнуть до безмежності та неврегульованості інформаційних потоків, і необхідність введення законодавчо визнаних обмежень на розповсюдження інформації з урахуванням загальнодержавних інтересів.

7.7.5. Розвиток інформаційних систем в Україні

Подальший розвиток мають отримати різного роду інформаційні системи. Така необхідність визнана на самому високому рівні. Для наукового забезпечення діяльності Президії та постійних комісій Верховної Ради України, наприклад, створено науково-експертний відділ з фахівців різних галузей знань. Цей відділ разом з відповідним сектором у складі Центру комп'ютеризованих інформаційних систем склали інформаційно-аналітичний центр Верховної Ради України.

Перспективи державної підтримки розвитку інформаційних систем в Україні викладено в Законі України "Про затвердження Завдань Національної програми інформатизації на 1998–2000 роки" від 4 лютого 1998 року. Крім загальних завдань у сфері інформатизації, таких, як розробка пакету законодавчих та нормативно-правових актів у цій сфері, за мету ставиться:

- розробка основоположних державних стандартів засобів інформатизації та інформаційно-телекомунікаційних систем, гармонізованих з міжнародними стандартами;
- розробка і впровадження системи державної сертифікації засобів інформатизації та інформаційно-телекомунікаційних систем у межах державної системи сертифікації УкрСЕПРО;
- вироблення та розвиток телекомунікаційної інфраструктури;
- створення національної системи інформаційних ресурсів;
- створення системи інформаційно-аналітичного забезпечення найважливіших галузей економіки, державних органів, навчальних закладів.

Конкретні заходи фінансування перерахованих завдань, безумовно, сприятимуть розвитку інформаційних систем і мереж в Україні. Про увагу, яка приділяється проблемам розвитку інформаційних систем, свідчать чисельні конференції та наукові семінари, що присвячені цій проблемі. Важливою серед них можна назвати конференцію, проведену за Постановою Президії Верховної Ради України "Інформаційна система Верховної Ради України та міжнародні інформаційні ресурси".

За роки існування України як суверенної держави з багатьма країнами були укладені декларації про принципи та основні напрями розвитку інформаційних відносин з обов'язковими положеннями про необхідність розвитку взаємовигідного співробітництва в науково-технічній та інформаційній сферах.

Так, Постановою Президії Верховної Ради України від 16 січня 1995 року було схвалено Договір про співробітництво між Президією Верховної Ради України та До-

слідницькою Службою Конгресу (ДСК) США з наступних питань програми парламентської допомоги, яка фінансується Агентством Міжнародного розвитку США:

- обмін інформацією, джерелами якої є будь-які існуючі, а також наново створювані загальнодоступні бази даних;
- допомога щодо забезпечення електронного зв'язку між ДСК і Секретаріатом Верховної Ради України з використанням найбільш ефективних телекомунікаційних систем, які є в їх розпорядженні для обміну інформацією.

У зв'язку з цим перспективи розвитку інформаційних систем в Україні характеризуються тим, що:

- вони все більше будуть входити до сфери нашої повсякденної діяльності;
- будуть розвиватися у формі інтегрованих, а також широко розподільних територіальних систем;
- все більша кількість інформаційних систем буде набувати ознак інтелектуальної обробки інформації в плані інформаційних систем штучного інтелекту, моделюючих інтелектуальних і експертних систем.

В Україні, як і в деяких інших країнах, має бути прийнята за національну програма створення інформаційних систем, які, будучи пов'язаними з національними системами інших країн, утворять Глобальну інформаційну світову інфраструктуру. З метою вирішення цієї програми ще 22 листопада 1994 року був підписаний Меморандум про взаєморозуміння щодо співробітництва у сфері телекомунікацій і розвитку всесвітньої інформаційної інфраструктури між Урядом України і Урядом Сполучених Штатів Америки.

Даний Меморандум є концентрованим програмним документом, який відображає мету і завдання України у стосунках з найбільш розвинутою і досвідченою в даних питаннях країною. Найбільший інтерес у зв'язку з цим викликає перелік тих завдань, що с перспективними для України.

Відповідно до положень Хартії українсько-американського партнерства, дружби і співробітництва, підписаної 22 листопада 1994 року у Вашінгтоні, сторони визнали надзвичайно важливим створення в Україні сучасної телекомунікаційної інфраструктури, яка необхідна для успішного розвитку торгових та інвестиційних процесів, інтеграції України в світову економіку, для розвитку охорони здоров'я, освіти, захисту навколишнього середовища і демократії, для надання можливості доступу до набутого людством знання, сприяння в інтелектуальному та духовному розвитку.

Таким чином, можна зробити висновки про те, що основними завданнями України на найближчий час буде:

- сприяння у співробітництві державних та приватних структур (компаній), які зайняті у сфері телекомунікацій; розробка в зв'язку з цим відповідного національного законодавства;
- участь у створенні Всесвітньої інформаційної інфраструктури, а саме — впровадження приватних інвестицій, конкурентоздатного ринку, гнучкої регулюючої системи;
- сприяння процесу розвитку мереж телекомунікацій між країнами на основі відкритого доступу до сучасних технологій, їх інтеграції до Всесвітньої мережі телекомунікацій для створення основи майбутньої Всесвітньої інформаційної інфраструктури;

- сприяння комерційній діяльності, проектам та інвестиціям приватного сектору, які спрямовані на розвиток телекомунікацій; дослідження їх можливостей в рамках меж міжнародних інститутів;
- заохочення проектів, які спрямовані на розвиток та виробництво телекомунікаційного обладнання для мереж загального користування.

Звичайно, для реалізації цих планів Україна повинна розробити спеціальні програми і проекти, залучивши до цього спеціалістів всіх галузей права.

7.7.6. Розвиток GII та Internet за участю України

Поточне тисячоліття поставить перед правознавцями складні питання, пов'язані з розвитком глобальної інформаційної інфраструктури (GII). Слід зазначити, що ідеї створення світової мережі з'явились одночасно в Європі, США і СНД. Різні країни різноманітними шляхами наближались до входження у світову інформаційну мережу. Однак, всі вони в результаті прийшли до розуміння створення єдиної глобальної інформаційної інфраструктури.

Концепція GII, тобто всесвітньої телекомунікаційної мережі, була вперше сформульована у березні 1994 року на першій конференції Міжнародного союзу телекомунікацій в Буенос-Айресі. Виникла необхідність укласти угоду між урядами всіх країн щодо основних положень, які дозволили б людям у всьому світі спілкуватись між собою. У зв'язку з цим Альберт Гор виклав такі основні правила, або принципи:

- стимулювання приватних інвестицій;
- розвиток конкуренції;
- надання вільного доступу до глобальної мережі всім постачальникам і користувачам інформації;
- створення гнучкої нормативно-правової бази, яка здатна адаптуватись до швидких змін в індустрії, на ринку інформаційних технологій;
- забезпечення універсального характеру послуг.

Слід зазначити, що в Росії, наприклад, послугами електронних інформаційних мереж користуються більше 600 000 чол., зареєстровано більше 3000 ресурсів російською мовою (проти 370 000 — у США), щоправда російська сфера розвивається більш динамічно.

Кількість користувачів українського сегменту Internet за останні два роки збільшилося більш ніж у три рази і на кінець 2000 року складала за різними оцінками від 320 до 370 тис. постійних користувачів (менше 1 відсотку дорослого населення України) і близько 300 тис. громадян, які користувалися послугами Internet час від часу. Значно зріс і обсяг інформаційних ресурсів, які доступні в Internet. Якщо на початку 1999 року такі ресурси розмішувалися на 1,4 тис. Web-серверів, то наприкінці 2000 року — на 9268 Web-серверів, а у 2005 — більш, ніж на 20000. За кількістю хостів, які підключені до Internet, Україна знаходиться на 28-му місці в Європі і на 45-му місці у світі.

У спеціальній літературі звертається увага на те, що від розвитку Internet виграють, перш за все, розвинуті країни, тому що це їх природній шлях розвитку. Тому для України необхідною є державна програма підтримки і розвитку Internet.

Частина III

Нейронні та віртуальні мережі

Розділ 8

Нейронні мережі

"Якщо вам незрозуміле якесь слово,
не звертайте на нього уваги.
Текст повністю зберігає сенс і без нього"
(Закон Купера)

8.1. Загальна характеристика нейронних мереж

Нейронні мережі (НМ) — це нова галузь прикладної математики, яка за останні десятиріччя набула бурхливого розвитку. Пояснюється це застосуванням даного математичного апарату до надзвичайно широкого кола проблем, до яких належать адаптивне керування, автоматизація процесів розпізнавання образів, апроксимація функціоналів, створення експертних систем, організація асоціативної пам'яті та багато інших застосувань в різних галузях людської діяльності. Серед цих галузей можна згадати медицину, фізику, фінансове планування, інженерні науки, комп'ютерні мережі, штучний інтелект в комп'ютерних іграх та інші.

За допомогою НМ можна, наприклад, прогнозувати показники біржового ринку, виконувати розпізнавання оптичних або звукових сигналів, створювати самонавчаючі системи, здатні керувати автомашиною при паркуванні або синтезувати мову по тексту. У той час як на заході застосування НМ досить розповсюджене, у нас це ще до деякої міри екзотика — українських фірм, що використовують НМ у практичних цілях, поки що небагато.

Теорія НМ є алгоритмічним базисом розвитку нейрокомп'ютерів, подібно тому, як булева алгебра більше 50 років була алгоритмічним базисом однопроцесорних та багатопроцесорних ЕОМ.

Сучасні цифрові обчислювальні машини перевершують людину за здатністю виконувати числові та символьні обчислення. Проте людина може без зусиль вирішувати складні задачі сприйняття зовнішніх даних (наприклад, пізнавати в натовпі знайомого тільки по його обличчю, що промайнуло) з такою швидкістю і точністю, що

найпотужніший в світі комп'ютер в порівнянні з ним виявиться безнадійним тугодумом. Причина такої значної відмінності в їх продуктивності у тому, що архітектура біологічної нейронної системи зовсім не схожа на архітектуру машини фон Неймана, а це істотно впливає на типи функцій, які більш ефективно виконуються кожною моделлю. Порівняння характеристик машини фон Неймана та біологічної нейронної системи наведено в табл. 8.1.

Таблиця 8.1

Параметри порівняння	Машина фон Неймана	Біологічна нейронна система
	1	2
Процесор	Складний	Простий
	Висока швидкість	Низька швидкість
	Один або декілька	Багато
Пам'ять	Відділена від процесора	Інтегрована в процесор
	Локалізована	Розподілена
	Адресація не по змісту	Адресація по змісту
Обчислення	Централізовані	Розподілені
	Послідовні	Паралельні
	Програми, що зберігаються	Самостійне навчання
Спеціалізація	Чисельні й символьні операції	Проблеми сприйняття
Середовище функціонування	Строго визначене	Погано визначена
	Строго обмежене	Без обмежень
Надійність	Висока вразливість	Живучість

Нейронна мережа — це мережа зі скінченою кількістю шарів (*layers*) з однотипних елементів — аналогів нейронів з різними типами зв'язків між шарами. При цьому число нейронів в шарах вибирається виходячи з необхідності забезпечення заданої якості розв'язання задачі, а число шарів нейронів — як можна менше, для скорочення часу рішення задачі.

Штучні нейронні мережі завдячують своєю появою перш за все нейробіологам, які довгий час вивчали структуру нервових клітин людини і намагалися проникнути в таємницю мозку.

В основу штучних нейронних мереж покладені наступні риси живих нейронних мереж, що дозволяють їм добре справлятися з нерегулярними задачами:

- простий обробний елемент — нейрон;
- в обробці інформації бере участь дуже велика кількість нейронів;
- один нейрон пов'язаний з великою кількістю інших нейронів, що змінюються по вазі зв'язку між нейронами;
- паралельність обробки інформації.

Спочатку з'явилися перші математичні моделі, які намагалися відтворити хоча б частково, на примітивному рівні, функціонування людського мозку, що складається з сотень мільярдів біологічних нейронів, кожен з яких має сотні тисяч зв'язків з іншими нейронами.

Перші нейронні мережі були виконані у вигляді електронних мереж, а вже потім вони були перенесені в сферу комп'ютерного моделювання.

Психологи та нейробіологи намагалися створити модель людського навчання, і найбільш плідною з перших таких моделей стала модель, запропонована Д. Хеббом в 1949 році, яка і стала базовою точкою у розвитку нейронних мереж як галузі прикладної математики.

В результаті робіт вчених, які почали розвивати цю галузь (а серед них найвідомішими на той час в п'ятидесятих-шестидесятих роках були Розенблатт, Уідроу, Мінський), з'явився одношаровий персептрон — нейронна мережа, яка складалася з одного шару штучних нейронів. Використовуючи персептрон можна було вирішувати широке коло задач, і завдяки цьому в той період склалася ілюзія, що штучні нейронні мережі здатні в повному обсязі моделювати діяльність людського мозку за наявності необхідних ресурсів для побудови нейронної мережі потрібних розмірів.

Але ці ілюзії, що дали величезний поштовх популярності цієї теми в ті роки, були все ж таки розвіяні, і причиною цьому стала робота одного з відомих вчених цієї галузі Мінського, який опублікував книгу "Персептрони". В ній він математично обґрунтовував межі можливостей персептрону, доводячи, що клас задач, які можна з його допомогою вирішувати, далеко не такий широкий, як здавалося раніше.

Авторитет Мінського, а також песимізм, з яким він описував майбутнє нейронних мереж, загасив бум, який було піднявся. І лиш деякі вчені, такі як Кохонен, Гроссберг, Андерсон, Хопфілд, Хеммінг проводили дослідження в цьому напрямку, не втрачаючи при цьому оптимізму.

Так з'явився ряд нових нейропарадигм, нових по структурі нейромереж, застосування яких дозволяло значно розширити клас задач, що до того вирішувалися нейронними мережами, і дати відповідь на контрприклади, наведені свого часу Мінським щодо неспроможності нейронних мереж.

Більше того, нейронні мережі почали все більше застосовуватися як прикладна галузь, і нею почали зацікавлюватися комерційні структури.

Наприкінці 80-х років виникла нова хвиля популярності нейронних мереж, які вже на той час значно розвинулися, і зараз ця проблема досить актуальна, про що свідчить величезна кількість публікацій на цю тему, комерційних застосувань та міжнародних конференцій, присвячених суто нейронним мережам.

Нейронна мережа являє собою сукупність великого числа порівняно простих елементів — нейронів, топологія з'єднань яких залежить від типу мережі. Щоб створити нейронну мережу для рішення конкретної задачі, необхідно вибрати, яким способом варто з'єднувати нейрони один з одним, і відповідним чином підібрати значення вагових параметрів на цих зв'язках. Чи може впливати один елемент на інший, залежить від встановлених сполучень. Вага сполучення визначає силу впливу.

Розвиток штучних нейронних мереж надихається біологією. Тобто, розглядаючи конфігурації мереж та алгоритми, дослідники осмислюють їх у термінах організації мозкової діяльності. Але на цьому аналогія може і закінчитись, бо наші знання про роботу мозку дуже обмежені. Тому розробникам мереж доводиться виходити за межі

сучасних біологічних знань у пошуках структур, здатних виконувати корисні функції. У багатьох випадках це призводить до необхідності відмовитись від біологічної правдоподібності. Мозок стає просто метафорою, і створюються мережі, які неможливі в живій матерії чи потребують неправдоподібно великих допущень про анатомію та функціонування мозку.

Незважаючи на те, що зв'язок з біологією слабкий і найчастіше несуттєвий, штучні нейронні мережі продовжують порівнюватися з мозком. Їхнє функціонування часто нагадує людське пізнання, тому важко уникнути цієї аналогії. Нажаль, такі порівняння неплодотворні і створюють невиправдані сподівання, що неминуче веде до розчарування.

Незважаючи на зроблені попередження, корисно все ж таки знати дещо про нервову систему ссавців, тому що вона успішно вирішує задачі, до виконання яких лише прагнуть штучні системи.

8.2. Структура нейронів

Нервова система людини, побудована з елементів з назвою "нейрони", має приголомшуючу складність. Близько 10^{11} нейронів беруть участь у приблизно 10^{15} передавальних зв'язках, що мають довжину близько одного метру. Кожен нейрон має багато властивостей, спільних з іншими елементами тіла людини, але його унікальною здатністю є прийом, обробка і передача електрохімічних сигналів по нервових шляхах, що утворять комунікаційну систему мозку.

Прототипом для створення нейрона послужив біологічний нейрон головного мозку. Спрощено функціонування нейрона можна уявити наступним способом:

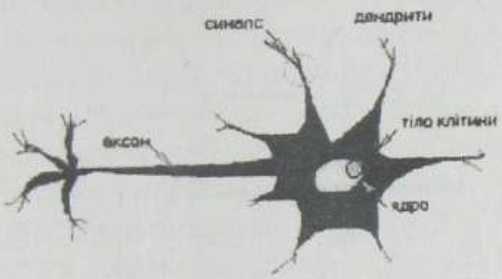
1. Нейрон одержує набір (вектор) вхідних сигналів.
2. У тілі нейрона оцінюється сумарне значення вхідних сигналів. Проте входи нейрона нерівнозначні. Кожен вхід характеризується деяким ваговим коефіцієнтом, що визначає важливість інформації, яка проходить по ньому. Таким чином, нейрон не просто підсумовує значення вхідних сигналів, а обчисляє скалярний добуток вектора вхідних сигналів і вектора вагових коефіцієнтів.
3. Нейрон формує вихідний сигнал, інтенсивність якого залежить від значення обчисленого скалярного добутку. Якщо воно не перевищує деякого заданого порога, то вихідний сигнал не формується зовсім — нейрон "не спрацьовує".
4. Вихідний сигнал передається на входи інших нейронів.

На рис. 8.1. наведено біологічну структуру нейрона (ліворуч) та його математична модель (праворуч) у класичному вигляді. Неважко при детальному розгляді помітити схожість між цими двома зображеннями, але разом з тим подібна формалізація вочевидь значно звужує функції нейрона.

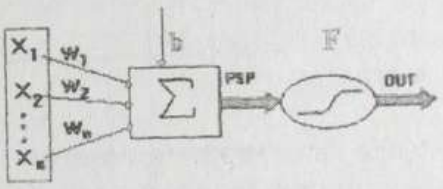
Як же функціонує біологічний нейрон? Дендрити йдуть від тіла нервової клітини до інших нейронів, де вони приймають сигнали в точках з'єднання, що називаються синапсами. Прийняті синапсом вхідні сигнали підводяться до тіла нейрона. Тут вони підсумовуються, причому одні входи прагнуть збудити нейрон, а інші — перешкодити його збудженню. Коли сумарне збудження в тілі нейрона перевищує деякий поріг,

нейрон збуджується, посилаючи по аксонові сигнал іншим нейронам. У цієї основної функціональної схеми багато ускладнень і виключень, проте більшість штучних нейронних мереж моделюють лише ці прості властивості.

Біологічний нейрон



Математичний нейрон



$$PSP = \sum_{i=1}^{n} w_i x_i - b$$

$$OUT = F(PSP)$$

Рис. 8.1. Класичний вигляд нейронів

Не зважаючи на величезну кількість нейронів, їхні тіла займають лише декілька відсотків загального об'єму мозку. Майже весь інший простір зайнятий міжнейронними зв'язками. Число зв'язків кожного нейрона не має аналогів в сучасній техніці. Розуміння того, що міжнейронні зв'язки відносяться до основних структурних компонентів мозку, що в першу чергу визначають його функціональні характеристики, є одним з найістотніших висновків, зроблених нейрофізіологами. На підтвердження можна привести вислів відомого нейрофізіолога Е. Кендела: "По переконанням багатьох нейробіологів врешті-решт буде доведено, що унікальні властивості кожної людини — здатність відчувати, думати, навчатися і пам'ятати — укладені в строго організованих мережах синаптичних взаємозв'язків між нейронами головного мозку".

Штучний нейрон імітує у першому наближені властивості біологічного нейрона. Він володіє групою синапсів — однонаправлених вхідних зв'язків, з'єднаних із виходами інших нейронів, а також має аксон — вихідний зв'язок даного нейрона, по якому сигнал (збудження або гальмування) надходить на синапси наступних нейронів. Кожен вхід множиться на відповідну вагу, аналогічну синаптичній силі, і всі добутки підсумовуються, визначаючи рівень активації нейрона.

Загальний вигляд нейрона, що реалізує ці властивості, наведений на рис. 8.2. Тут множина вхідних сигналів позначена вектором X. Кожна вага w_i відповідає "силі" одного біологічного синаптичного зв'язку. Множина ваг у сукупності позначається вектором W.

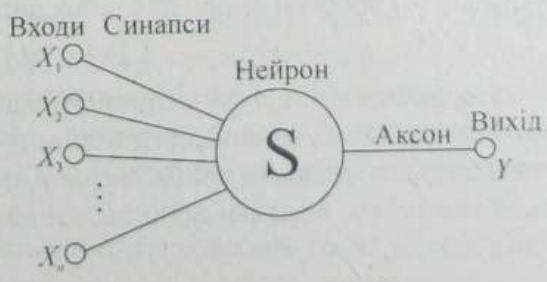


Рис. 8.2. Штучний нейрон

Підсумовуючий блок, що відповідає тілу біологічного елемента, складає зважені входи алгебраїчно

$$S = \sum_{i=1}^{n} x_i \cdot w_i \,. \tag{8.1}$$

Вихід нейрона є функцією його стану:

$$Y = F(x)$$

Нелінійна функція F називається ***активуючою*** й може мати різний вигляд, що показано на рис. 8.3.

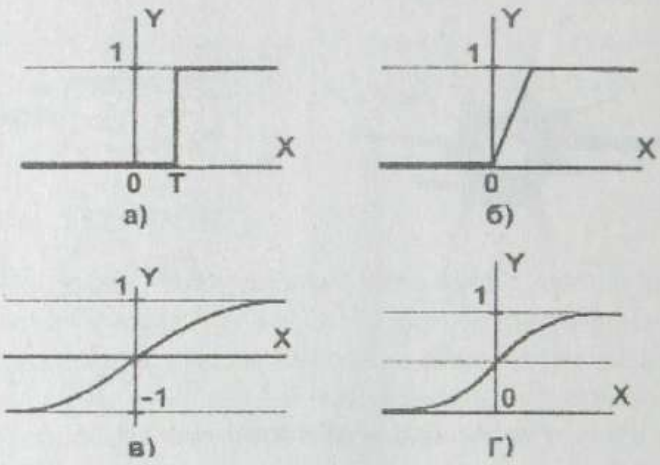


Рис. 8.3 . Типи активуючих функцій:
а — функція одиничного стрибка; б — лінійний поріг (гістерезис);
в — сигмоїд — гіперболічний тангенс; г — сигмоїд — формула (8.2).

У випадку, коли функція активації одна і таж для усіх нейронів мережі, мережу називають ***однорідною*** (***гомогенною***). Якщо ж активуюча функція залежить ще від одного або декількох параметрів, значення яких змінюються від нейрона до нейрона, то мережу називають ***неоднорідною*** (***гетерогенною***).

Однією із найбільш розповсюджених є нелінійна функція з насиченням, або, як її ще називають, ***логістична функція*** чи ***сигмоїд*** (тобто функція S-подібного вигляду):

$$f(x) = \frac{1}{1 + e^{-\alpha x}} \tag{8.2}$$

При зменшенні α сигмоїд стає більш пологим, в межі при $\alpha = 0$, вироджуючись в горизонтальну лінію на рівні 0,5. При збільшенні α сигмоїд наближається до функції одиничного стрибка з порогом T в точці $x = 0$. Із виразу для сигмоїда видно, що вихідне значення нейрона лежить в діапазоні $[0,1]$. Одна з цінних властивостей сигмоїдной функції — простий вираз для її похідної:

$$f'(x) = \alpha f(x)(1 - f(x))$$

Слід зазначити, що сигмоїдна функція диференційована по всій осі абсцис, що широко використовується в багатьох алгоритмах навчання. Крім того, вона володіє властивістю підсилювати слабкі сигнали краще, ніж сильні, і запобігає насиченню від сильних сигналів, оскільки вони відповідають областям аргументів, де сигмоїд має пологий нахил.

Іншою активуючою функцією, що широко використовують, є гіперболічний тангенс. На відміну від логістичної функції, гіперболічний тангенс приймає значення різних знаків, що для ряду мереж виявляється вигідним.

Кажучи про можливу класифікацію НМ, важливо відзначити існування бінарних та аналогових мереж. Перші оперують з бінарними сигналами, і вихід кожного нейрона може приймати тільки два значення: логічний нуль ("загальмований" стан) або логічна одиниця ("збуджений" стан). У аналогових мережах вихідні значення нейронів здатні приймати безперервні значення.

Ще одна класифікація ділить НМ на синхронні та асинхронні. В першому випадку в кожен момент часу свій стан міняє лише один нейрон. У другому — стан міняється одразу у цілої групи нейронів, як правило, у всього шару. Для програмних імітаторів нейронних мереж на цифрових ЕОМ, питання, пов'язані з синхронізацією, розв'язуються комп'ютером, на якому реалізуються НМ.

Розглянута проста модель штучного нейрона істотно спрощує ряд властивостей свого біологічного двійника. Наприклад, вона не бере до уваги затримки в часі, які впливають на динаміку системи. Вхідні сигнали одразу ж породжують вихідний сигнал, і, що більш важливо, вона не враховує дію синхронізуючої функції біологічного нейрона, яку ряд дослідників вважає вирішальною. Не зважаючи на ці обмеження, мережі, побудовані з таких нейронів, мають властивості, що дуже нагадують біологічну систему. Тільки час і подальші дослідження можуть дати відповідь на питання, чи є подібні збіги випадковими, чи це наслідок того, що в моделі вірно відтворені основні риси біологічного нейрона.

Перейдемо тепер до питання утворення цих мереж та їх конструювання. Строгих обмежень тут немає, лише б входи одержували деякі сигнали. Можливості безмежні, але звичайно використовують декілька стандартних архітектур, з яких при деяких невеликих модифікаціях будують більшість мереж.

Якщо задача не може бути зведена до жодного з відомих типів, розробнику доводиться вирішувати складну проблему синтезу нової конфігурацій. При цьому він керується декількома основними принципами:

- можливості мережі зростають із збільшенням числа осередків мережі, густини зв'язків між ними і числа виділених шарів;
- введення зворотних зв'язків разом із збільшенням можливостей мережі "піднімає" питання про динамічну стійкість мережі;
- складність алгоритмів функціонування мережі (зокрема, введення декількох типів синапсів — збудливих, гальмуючих та ін.) також сприяє посиленню потужності НМ.

Питання про необхідні і достатні властивості мережі для вирішення того або іншого роду задач є цілим напрямом нейрокомп'ютерної науки. Оскільки проблема синтезу НМ сильно залежить від задачі, що вирішується, дати загальні детальні рекомендації важко. В більшості випадків оптимальний варіант отримують на основі інтуїтивного підбору. Єдина жорстка вимога, що пред'являється архітектурою до елементів мережі, — це відповідність розмірності вектора вхідних сигналів мережі числу її входів.

Проста одношарова мережа, що складається з групи нейронів, показана на рис. 8.4

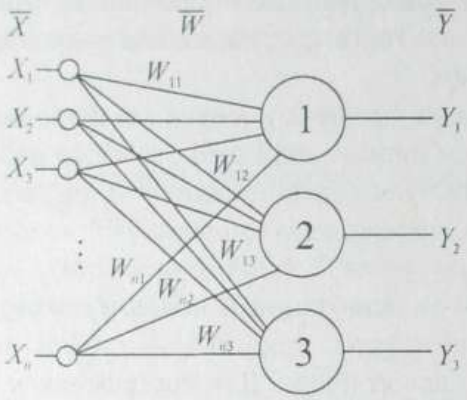


Рис. 8.4. Одношарова НМ

На n входів поступають якісь сигнали, що проходять по синапсах на три нейрони, створюючи єдиний шар цієї НМ і видаючи три вихідних сигнали:

$$y_j = f\left[\sum_{i=1}^{n} x_i w_{ij}\right], \text{ де } j = 1,2,3.$$

У штучних і природних (біологічних) мережах багато з'єднань можуть бути відсутніми, усі з'єднання показані у цілях спільності.

Вочевидь, що всі вагові коефіцієнти синапсів одного шару нейронів можна звести до матриці W, в якій кожен елемент w_{ij} задає величину i-го синаптичного зв'язку j-го нейрона. Таким чином, процес, що відбувається у НМ, може бути записаний у матричному вигляді:

$$Y = F(XW)$$

де X та Y — відповідно вхідні і вихідні сигнальні вектори; $F(V)$ — активуюча функція, що застосовується поелементно до компонент вектора V.

На рис. 8.5 наведена двошарова НМ, отримана із одношарової шляхом добавлення другого шару, що складається з двох нейронів.

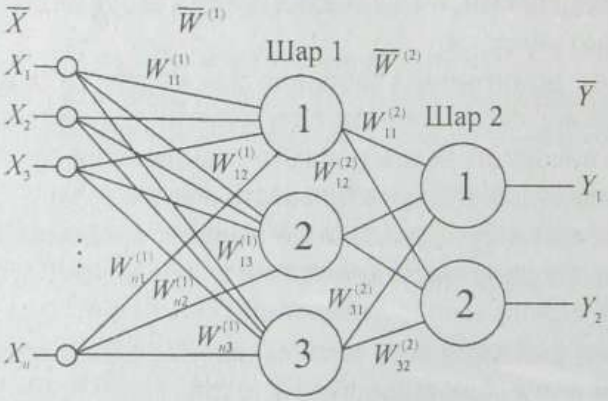


Рис. 8.5. Двошарова НМ

Тут доречно відмітити важливість ролі раніше розглянутої нелінійної активуючої функції, так якби вона не мала даної властивості або не належала алгоритму роботи кожного нейрона, результат функціонування будь-якої p-шарової НМ з ваговими матрицями $W^{(i)}$, де $i = 1, 2, ..., p$ для кожного шару і зводився б до множення вхідного вектора сигналів X на матрицю

$$W^{(\Sigma)} = W^{(1)} \cdot W^{(2)} \cdot \ldots \cdot W^{(p)},$$

тобто є фактично така p-шарова НМ, еквівалентна одношаровій НМ з ваговою матрицею єдиного шару $W^{(\Sigma)}$:

$$Y = XW^{(\Sigma)}.$$

Таким чином, для розширення обчислювальних можливостей багатошарових НМ в порівнянні з одношаровими НМ необхідно використовувати нелінійні активуючі функції.

Продовжуючи розмову про нелінійність, необхідно відзначити, що вона іноді може вводитись і в синаптичні зв'язки. У більшості відомих на сьогоднішній день НМ для знаходження зваженої суми входів нейрона використовують формулу (8.1), проте в деяких додатках НМ корисно ввести інший запис, наприклад:

$$S = \sum_{i=1}^{n} x_i^2 w_i .$$

або

$$S = \sum_{i=1}^{n} x_i x_{((i+1) \bmod n)^{w_i}} .$$

Питання в тому, щоб розробник НМ чітко розумів, для чого він це робить, якими цінними властивостями він тим самим додатково наділяє нейрон, і яких позбавляє. Введення такого роду нелінійності, взагалі кажучи, збільшує обчислювальну потужність мережі, тобто дозволяє з меншого числа нейронів з "нелінійними" синапсами сконструювати НМ, що виконує роботу звичної НМ з великим числом стандартних нейронів і складнішою конфігурацією.

У мереж, що були розглянуті, не було зворотних зв'язків, тобто з'єднань, що йдуть від виходів деякого шару до входів цього ж шару або попередніх шарів. Цей спеціальний клас мереж представляє інтерес і широко використовується.

Декілька слів необхідно сказати про необхідну потужність вихідного шару мережі, що виконує остаточну класифікацію простору станів. Річ у тому, що для розділення безлічі вхідних образів, наприклад, по двох класах, достатньо всього одного виходу. При цьому кожен логічний рівень ("1" чи "0") позначатиме окремий клас.

На двох виходах можна закодувати вже чотири класи і т. ін. Однак, результати роботи мережі, організованої таким чином, недостатньо надійні. Для підвищення достовірності класифікації бажано ввести надмірність шляхом виділення кожному класу одного нейрона у вихідному шарі або, що ще краще, — декількох, кожний з яких навчається визначати належність конкретних станів входів до конкретного класу зі своєю мірою достовірності, наприклад, високої, середньої і низької. Такі НМ дозволяють проводити класифікацію вхідних неявно виражених станів, об'єднаних в нечіткі (розмиті або пересічні) множини. Ця властивість дозволяє широко використовувати НМ в практичних додатках.

8.3. Математична модель штучного нейрона

Штучний нейрон імітує в першому наближенні властивості біологічного нейрона. На вхід штучного нейрона надходить множина сигналів, кожний з який є виходом іншого нейрона. Кожен вхід збільшується на відповідну вагу, аналогічну синаптичній силі, і всі добутки підсумовуються, визначаючи рівень активації нейрона.

Хоча мережеві парадигми (вони ж "нейропарадигми" методики побудови та навчання штучних нейронних мереж) дуже різноманітні, в основі майже кожної з них лежить ця конфігурація. Тут множина вхідних сигналів, позначених x_1, x_2, ..., x_n надходить на штучний нейрон. Ці вхідні сигнали, які у сукупності позначаються вектором X, відповідають сигналам, що приходять у синапси біологічного нейрона.

Кожен сигнал коригується на відповідну вагу w_1, w_2, ..., w_n, і надходить на підсумовуючий блок, позначений як Σ. Кожна вага відповідає "силі" одного біологічного синаптичного зв'язку. Множина ваг у сукупності позначається вектором W. Підсумовуючий блок, що відповідає тілу біологічного елемента, складає зважені входи алгебраїчно, створюючи вихід (будемо називати його PSP). У векторних позначеннях це може бути компактно записано в такий спосіб:

$$PSP = XW.$$

Далі сигнал, як правило, перетворюється активуючою функцією F і дає вихідний сигнал нейрону OUT. Активуюча функція може бути звичайною лінійною функцією

$$OUT = K(PSP),$$

де K — постійна порогової функції

$$OUT = 1, \text{ якщо } P > T,$$
$$OUT = 0 \text{ в інших випадках,}$$

де T — деяка постійна гранична величина, або ж функція, що більш точно моделює нелінійну передатну характеристику біологічного нейрона і що надає нейронній мережі більшої можливості.

В такому випадку вихідний сигнал OUT нейрона утворюється шляхом нелінійного перетворення сигналу вхідного збудження нейрона, постсинаптичного потенціалу PSP.

Постсинаптичний потенціал нейрона формується як зважена сума вихідних сигналів x_i, інших n нейронів мережі, зв'язаних із даним нейроном:

$$PSP = \sum_{i=1}^{n} w_i x_i - b.$$

Тут w_i, — вектор вагових коефіцієнтів синаптичних зв'язків даного нейрона, b — порогове значення нейрона. Взагалі кажучи, використання порогу не є обов'язковим. Але його присутність в більшості випадків покращує здатність мережі до запам'ятовування образів.

Цьому явищу можна дати наступну геометричну інтерпретацію. Один нейрон (також цей одиничний елемент називають ***персептроном***) має здатність до лінійного розділення вхідних образів на класи. Тобто фактично він реалізує деяку гіперплощину в просторі вхідних образів, яка розбиває ці образи на два класи.

Наявність порогу дозволяє змістити цю гіперплощину з початку координат, таким чином розширюючи клас навчаючих вибірок, на яких нейрон може навчатися і потім правильно їх класифікувати.

Нелінійне перетворення задається активуючою функцією $f()$, що вибирається з класу так званих неперервних сигмоїдних функцій $U(x)$, таких, що для будь-якого як завгодно малого $\varepsilon > 0$ значення можна зазначити деякий кінцевий $[-\delta, \delta]$ інтервал:

$$U(x) = \begin{cases} \varepsilon, x < -\delta; \\ \varepsilon < U(x) < 1 - \varepsilon, |x| \le \delta; \\ 1 - \varepsilon, x > \delta. \end{cases}$$

У якості сигмоїдних функцій, що при конструюванні конкретних нейромереж можуть бути використані як математичні моделі нейронів мережі, виступають:

а) $U(x) = \lim\limits_{\alpha\to\infty} \dfrac{1}{1+e^{-\alpha x}}$, для якої $U'(x) = aU(x)h[1-U(x)]$, $\delta = -\dfrac{\ln(1-\varepsilon)}{\alpha}$;

б) $U(x) = \lim\limits_{\alpha\to\infty}\left(\dfrac{1}{2} + \dfrac{1}{\pi} arctg\,\alpha x\right)$, для якої $U'(x) = \dfrac{\alpha}{\pi[1+(\alpha x)^2]}$,

$\delta = \dfrac{1}{\alpha} tg\,\pi(\varepsilon - \dfrac{1}{2})$;

в) $U(x) = \lim\limits_{\alpha\to\infty} 2^{-e^{-\alpha x}}$, для якої $U'(x) = \alpha \ln 2 e^{-\alpha x} 2^{-e^{-\alpha x}}$, $\delta = -\dfrac{1}{\alpha}\ln(-\log_2 \varepsilon)$.

Ці функції включають параметр α, задаючи який можна регулювати крутизну порогу. Задаючи значення ε на межі інтервалу чутливості $[-\delta, \delta]$ можна обчислити сам цей інтервал чутливості δ порогової функції. Тут же наведені вирази похідних від цих функцій $U'(x)$, що у режимі навчання мережі (для мереж прямого розповсюдження) обчислюються паралельно з $U(x)$.

Не зважаючи на те, що наведена вище класична математична модель штучного нейрону є разом з тим і найбільш розповсюдженою, все ж існують й інші моделі, які знайшли успішне практичне застосування.

Значно більшу різноманітність спостерігається тоді, коли штучні нейрони починають об'єднуватися у штучні нейронні мережі і ці нейронні мережі починають "навчатися". В цьому полягає суть так званих нейропарадигм, котрих на даний момент вже існує досить велика кількість.

8.4. Типи нейронних мереж

Найбільш загальний поділ нейропарадигм наведений на рис. 8.6.

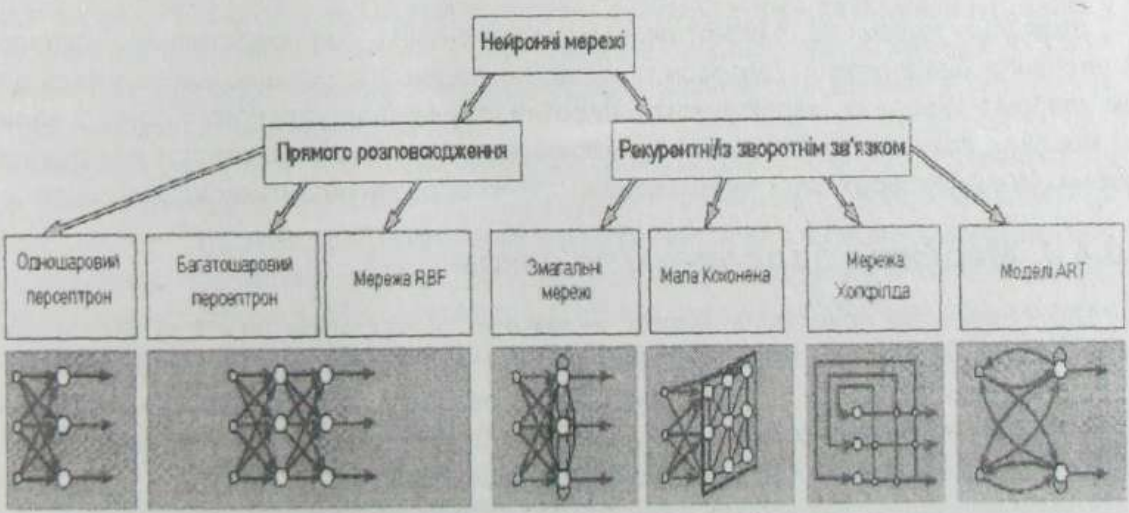


Рис. 8.6. Класифікація нейронних мереж

Охарактеризуємо коротко деякі із зображених типів нейронних мереж, та коло задач, які цими мережами вирішуються в сучасних інформаційних системах.

8.4.1. Одношаровий персептрон

Одношаровий персептрон являє собою чи не найпростіший тип нейронних мереж. Історично він з'явився одним з найперших і реалізовував ідею спільної роботи групи нейронів. Він використовувався для розв'язання таких задач як прогнозування курсу акцій на ринку цінних паперів, прогнозування погоди та інших задач подібного характеру. Саме вони свого часу підштовхнули бум на нейронні мережі, що виник в 60-ті роки. Але насправді клас задач, що вирішується одношаровим персептроном досить вузький, що було досить швидко помічено і доведено математично.

8.4.2. Багатошарові персептрони та мережі RBF

Багатошарові персептрони та мережі RBF (Radial Basis Function — радіальна базисна функція) являють собою розвиток одношарових персептронів з більш широкими можливостями. Багатошаровий персептрон складається, як видно з назви, з двох або більше послідовних шарів, кожен з яких є одношаровим персептроном. Мережа RBF — це різновид двошарових мереж прямого розповсюдження, що використовує спеціальну структуру кожного з шарів, а також спеціальну активаційну функцію в нейронах другого шару.

Можливості цих мереж ширші, ніж у одношарових пєрсептронів. З їх допомогою також вирішуються різноманітні задачі прогнозування, наприклад класифікація підводних об'єктів за даними радіолокації, розпізнавання нескладних образів (літер, звуків мови, ознак різноманітних об'єктів для класифікації тощо), виявлення втручання зловмисників ("хакерів") в комп'ютерні мережі, встановлення медичних діагнозів, моделювання процесів в ядерних установках та багато інших прикладних задач.

8.4.3. Змагальні мережі та мапа Кохонена

Змагальні мережі та їх розвиток — мапа Кохонена — це представники технологій навчання без вчителя. Діапазон прикладних задач, що успішно вирішуються даним класом нейронних мереж досить широкий: від інтелектуального пошуку в великих масивах документів (технології, які починають використовуватися в новітніх пошукових Web-серверах) до розпізнавання мовних образів, відбитків пальців тощо.

8.4.4. Мережі із зворотнім зв'язком

Ще один клас нейронних мереж — мережі із зворотнім зв'язком (Хопфілда, ART) — являє собою так звану "асоціативну пам'ять". Специфіка цих мереж в тому, що вони в своїй більшості не вимагають багаторазового представлення навчальних образів. Використовуються для розпізнавання зображень (портретів людей, відбитків пальців, геоінформації та ін.), мови людини та інших задач.

Насправді варто зазначити, що на практиці для вирішення складних задач не використовуються якийсь один із класів нейронних мереж, а скоріше їх комбінація, яка є найбільш оптимальною для даної задачі.

8.5. "Навчання" нейронних мереж

Розглянемо більш детально, що ж являє собою процес набуття знань штучними нейронними мережами. Серед усіх цікавих властивостей штучних нейронних мереж жодне не захоплює уяви так, як їхня здатність до навчання. Це навчання до такого ступеня нагадує процес інтелектуального розвитку людської особистості, що може показатися, що досягнуто глибокого розуміння цього процесу. Можливості навчання штучних нейронних мереж обмежені, і у цьому напрямку потрібно ще вирішити багато складних задач.

Мережа навчається, щоб для деякої множини входів давати бажану (чи, принаймні, подібну до неї) множину виходів. Кожна така вхідна (чи вихідна) множина розглядається як вектор. Навчання здійснюється шляхом послідовного пред'явлення вхідних векторів з одночасним налаштовуванням ваг відповідно до визначеної процедури. У процесі навчання ваги мережі поступово стають такими, щоб кожен вхідний вектор виробляв вихідний вектор.

Розрізняють алгоритми навчання з вчителем і без вчителя. Навчання з вчителем припускає, що для кожного вхідного вектора існує цільовий вектор, що представляє собою необхідний вихід. Разом вони називаються ***навчальною парою***. Звичайно мережа навчається на деякім числі таких навчальних пар. Формується вихідний вектор, після чого обчислюється вихід мережі, який порівнюється з відповідним цільовим вектором. Різниця (помилка) за допомогою зворотного зв'язку подається в мережу, і ваги змінюються відповідно до алгоритму, що прагне мінімізувати помилку. Вектори навчальної множини формуються послідовно, обчислюються помилки і ваги будуються для кожного вектора доти, доки помилка по всьому навчальному масиву не досягне прийнятного низького рівня.

Незважаючи на численні прикладні досягнення, навчання з учителем критикувалося за свою біологічну неправдоподібність. Важко уявити навчальний механізм у мозку, який би порівнював бажані і дійсні значення виходів, виконуючи коригування за допомогою зворотного зв'язку. Якщо допустити подібний механізм у мозку, то звідки тоді виникають бажані виходи?

Навчання без учителя є набагато більш правдоподібною моделлю навчання в біологічній системі. Розвита Кохоненом і багатьма іншими, вона не має потреби в цільовому векторі для виходів і, отже, не вимагає порівняння з визначеними ідеальними відповідями. Навчальна множина складається лише з вхідних векторів. Навчальний алгоритм налаштовує ваги мережі так, щоб виходили погоджені вихідні вектори, тобто щоб пред'явлення достатньо близьких вхідних векторів давало однакові виходи.

Процес навчання, таким чином, виділяє статистичні властивості навчальної множини і групує подібні вектори в класи. Пред'явлення на вхід вектора з даного класу дасть визначений вихідний вектор, але до навчання неможливо зпрогнозувати, який вихід буде вироблятися даним класом вхідних векторів. Отже, виходи подібної мережі повинні трансформуватися в деяку зрозумілу форму, обумовлену процесом навчання. Це не є серйозною проблемою. Звичайно не складно ідентифікувати зв'язок між входом і виходом, установлений мережею.

Більшість сучасних алгоритмів навчання виросло з концепцій Хебба. Ним була запропонована модель навчання без вчителя, у якій синаптична сила (вага) зростає,

якщо активовані обидва нейрони, джерело і приймач. Таким чином, часто використовувані зв'язки в мережі підсилюються, і феномен звички й навчання через повторення пояснює дану технологію.

У штучної нейронної мережі, що використовує навчання по Хеббу, нарощування ваг визначається добутком рівнів збудження передавального і приймаючого нейронів. Це можна записати як

$$w_{ij}(n+1) = w(n) + \alpha OUT_i OUT_j.$$

де w_{ij} — значення ваги від нейрона i до нейрона j до налаштовування, $w_{ij}(n+1)$ — значення ваги від нейрона i до нейрона j після налаштовування, α — коефіцієнт швидкості навчання, OUT_i — вихід нейрона i та вхід нейрона j, OUT_j — вихід нейрона j.

Мережі, що використовують навчання по Хеббу, конструктивно розвивались, однак за останні 20 років були розвинуті більш ефективні алгоритми навчання. Зокрема було розвинуто алгоритми навчання з учителем, що призводять до виникнення мереж з більш широким діапазоном характеристик навчальних вхідних образів і великими швидкостями навчання, ніж ті, що використовують просте навчання по Хеббу. В даний час використовується величезна розмаїтість навчальних алгоритмів.

8.6. Архітектура штучних нейронних мереж

8.6.1. Штучні нейронні мережі прямого розповсюдження

Хоча один нейрон і здатний виконувати прості процедури розпізнавання, сила нейронних обчислень виникає від з'єднань нейронів в мережах.

Широке коло задач, що вирішує НМ, не дозволяє в даний час створювати універсальні, потужні мережі. Це змушує розробляти спеціалізовані НМ, що функціонують по різним алгоритмам.

Вибір структури НМ здійснюють відповідно до особливостей і складності задачі. Для вирішення деяких окремих типів задач вже існують оптимальні на сьогоднішній день конфігурації. Основні з них наведені на рис. 8.6.

8.6.2. Штучні нейронні мережі із зворотними зв'язками

Мережа Хопфілда

Відсутність зворотних зв'язків гарантує стійкість мереж. Проте така бажана властивість обмежує можливості мереж прямого розповсюдження в порівнянні з мережами із зворотними зв'язками. Важливий внесок як в теорію, так і в застосування систем зі зворотними зв'язками зробив Дж. Хопфілд. З цієї причини деякі з конфігурацій носять його ім'я. Структурна схема мережі Хопфілда наведена на рис. 8.7. Вона складається з єдиного шару нейронів, число яких є одночасно числом входів і виходів мережі. Кожен нейрон зв'язаний синапсами зі всією рештою нейронів, а також має один вхідний синапс, через який здійснюється введення сигналу. Вихідні сигнали, як завжди, утворюються на аксонах.

Задача, що вирішується даною мережею в якості асоціативної пам'яті, як правило, формулюється наступним чином. Відомий деякий набір двійкових сигналів (зображень, звукових оцифровок і т.д.), які вважаються зразковими. Мережа повинна вміти із довільного неідеального сигналу, поданого на її вхід, виділити ("згадати" по частковій інформації) відповідний образ (якщо такий є) або "дати висновок" про те, що вхідні дані не відповідають жодному із зразків. У загальному випадку, будь-який сигнал може бути описаний вектором $X = \{x_i : i = 0...(n-1)\}$, де n — число нейронів у мережі і розмірність вхідних та вихідних векторів.

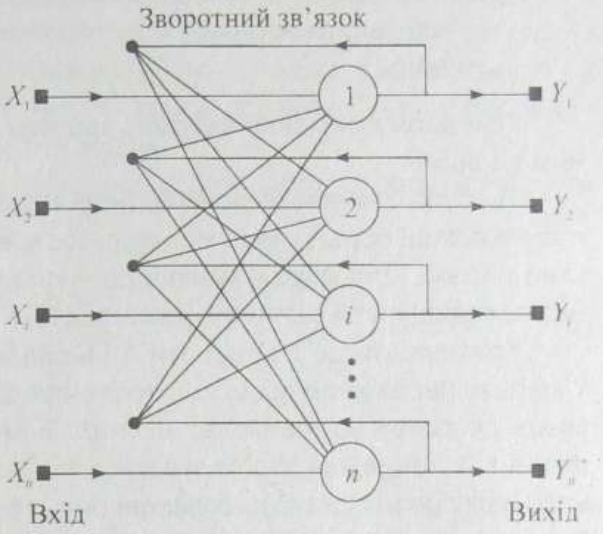


Рис. 8.7. Структурна схема мережі Хопфілда

Кожен елемент x_i дорівнює або +1, або –1. Позначимо вектор, що описує k-й зразок, як X^k, а його компоненти, відповідно, як x_i^k, $k = 0 \ldots (m-1)$, де m — число зразків. Коли мережа розпізнає (або "згадує") який-небудь зразок на основі пред'явлених їй даних, її виходи будуть містити саме його, тобто $Y = X^k$, де Y — вектор вихідних значень мережі: $Y = \{y_i : i = 0...(n-1)\}$. Інакше, вихідний вектор не співпаде з жодним зразковим. На стадії ініціалізації мережі вагові коефіцієнти синапсів визначають наступним чином:

$$w_{ij} = \begin{cases} \sum_{k=0}^{m-1} x_i^k x_j^k, & i \neq j \\ 0, & i = j \end{cases}$$

Тут i та j — індекси, відповідно передсинаптичного та післясинаптичного нейронів; $x_i^k x_j^k$ — i-й та j-й елементи вектора k-го зразка.

Розглянемо алгоритм функціонування мережі (p – номер ітерації):

1. На вхід мережі подається невідомий сигнал. Фактично його введення відбувається безпосередньою установкою значення аксонів:

$$y_i(0) = x_i, i = 0...(n-1)$$

тому позначення на схемі мережі вхідних синапсів у явному вигляді має суто умовний характер. Нуль в дужках справа від y_i означає нульову ітерацію у циклі роботи мережі.

2. Розраховується новий стан нейронів:

$$s_j(p+1) = \sum_{i=0}^{n-1} w_{ij} y_i(p), j = 0...(n-1)$$

і нові значення аксонів:

$$y_j(p+1) = f\lfloor s_j(p+1) \rfloor.$$

де f — активуюча функція у вигляді стрибка, що наведена на рис. 8.3 *а*.

3. Перевірка, чи змінилися вихідні значення аксонів за останню ітерацію. Якщо так, то перехід до пункту 2, інакше (якщо виходи стабілізувались) — кінець.

При цьому вихідний вектор є зразком, що найкращим чином поєднується з вхідними даними.

Як було зазначено раніше, іноді мережа не може виконати розпізнавання і видає на виході образ, якого не існує. Це пов'язано з проблемою обмежених можливостей мережі. Для мережі Хопфілда число образів, що запам'ятовуються, m не повинно перевищувати значення рівного $0{,}15 \cdot n$.

Крім того, якщо два образи А і Б сильно схожі, то вони, можливо, викликатимуть у мережі перехресні асоціації, тобто пред'явлення на входи мережі вектора А призведе до появи на її виходах вектора Б і навпаки. Ще одним недоліком мереж Хопфілда є їх тенденція стабілізуватися в локальному, а не в глобальному мінімумі. Цю ваду можливо подолати, в основному, за допомогою класу мереж, відомих під назвою машин Больцмана, в яких зміни станів нейронів обумовлені статистичними, а не детермінованими закономірностями.

Мережа Хеммінга

Коли немає потреби в тому, щоб мережа в явному вигляді видавала зразок (тобто достатньо, скажімо, одержувати номер зразка), асоціативну пам'ять успішно реалізує мережа Хеммінга. Дана мережа характеризується, в порівнянні з мережею Хопфілда, меншими витратами на пам'ять і об'ємом обчислень, що стає очевидним з її структури (рис. 8.8).

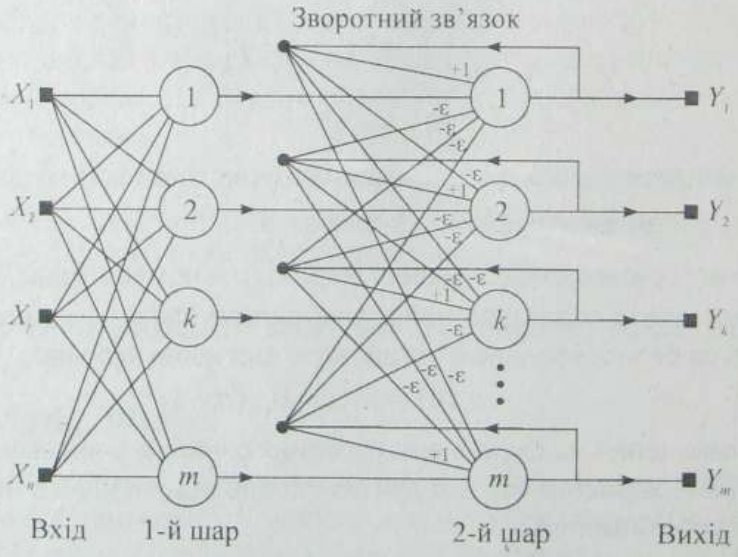


Рис. 8.8. Структурна схема мережі Хеммінга

Мережа складається з двох шарів. Перший і другий шари мають по m нейронів, де m — число зразків. Нейрони першого шару мають по n синапсів, сполучених із входами мережі (створюючи фіктивний нульовий шар). Нейрони другого шару зв'язані між собою інгибіторними (від'ємними зворотними) синаптичними зв'язками. Єдиний синапс з позитивним зворотним зв'язком для кожного нейрона сполучений з його ж аксоном.

Ідея роботи мережі полягає в знаходженні відстані Хеммінга від образу, що тестується, до всіх зразків. ***Відстанню Хеммінга*** називається число відмінних (різних) бітів в двох бінарних векторах. Мережа повинна вибрати зразок з мінімальною відстанню Хеммінга до невідомого вхідного сигналу, внаслідок чого буде активізований лише один вихід мережі, що відповідає цьому зразку.

На стадії ініціалізації ваговим коефіцієнтам першого шару і порогу активуючої функції надаються наступні значення:

$$w_{ij} = \frac{x_i^k}{2},\ i = 0...(n-1),\ k = 0...(m-1),$$

$$T_k = \frac{n}{2},\ k = 0...(m-1).$$

Тут x_i^k — i-й елемент k-го зразка.

Вагові коефіцієнти гальмуючих синапсів у другому шарі приймають рівними деякій величині $0 < \varepsilon < 1/m$. Синапс нейрона, пов'язаний з його ж аксоном, має вагу +1.

Розглянемо алгоритм функціонування мережі Хеммінга:

1. На вхід мережі подається невідомий вектор $X = \{x_i;\ i = 0\ldots(n-1)\}$, виходячи з якого розраховуються стани нейронів першого шару (верхній індекс у дужках показує номер прошарку):

$$y_j^{(1)} = s_j^{(1)} = \sum_{i=0}^{n-1} w_{ij} x_i + T_j,\ j = 0\ldots(m-1).$$

Після цього отриманими значеннями ініціалізують значення аксонів другого шару:

$$y_j^{(2)} = y_j^{(1)},\ j = 0\ldots(m-1).$$

2. Обчислюються нові стани нейронів другого шару:

$$s_j^{(2)}(p+1) = y_j(p) - \varepsilon \sum_{k=0}^{m-1} y_k^{(2)}(p),\ k \neq j,\ j = 0...(m-1)$$

та значення їх аксонів:

$$y_j^{(2)}(p+1) = f\left\lfloor s_j^{(2)}(p+1) \right\rfloor,\ j = 0...(m-1).$$

Активуюча функція f має вигляд порогу (див. рис. 8.3 *б*), причому величина f повинна бути достатньо великою, щоб будь-які можливі значення аргументу не призводили до насичення.

3. Перевірка, чи змінилися виходи нейронів другого шару за останню ітерацію. Якщо так, то перейти до кроку 2, інакше — кінець.

З алгоритму видно, що роль першого шару досить умовна: скориставшись один раз на кроці 1 значеннями його вагових коефіцієнтів, мережа більше не звертається до нього, тому перший шар може бути взагалі виключений з мережі (замінений на матрицю вагових коефіцієнтів).

Двонаправлена асоціативна пам'ять

Обговорення НМ із зворотними зв'язками, що реалізовують асоціативну пам'ять, було б неповним без хоча б короткої згадки про двонаправлену асоціативну пам'ять (ДАП) .Вона є логічним розвитком парадигми мережі Хопфілда, до якої для цього достатньо додати другий шар. Структура ДАП наведена на рис. 8.9.

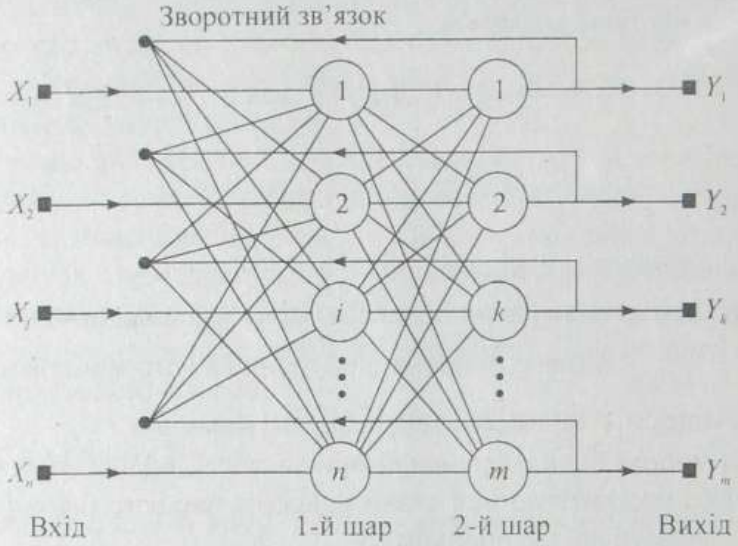


Рис. 8.9. Структурна схема ДАП

Як і мережа Хопфілда, ДАП здатна до узагальнення, виробляючи правильні реакції, не зважаючи на спотворені входи. Крім того, можуть бути реалізовані адаптивні версії ДАП, що виділяють еталонний образ із зашумлених екземплярів. Ці можливості дуже нагадують процес мислення людини і дозволяють штучним НМ зробити крок в напрямку моделювання мозку. Мережа здатна запам'ятати пари асоційованих між собою образів. Нехай пари образів записуються у вигляді векторів

$$X^k = \{x_i^k : i = 0...(n-1)\} \quad \text{та} \quad Y^k = \{y_j^k : j = 0...(m-1)\}, k = 0...(r-1),$$

де r — число пар.

Подання на вхід першого шару деякого вектора

$$P = \{p_i : i = 0...(n-1)\}$$

викликає утворення на вході другого шару деякого іншого вектора

$$Q = \{q_j : j = 0...(m-1)\}$$

який потім знову поступає на вхід першого шару.

При кожному такому циклі вектори на виході обох шарів наближаються до пари зразкових векторів, перший з яких (X) є найбільш схожим на P, що подавався на вхід мережі на початку, а другий (Y) — асоційований з ним.

Асоціації між векторами кодуються у ваговій матриці $W^{(1)}$ першого шару. Вагова матриця другого шару $W^{(2)}$ дорівнює транспонованій першій $\left(W^{(1)}\right)^T$. Процес навчання, так як і у випадку мереж Хопфілда, полягає у попередньому розрахунку елементів матриці W (і відповідно W^T) за формулою

$$w_{ij} = \sum_k x_i y_j, \; i = 0...(n-1), j = 0...(m-1). \qquad (8.3)$$

Ця формула є розгорнутим записом матричного рівняння

$$W = \sum_k X^T Y$$

для окремого випадку, коли образи записані у вигляді векторів, при цьому добуток двох матриць розміром відповідно $[n \times 1]$ і $[1 \times m]$ призводить до (8.3).

Як і мережі Хопфілда, ДАП має обмеження на максимальну кількість асоціацій, які вона може точно відтворити. Якщо цей ліміт перевищений, мережа може виробити невірний вихідний сигнал, відтворюючи асоціації, яким не навчена. При цьому передбачається, що місткість пам'яті максимізується за допомогою спеціального кодування, при якому кількість компонент із значеннями +1 дорівнює числу компонент із значеннями –1 у кожному біполярному векторі.

Не зважаючи на ці проблеми, ДАП є об'єктом інтенсивних досліджень. Основна привабливість ДАП, як і мереж Хопфілда та Хеммінга, полягає в їх простоті. Легкість побудови програмних та апаратних моделей робить ці мережі привабливими для багатьох застосувань.

Мапи Кохонена

Ще одним різновидом штучних НМ із зворотними зв'язками є мапи Кохонена, що самоорганізовуються. Така мережа є спеціальним випадком мережі, що навчається методом змагання і являє собою векторний квантувателъ, задачею якого є визначення належності вхідного вектора $X = \{x_1, x_2, ..., x_n\}^T$ до одного із M можливих кластерів, представлених векторними центрами $W = \{w_{j1}, w_{j2}, ..., w_{jn}\}^T$ $(j = 1, 2, ... M)$.

Вважається, що вектор X належить j-му кластеру, якщо відстань d_j, що визначається за формулою

$$d_j = \sum_{i=1}^{N} \left(x_i - w_{ji}\right)^2,$$

для j-го центру кластера W_j мінімальна, тобто якщо $d_j \le d_k$ для кожного $k \neq j$.

Якщо вузли квантувателя є лінійними, а вага i-го входу j -го вузла дорівнює w_{ij} для кожного i та j, то очевидно, що при відповідних значеннях порогів кожен i-й вихід мережі з точністю до неістотних постійних буде рівний евклідовій відстані d_j між пред'явленим вхідним вектором X та j-м центром кластера.

При навчанні НМ квантувателя пред'являються вхідні вектори без вказівки бажаних виходів і коректується вага згідно алгоритму, запропонованому Кохоненом.

Мапи Кохонена, що самоорганізовуються, можуть бути використані для проектування багатовимірних даних, апроксимації густини та кластеризації. Ці мережі успішно застосовуються для розпізнавання мови, обробки зображень, в робототехніці і в задачах управління.

8.7. Практичне використання нейронних мереж

У літературі зустрічається значне число ознак, якими повинна володіти задача, щоб застосування НМ було виправдано і НМ могла б її вирішити:

- відсутній алгоритм або невідомі принципи рішення задач, але накопичене достатнє число прикладів;

- проблема характеризується великими обсягами вхідної інформації;
- вхідні дані неповні або надлишкові, зашумлені, частково суперечливі тощо.

Таким чином, НМ добре підходять для розпізнавання способів й рішення задач класифікації, оптимізації та прогнозування. Нижче наведений перелік можливих застосувань нейронних мереж, на базі яких або вже створені комерційні продукти, або реалізовані демонстраційні прототипи:

- Банки та страхові компанії:
  - автоматичне зчитування чеків і фінансові документи;
  - перевірка вірогідності підписів;
  - оцінка ризику для позик;
  - прогнозування змін економічних показників.
- Адміністративне обслуговування:
  - автоматичне зчитування документів;
  - автоматичне розпізнавання штрихових кодів.
- Нафтова та хімічна промисловість:
  - аналіз геологічної інформації;
  - ідентифікація несправностей устаткування;
  - розвідка покладів мінералів за даними аерофотозйомки;
  - аналіз складу домішок;
- Воєнна промисловість та аеронавтика:
  - обробка звукових сигналів (поділ, ідентифікація, локалізація, усунення шуму, інтерпретація);
  - обробка радарних сигналів (розпізнавання цілей, ідентифікація й локалізація джерел);
  - обробка інфрачервоних сигналів (локалізація);
  - автоматичне пілотування.
- Промислове виробництво:
  - керування маніпуляторами;
  - керування процесами;
  - виявлення несправностей;
  - керування голосом.
- Служба безпеки:
  - розпізнавання осіб, голосів, відбитків пальців, райдужної оболонки ока.
- Біомедична промисловість:
  - аналіз рентгенограм;
  - виявлення відхилень у ЕКГ.

Наведений перелік далеко не повний. Щомісяця засоби масової інформації повідомляють про нові комерційні продукти на базі нейронних мереж. Так, фірма LIAC випускає апаратуру для контролю якості води. Нейросистеми фірми SAIC знаходять пластикові бомби в багажі авіапасажирів. Фахівці інвестиційного банку Citicomp (Лондон) за допомогою програмного нейропакету роблять короткострокові прогнози коливань курсів валют.

8.8. Способи реалізації нейронних мереж

Нейронні мережі можуть бути реалізовані двома шляхами: перший — це програмна модель НМ, другий — апаратна модель. На сучасному ринку вироби, засновані на використанні механізму дії НМ, спочатку з'явилися у вигляді нейроплат. В якості типового приклада нейроплати можна назвати плату MB 86232 японської компанії Fujitsu. На платі розміщений процесор цифрової обробки сигналів і оперативної пам'яті ємністю 4 Мбайт, що дозволяє використовувати таку плату для реалізації НМ, що містять до тисячі нейронів. Є і більш досконалі плати.

Головними комерційними апаратними виробами на основі НМ є і, мабуть, найближчим часом будуть залишатися нейро-ВІС. Зараз випускаються більш 20 типів нейро-ВІС, параметри яких часом розрізняються на декілька порядків. Серед них — модель ETANN фірми Intel, яка є реалізацією НМ із 64 нейронами та 10240 синапсами. До числа найдешевших нейро-ВІС відноситься модель MD 1220 компанії Micro Devices. Ця ВІС реалізує НМ із 8 нейронами та 120 синапсами.

Серед розроблювальних у даний час нейро-ВІС виділяються моделі компанії Adaptive Solutions (США) і Hitachi (Японія). Нейро-ВІС компанії Adaptive Solutions, мабуть, стане однією із самих швидкодіючих: заявлена швидкість обробки складає 1,2 млрд. з'єднань/с. Ця НМ містить 64 нейрона та 262144 синапса. Нейро-ВІС компанії Hitachi дозволяє реалізувати НМ, що містить до 576 нейронів.

Більшість сьогоднішніх нейрокомп'ютерів являють собою просто персональний комп'ютер або робочу станцію, до складу яких входить додаткова нейроплата. До їх числа відносять, наприклад, комп'ютери серії FMR компанії Fujitsu. Такі системи мають безперечне право на існування, оскільки їхніх можливостей цілком достатньо для розробки нових алгоритмів і вирішення великого числа прикладних задач методами нейроматематики. Проте найбільший інтерес викликають спеціалізовані нейрокомп'ютери, що безпосередньо реалізують принципи НМ. Типовими представниками таких систем є комп'ютери сімейства Mark компанії TRW.

8.9. Прогнозування на основі нейронних мереж

8.9.1. Основні поняття прогнозу

Прогнозування — це ключовий момент при прийнятті рішень в управлінні, від якого залежить кінцева ефективність будь-якого рішення. Тому системи планування й управління, звичайно, реалізують функцію прогнозу.

Розглянемо два приклади прогнозування:

- **Фінансове планування**. Фінансового менеджера цікавить як буде змінюватися грошовий обіг компанії з часом. Менеджер, може побажати дізнатися, в який період часу в майбутньому обіг компанії почне падати, із тим, щоб прийняти відповідне рішення вже зараз.
- **Планування нового продукту**. Рішення про розробку нового продукту звичайно потребує довгочасного прогнозу того, яким попитом він буде користуватися. Цей прогноз не менш важливий, ніж визначення інвестицій, необхідних для його виробництва.

На підставі вищевикладеного можна сказати, що ***прогнозування*** — це передбачення майбутніх подій. Метою прогнозування є зменшення ризику при прийнятті рішень. Прогноз зазвичай часто виходить помилковим, але похибка залежить від використаної системи, що прогнозує. Надаючи прогнозу більше ресурсів, можна збільшити точність прогнозу і зменшити збитки, пов'язані з невизначеністю при прийнятті рішень. При деякому рівні похибки прогнозування витрати на прогнозування мінімальні.

Оскільки прогнозування ніколи не зможе цілком знищити ризик при прийнятті рішень, необхідно визначати неточність прогнозу. Система, що прогнозує, повинна забезпечувати визначення похибки прогнозування, також як і саме прогнозування.

Необхідно відзначити, що прогнозування — це не кінцева мета. Система, що прогнозує — це частина великої системи менеджменту, і, як підсистема, вона взаємодіє з іншими компонентами системи, відіграючи чималу роль у кінцевому результаті.

8.9.2. Методи прогнозування

Методи прогнозування можна розділити на два класи:

- Методи, що ***роблять суб'єктивну оцінку***, засновану на думці експертів, що використовують результати опитувань, тести, оцінки ефективності продаж та історичні дані.
- Методи, що ***явно оголошують*** яким чином отриманий прогноз. Чітко видно логіку та зрозумілі математичні операції. Ці методи, наприклад, використовуються для дослідження історичних даних з тою метою, щоб, припустивши, що процес є стабільним, використовувати знання про нього для екстраполювання процесу в майбутнє.

Практично, прогнозуючі системи часто використовують комбінацію першого та другого типів методів.

Оцінка ефективності системи, що прогнозує

Існує ряд вимірів, що можуть бути використані для оцінки ефективності системи, що прогнозує. Серед них найбільше важливими є: точність прогнозування; вартість системи; результуюча користь; властивості стабільності.

Точність методу прогнозування визначається на основі аналізу похибки прогнозування. Якщо X — це реальне спостереження за період t, а X_t — це зроблений раніше прогноз, то похибка прогнозування за період t складає $e_t = X - X_t$.

Вартість є важливим елементом при оцінці і порівнянні методів прогнозування, її можна розділити на одноразові витрати на розробку і впровадження системи і витрати на її експлуатацію. Що стосується витрат на експлуатацію, то різні процедури, що прогнозують, можуть дуже сильно відрізнятися по вартості отримання даних, ефективності обчислень та рівню дій, необхідних для підтримки системи.

Користь прогнозу в поліпшенні прийнятих рішень залежить від способу прогнозування та форми прогнозу також як і від його точності.

Прибуток повинен вимірюватися для усієї системи керування як єдиного цілого, а прогнозування — лише один елемент цієї системи.

8.9.3. Застосування нейронних мереж у фінансовій сфері

Характерний приклад успішного застосування нейронних обчислень у фінансовій сфері — управління кредитними ризиками. До видачі кредиту банки проводять складні статистичні розрахунки щодо фінансовій надійності позичальника, щоб оцінити можливість власних збитків від невчасного повернення фінансових коштів. Такі розрахунки зазвичай базуються на оцінці кредитної історії, динаміці розвитку компанії, стабільності її головних фінансових показників та багатьох інших факторів.

Один широко відомий банк США випробував метод нейронних обчислень і прийшов до висновку, що та ж задача по вже виконаних розрахунках подібного роду вирішується швидше й точніше. Наприклад, в одному із випадків оцінки 100 тис. банківських рахунків нова система, побудована на базі нейронних обчислень, визначила понад 90% потенційних неплатників.

Інша дуже важлива область застосування нейронних обчислень у фінансовій сфері — передбачення ситуації на фондовому ринку. Стандартний підхід до цієї задачі базується на жорстко фіксованому наборі "правил гри", що згодом втрачають свою ефективність через зміну умов торгів на фондовій біржі. Крім того, системи, побудовані на основі такого підходу, виявляються занадто повільними для ситуацій, що потребують миттєвого прийняття рішень.

Саме тому головні японські компанії, що оперують на ринку цінних паперів, вирішили застосувати метод нейронних обчислень. У типову систему на базі нейронної мережі ввели інформацію загальним обсягом у 33 роки ділової активності декількох організацій, включаючи обіг, попередню вартість акцій, рівні прибутку і т. ін. Самонавчаючись на реальних прикладах, система нейронної мережі показала велику точність передбачення і кращу швидкодію: у порівнянні зі статистичним підходом дала поліпшення результативності в цілому на 19%.

Ще один приклад, досить близький до області фінансового ринку, — оцінка вартості нерухомості. Вирішення цієї задачі залежить, головним чином, від досвіду співробітника агенції нерухомості, що враховує множину таких нерівноцінних факторів, як частка власності, якість будівництва, навколишнє оточення і т.ін. Група дослідників з університету міста Портсмут (Великобританія) заклала в обчислювальну систему на базі нейронної мережі дані по оцінці нерухомості з оглядів агенцій та списків аукціонних цін. Результат показав, що система, яка самонавчилась, дає оцінки вартості, які добре корелюють з експертними висновками фахівців цього профілю.

Приклад вдалого прогнозування динаміки біржових курсів за замовленням Chemical Bank продемонструвала фірма Logica. На технічній базі Sun SPARCstation LX за допомогою нейронних обчислень моделювалися ринки валютних курсів долар/швейцарський франк і німецька марка/швейцарський франк. Вибір саме цих валют пояснювався високим рівнем рухливості першого співвідношення і малим — другого. Дані про динаміку крос-курсів цих валют збиралися протягом двох років, при цьому цінові прогнози характеризувалися п'ятьма категоріями: великий ріст, малий ріст, без змін, малий спад, великий спад. В результаті нейронна система передбачила за вищезгаданий річний період 55% реальних даних по першому співвідношенню валют і 23 % — по другому.

Розділ 9

Віртуальні мережі

"По-справжньому освічений той, хто вміє читати між рядками"
(Народна мудрість)

9.1. Проблеми класичних комп'ютерних мереж

Першоджерелом проблем при експлуатації комп'ютерних мереж, побудованих на базі класичних технологій, є принципи, на яких ці технології засновані: розподіл середовища (Ethernet) або розподіл доступу до середовища (Token Ring, FDDI). Вони активно використовують повідомлення, які породжують досить інтенсивний трафік, що понижує корисну пропускну спроможність мережі.

Класична мережева структура складається із сегментів, що базуються на простих концентраторах та взаємодіють через мости та маршрутизатори. При цьому кожний елемент структури є місцем виникнення заторів: сегмент — в результаті колізій (сутичок) (Ethernet, Fast Ethernet) або збільшення часу очікування (Token Ring, FDDI); мости та маршрутизатори — з причини їх обмеженої пропускної спроможності.

Відповідаючи на потреби, які постійно збільшуються, щодо підвищення продуктивності мережі, класична структура формується в структуру, що комутується, при якій кількість станцій в сегменті зводиться до одної, а функції мостів та маршрутизаторів покладаються на багатопортові комутатори.

Але мережа все одно складається з вузлів — "перевалочних пунктів", де для кожного пакета, що надходить, вирішується питання, куди він повинен бути відправлений. Це — принцип поштової системи: опустивши лист у поштову скриньку, ми більше не турбуємося про те, як він дійде до одержувача. Замість нас цим займаються поштові відділення, кожний раз звіряючи адресу призначення із довідником та вирішуючи, куди далі повинна бути відправлена кореспонденція.

Крім того, оскільки логічна структура мережі фактично точно повторює її фізичну структуру, територіальні переміщення користувачів зв'язані із значними складностями.

Початкове рішення цих проблем — застосування технології ATM (Asynchronous Transfer Mode — асинхронний режим передачі). У протилежність технологіям, заснованим на розділі доступу, у ATM з'єднання між кінцевими точками встановлюється один раз, причому ще до того, як буде передаватись потік інформації. Після встановлення маршруту вже не потрібно на кожному вузлі вирішувати, куди направити наступну порцію даних: весь потік пройде без затримки від джерела до приймача, завдяки чому отримують гарантовану швидкість передачі (сьогодні — від 25 до 622 Мбіт/с) та ізохронний потік даних.

Друга важлива властивість ATM полягає в призначенні гарантованої смуги пропуску для кожного віртуального з'єднання, причому це може бути реалізовано як статично (шляхом прямого втручання адміністратора), так і динамічно — в залежності від реальних потреб джерела даних.

Фізична структура мережі може не мати нічого спільного з її логічною структурою, і переміщення користувачів не являє собою ніяких проблем. Технологія ATM однаково підходить для створення як локальних, так і територіальних мереж, між якими вже немає ніякої особливої різниці, оскільки, використовуючи єдину технологію та ідентичні апаратні та програмні засоби, вони разом складають гомогенну структуру.

Безумовно, майбутнє — за ATM, але якими б різноманітними властивостями не характеризувалась ця технологія, це зовсім не означає, що господарі вже існуючих комп'ютерних мереж, побудованих на традиційних технологіях, повинні негайно від них відмовитись. Вони можуть і будуть працювати ще досить довго, тим більше що далеко не всі їх можливості вичерпані.

9.2. Призначення віртуальних мереж

Окрім свого основного призначення — підвищення пропускної спроможності зв'язків в мережі — комутатор дозволяє локалізувати потоки інформації у мережі, а також контролювати ці потоки і керувати ними, використовуючи фільтри користувачів. Однак такий фільтр може заборонити передачу кадрів лише за конкретними адресами, а трафік широкого розповсюдження, він передає всім сегментам мережі. Так вимагає алгоритм роботи мосту, що реалізований в комутаторі, тому мережі, створені на основі мостів та комутаторів інколи називають плоскими з причини відсутності бар'єрів на шляху трафіку широкого розповсюдження.

Технологія віртуальних мереж (Virtual LAN, VLAN) дозволяє подолати вказане обмеження. ***Віртуальною мережею*** називається група вузлів мережі, трафік якої, в тому числі і широкого розповсюдження, на канальному рівні повністю ізольований від інших вузлів мережі. Це означає, що передача кадрів між різними віртуальними сегментами на підставі адреси канального рівня неможлива, незалежно від типу адреси — унікальної, групової або широкого розповсюдження. В той же час всередині віртуальної мережі кадри передаються за технологією комутації, тобто на той порт, що зв'язаний з адресою призначення кадру.

Говорять, що віртуальна мережа утворює домен трафіку широкого розповсюдження (*broadcast domain*), по аналогії з доменом колізій, який утворюється повторювачем мереж Ethernet.

Призначення технології віртуальних мереж полягає в полегшенні процесу створення незалежних мереж, що після цього повинні зв'язуватись за допомогою протоколів мережевого рівня. Для вирішення цієї задачі до появи технології віртуальних мереж використовувались окремі повторювачі, кожний з яких утворював незалежну мережу. Після цього ці мережі зв'язувались маршрутизаторами в єдину інтермережу.

При зміні складу сегментів (перехід користувача в іншу мережу, ділення великих сегментів) потрібно здійснювати фізичну перекомутацію з'єднувачів на передніх панелях повторювачів або в кросових панелях, що не дуже зручно у великих мережах — багато фізичної роботи, до того ж висока ймовірність помилки. Тому для усунення необхідності фізичної перекомутації вузлів стали застосовувати багатосегментні повторювачі. У найбільш досконалих моделях таких повторювачів приписування окремого порту до будь-якого з внутрішніх сегментів здійснюється програмним шляхом,

зазвичай з допомогою зручного графічного інтерфейсу. Прикладами таких повторювачів можуть служити концентратор Distributer 5000 компанії Bay Networks та концентратор PortSwitch компанії 3Ccom.

Програмне приписування порту сегменту часто називають статичною або ***конфігураційною комутацією***. Однак вирішення задачі зміни складу сегментів за допомогою повторювачів накладає деякі обмеження на структуру мережі — кількість сегментів такого повторювача зазвичай невелика, тому виділити кожному вузлу свій сегмент, як це можна зробити за допомогою комутатора, нереально. Тому мережі, побудовані на основі повторювачів із конфігураційною комутацією, досі засновані на розподілі середовища передачі даних між великою кількістю вузлів і, отже, володіють значно меншою продуктивністю у порівнянні з мережами, побудованими на основі комутаторів.

При виконанні технології віртуальних мереж на комутаторах водночас вирішуються дві задачі:

- підвищення продуктивності в кожній з віртуальних мереж, бо комутатор передає кадри в такій мережі лише вузлу призначення;
- ізоляція мереж одна від одної для управління правами доступу користувачів і створення захисних бар'єрів на шляху штормів широкого розвосюдження.

Для зв'язку віртуальних мереж в інтермережу вимагається залучення мережевого рівня. Він може бути реалізований в окремому маршрутизаторі, а може працювати і у складі програмного забезпечення комутатора.

Технологія VLAN є одним з найбільш важливих аспектів комутованих мереж. Віртуальні LAN є необхідним елементом комутованих мереж, забезпечуючи перехід від мереж з середовищем, що поділяється, до повністю комутованих систем.

В сучасних мережах чітко відслідковується тенденція переходу від систем з середовищем, що працюють на базі концентраторів, до застосування комутаторів для сполучення робочих станцій та мережевих ресурсів. Однак комутатори працюють на канальному рівні моделі OSI та є приладами пересилки кадрів, що неспроможні забезпечити достатньо ефективне масштабування великих мереж.

В традиційних мережах на основі концентраторів та маршрутизаторів користувачі групуються в домени широкого розповсюдження (сегменти), для сполучення яких служать маршрутизатори. Таке рішення дозволяє усім користувачам в групі розподіляти смугу свого концентратора або кільця і знижувати рівень насичення смуги і число колізій за рахунок зменшення числа користувачів у кожному сегменті.

Комутовані мережі не мають доменів широкого розповсюдження, тому продуктивність мережі знижується за рахунок трафіку широкого розповсюдження, особливо при використанні таких протоколів як IPX. Віртуальні LAN дозволяють організувати в комутованій мережі домени широкого розповсюдження, забезпечуючи завдяки цьому можливість ефективного масштабування.

Віртуальні LAN мають й інші переваги: вони дозволяють більш ефективно використовувати IP-адреси, істотно спрощують перенесення станцій або ресурсів в мережах, забезпечують більш ефективну організацію користувачів по групам. По мірі розвитку комутаційних технологій будуть запропоновані численні варіанти організації комутованих мереж. Різні варіанти побудови VLAN мають свої переваги й недоліки.

Комутатори в WLAN забезпечують високу продуктивність за рахунок процесів, подібних з тими, що використовуються в традиційних мостах. Комутатори використовують адреси канального рівня (MAC-адреси) для визначення порту, в який потрібно передати пакет. Робота з адресами канального рівня на багато простіша й швидкіша у порівнянні з маршрутизацією, що використовує протокол мережевого рівня.

На відміну від маршрутизації, надіслання кадрів на канальному рівні (комутація) не вимагає зміни вмісту кадрів, тоді як маршрутизатори вимушені додавати до пакетів адреси й лічильник інтервалів. Крім того, маршрутизатори при переміщенні даних по мережі повинні підтримувати ще цілий ряд функцій: вибір маршруту, перетворення адрес тощо. Не треба забувати і про те, що в більшості мереж використовується декілька протоколів, отже маршрутизатор повинен забезпечувати роботу з усіма протоколами і передавати дані для різних протоколів через мережу.

У використанні приладів канального рівня для побудови мереж немає нічого нового. Перші VLAN створювались на основі простих мостів. Основна відмінність комутаторів від традиційних мостів полягає в продуктивності обробки пакетів і підтримці інтенсивних потоків трафіку. Однак усім комутованим мережам притаманне одне обмеження. Оскільки комутатор не має справи з протоколами мережевого рівня, він не може знати куди спрямувати пакети широкого розпосвюдження. Хоча трафік з конкретними адресами (сполучення "точка-точка") ізольований парою портів, пакети широкого розпосвюдження передаються в усю мережу (кожний порт).

Те ж саме відбувається і у випадку з пакетами, призначеними станції з невідомою MAC-адресою, — такі пакети просто передаються всім станціям. Пакети широкого розпосвюдження та пакети з невідомими адресами можуть призвести до перенасичення смуги та зростанню числа колізій, особливо у великих мережах.

В деяких мережах середніх розмірів з невисоким рівнем трафіку широкого розпосвюдження кадри, що передаються всім (*flooding* або лавинна маршрутизація), майже не впливають на картину трафіку — число пакетів широкого розпосвюдження невелике, а невідомі адреси достатньо швидко розпізнаються комутатором. Однак у великих мережах (або невеликих, але з високим рівнем трафіку широкого розпосвюдження) ситуація може виявитися цілком іншою і розсилка пакетів широкого розпосвюдження в усі порти може звести нанівець усі переваги комутації. Для того щоб цього не відбувалось, важливо обмежити область розповсюдження трафіку — організувати невеликі домени широкого розпосвюдження або віртуальні LAN.

В мережах на базі маршрутизаторів кожен порт маршрутизатора є доменом широкого розпосвюдження. Для комутаторів вимагається можливість підтримки трафіку широкого розпосвюдження без зниження продуктивності. Для того щоб добитися цього, виробники комутаторів пропонують створювати VLAN, що є просто доменами широкого розпосвюдження. Віртуальні мережі забезпечують сегментацію за рахунок створення логічних, динамічних доменів широкого розпосвюдження. Більш того, віртуальні LAN істотно підвищують можливості масштабування мереж.

9.3. Вимоги до віртуальних мереж

Усі виробники комутаторів розуміють необхідність підтримки віртуальних LAN, тому більшість комутаторів дозволяє організувати VLAN. Однак реалізації віртуаль-

них мереж дуже відрізняються і практично несумісні (зокрема, різні реалізації одного виробника несумісні одна з одною). Які існують вимоги до віртуальних мереж?

9.3.1. Підтримка різнотипних середовищ

Віртуальна LAN (пов'язані з нею комутатори) повинна підтримувати різноманітні типи фізичних середовищ. В комутованих мережах можлива робота централізованих ресурсів (магістралей) з більш високими швидкостями, ніж швидкість робочих станцій. Наприклад, робочі станції Ethernet (10 Мбіт/с) можуть працювати з серверами Fast Ethernet, Gigabit Ethernet або ATM. Адміністратор мережі повинен бути впевнений, що VLAN можна організувати для всіх типів в організації мережевих середовищ з урахуванням перспектив розвитку.

9.3.2. Комутатори та концентратори

В минулому мережі будувалися на основі концентраторів і маршрутизаторів. Величезне число таких мереж неможливо разом перевести на комутаційні технології. Навіть припустивши, що такий перехід можливий, його не слід негайно реалізувати — не кожному користувачу потрібна вся смуга, що надається виділеним портом комутатора.

Користувачів, яким виділені порти не потрібні, можна згрупувати за допомогою традиційних концентраторів, виділяючи для зв'язку концентратора з мережею окремий порт комутатора. Така схема схожа на структуру традиційних мереж, де для об'єднання концентраторів використовуються комутатори замість маршрутизаторів. Кожний порт комутатора повинен забезпечувати підтримку більш, ніж однієї віртуальної LAN. Це актуально навіть у таких випадках, коли до порту комутатора підключаються безпосередньо робочі станції (одна станція може бути присутня у декількох віртуальних мережах).

9.3.3. Об'єднання комутації та маршрутизації

Деякі комутатори VLAN здатні виконувати функції стандартної маршрутизації на мережевому рівні (IP та IPX). Така можливість дозволяє організувати обмін даними між віртуальними LAN без використання зовнішніх маршрутизаторів. Однак у сучасних мережах уже встановлене величезне число маршрутизаторів, і ніхто не схоче від них відмовлятись на користь нових приладів.

Одним з засобів (можливо, найкращим) використання наявних в організаціях маршрутизаторів є зв'язок між віртуальними LAN. В цьому випадку маршрутизатори роблять те, для чого вони призначені — маршрутизують міжмережевий трафік.

9.3.4. Можливість увімкнення серверів у декілька VLAN

Подібно робочим станціям, сервери за останні п'ять років стали значно потужнішими. Численні робочі станції зараз підключаються до потужних, високопродуктивних серверів, а дані зазвичай передаються через мережеві магістралі, оскільки сервер і станція знаходяться в різних сегментах. Передача трафіка між станціями та серверами через магістралі може призводити до виникнення магістральних пробок. Крім того, при обробці кадрів маршрутизатором виникає затримка. В ефективних ре-

алізаціях віртуальних мереж сервери можуть входити в декілька VLAN. Трафік у такому випадку не передається через маршрутизатор або магістраль. Таким чином знижується навантаження на мережеві магістралі і зменшується затримка.

9.3.5. Підключення станцій до декількох VLAN

Деякі робочі станції необхідно підключити до декількох віртуальних мереж одночасно. Наприклад, при створенні віртуальних мереж по підрозділам підприємства віце-президенту компанії може знадобитись доступ до груп всіх підпорядкованих йому підрозділів (таким чином, робоча станція віце-президента повинна бути включена у віртуальні мережі відповідних підрозділів).

Іншим прикладом може бути ситуація, коли робочій станції потрібен доступ до файлового серверу Novell NetWare та протоколу TCP/IP для роботи з сервером програм на базі UNIX. Ці два ресурси можуть бути розташовані в різних віртуальних мережах, а, отже, станції знадобиться доступ в обидві мережі. При такому рішенні трафік TCP/IP не буде впливати на віртуальну мережу IPX і навпаки — трафік IPX не вплине на програми TCP/IP.

9.3.6. Мережі на базі декількох комутаторів

Переваги віртуальних мереж дуже знижуються в тих випадках, коли VLAN неможливо організувати на базі декількох комутаторів. Можливість розширення віртуальних мереж за межі одного комутатора особливо важлива для мереж середніх та великих організацій, де число комутаторів може бути досить великим. В багатьох випадках може знадобитись об'єднання в одну віртуальну мережу робочих станцій або серверів, підключених до різних комутаторів.

9.3.7. ATM

Стандарт ATM забезпечує підтримку різноманітних типів трафіку (дані, аудіо-, відео- та ін.). Крім того, ця технологія дозволяє організувати мережі значної довжини (у масштабі міста, країни і т. ін.). Швидкість ATM може знаходитись у діапазоні від 56 Кбіт/с до 622 Мбіт/с (у перспективі можливі і більш високі швидкості).

В нинішній час ця технологія є достатньо дорогою. Вже зараз ATM усе ширше використовується для підключення швидкісних серверів. Якщо ATM використовується для магістралей, то слід пам'ятати, що зручні сьогодні варіанти комутаторів можуть виявитися цілком непридатними завтра. Важливо, щоб механізми організації віртуальних мереж змогли ефективно працювати через магістралі ATM і забезпечувалась підтримка емуляції VLAN та технології MPOA (Multi-Protocol Over ATM).

9.3.8. Додавання та перенесення станцій

В багатьох мережах обладнання досить часто переміщують з одного місця на інше в межах будинку або території об'єкту. Адміністратор мережі повинен мати можливість зв'язати прилад або користувача з віртуальними мережами незалежно від місцеположення.

9.3.9. Швидкість роботи

Використання комутаторів звичайно пов'язане з необхідністю підвищення продуктивності мережі при одночасному зниженні видатків на обладнання. Організація віртуальних LAN не повинна знижувати продуктивність мережі.

9.4. Особливості мережевих технологій для VLAN

9.4.1. Мости та маршрутизатори

Мости та маршрутизатори є приладами, що дозволяють з'єднувати локальні мережі між собою і з WAN-мережами. Таке з'єднання дає користувачам можливість розподілу даних і ресурсів в локальних і глобальних мережах. Функції мостів і маршрутизаторів схожі, однак працюють ці прилади по-різному.

Мости передають дані з однієї мережі в іншу, не змінюючи вмісту вихідних кадрів LAN. Вони забезпечують сегментацію з однієї мережі, що дозволяє вирішити проблему насичення смуги. Однак мости неприйнятні для великих мереж, оскільки вони пропускають трафік широкого розповсюдження і всі пакети з невідомими адресами. Проблему трафіку широкого розповсюдження спочатку вирішували за допомогою маршрутизаторів, а зараз для цього застосовуються комутатори типу OmniSwitch корпорації Xylan.

На відміну від мостів, маршрутизатори вносять зміни в прийняті кадри VLAN і передають їх після обробки в мережу, де розміщений адресат. Маршрутизатори мають переваги перед мостами з точки зору обробки трафіку широкого розповсюдження, дозволяючи ізолювати його всередині однієї VLAN. Маршрутизатори також можуть бути біль ефективні, ніж мости, в частині використання WAN-каналів. Однак маршрутизатори зазвичай не можуть забезпечити потреб сучасних програм з точки зору забезпечення пропускної спроможності і якості обслуговування.

Одним з рішень проблеми обмеження смуги є ділення мереж на сегменти та кільця за допомогою мультисегментних концентраторів та підвищення продуктивності маршрутизаторів. Однак таке рішення буде достатньо дорогим і складним в управлінні. Більш ефективним рішенням є використання комутаторів сполучення "точка-точка" між будь-якими парами мережевих приладів, що в багато разів збільшують інтегральну пропускну спроможність мережі.

Управління комутаторами значно простіше у порівнянні з маршрутизаторами. Комутатори OmniSwitch корпорації Xylan забезпечують підтримку обох технологій (комутації та маршрутизації), дозволяючи організувати комутацію трафіка в рамках віртуальних LAN та маршрутизацію між ними.

9.4.2. Комутація

Комутатори забезпечують істотне підвищення продуктивності при незначній вартості. За рахунок цього технології комутації починають частіше використовуватися замість традиційних рішень на базі концентраторів та мостів (а в деяких випадках — і машрутизаторів), особливо при організації нових мереж і розширенні існуючих.

На основі досліджень, проведених в американських організаціях, з'ясувалося, що біля 22,5% користувачів мережі на протязі року переміщуються. Такі масові переміщення користувачів ведуть до значних видатків на реорганізацію кабельних систем і повторне налаштування мереж. На щастя, використання комутаторів і віртуальних LAN дозволяє вирішити цю проблему. При переміщенні користувача VLAN автоматично визначає його нове місцезнаходження та забезпечує можливість продовження роботи користувача без зміни кабельної системи параметрів конфігурації мережі.

Комутатори також забезпечують можливість простого підключення до магістралей переходу на режим асинхронної передачі (ATM) підключення до магістралей FDDI. В доповнення до невисокої ціни, комутатори в багато разів підвищують продуктивність мереж за рахунок забезпечення виділених високошвидкісних сполучень, а також істотно підвищують рівень забезпечення безпеки. Прикладні програми, які потребують високої пропускної спроможності, більше не будуть уповільнюватись через нестачу смуги концентраторів і значних затримок в мережі. Локалізація трафіку широкого розповсюдження і прямі сполучення "точка-точка" забезпечують високий рівень безпеки в мережі.

Комутація є оптимальним рішенням, що забезпечує високу пропускну спроможність і малі затримки, тоді як маршрутизація дозволяє проектувати мережі з брандмауерами і позбавляє необхідності передачі трафіку широкого розповсюдження через низькошвидкісні WAN-канали.

Наприклад, комутатори OmniSwitch поєднують в собі кращі характеристики обох методів:

- використання комутації всередині віртуальних LAN забезпечує високошвидкісний зв'язок між станціями й серверами;
- маршрутизація між віртуальним LAN дозволяє поділити ресурси і скоротити до мінімуму трафік широкого розповсюдження.

9.5. Переваги віртуальних мереж

Віртуальні мережі забезпечують можливість створення логічних груп користувачів в масштабі корпоративної мережі. За рахунок використання VLAN адміністратор мережі може організувати користувачів у логічні групи незалежно від фізичного розташування робочих станцій цих користувачів. Це одне з основних досягнень мережевих технологій — можливість створювати робочі групи на основі службових функцій користувачів, не прив'язуючись до мережевої топології.

Віртуальні мережі дозволяють організувати роботу в мережі більш ефективно і забезпечують цілий ряд переваг:

- простота внесення змін в мережу, додання або вилучення приладів;
- більш ефективне використання обмежених ресурсів;
- відносно високий рівень забезпечення безпеки.

Можливість організації віртуальних LAN зумовлена переходом від відокремлених середовищ до комутованих.

Наприклад, корпорація Xylan розробила архітектуру комутаторів OmniSwitch та програми для них таким чином, щоб забезпечити ефективну підтримку віртуальних

мереж. Центральним елементом технології віртуальних мереж Xylan є технологія AutoTracker, що дозволяє задати для включення прилади у VLAN на основі інформації з будь-якої частини заголовку пакетів, або кадрів.

Надаючи десятки засобів автоматичного визначення VLAN, комутатори OmniSwitch забезпечують високопродуктивні можливості, що не мають аналогів. Одна мережа Auto Tracker VLAN може включати в себе прилади з різнотипними MAC-адресами. Робочі станції Ethernet та Token Ring можуть отримати доступ до магістралей і хостів FDDI і ATM, кожна віртуальна мережа може містити безліч комутаторів. Оскільки в усіх комутаторах реалізовані однакові правила, при переміщенні робочої станції або сервера в мережі його положення у віртуальних мережах зберігається автоматично.

9.5.1. Технологія ATM

Технологія ATM (Asynchronous Transfer Mode — асинхронний режим передачі) є технологією комутації, призначеною для одночасної передачі голосу і даних в формі пакетів. ATM організує дані в короткі проміжки фіксованої довжини. Використання коротких проміжків зменшує час на обробку і дозволяє забезпечити більш рівномірне завантаження процесора. Короткий час процесорної обробки тактів фіксованої довжини дозволяє забезпечити ефективне, високошвидкісне управління змішаним трафіком "голос/дані", оскільки в ATM для коммутації використовуються спеціалізовані контроллери (мікросхеми). При інтеграції з ISDN-технологією ATM може забезпечувати перенесення даних зі швидкістю 1,5 Мбіт/с, максимальна швидкість ATM перевищує 600 Мбіт/с.

Потужні технології коммутації засновані на використанні одного загальноприйнятого стандарту. Така стандартизація забезпечує сумісність обладнання та постійне зниження цін на устаткування ATM. Наприклад, комутатори OmniSwitch корпорації Xylan забезпечують економічне ефективне переведення мереж на використання технологій ATM.

Підтримка спеціальних транкових протоколів в комутаторах OmniSwitch дозволяє організувати віртуальні LAN через будь-які магістралі, включаючи ATM. Наприклад, при використанні комутаторів OmniSwitch в мережі з магістраллю FDDI та плануванні згодом перейти на ATM, потрібно:

- додати комутатор ATM;
- установити один чи декілька комутаційних модулів ATM в комутатори OmniSwitch;
- підключити деякі сервери до комутатору ATM, залишивши інші в мережі FDDI;
- поступово переносити файли і програми з серверів FDDI на сервери ATM.

Останні роки виробники комутаційного обладнання для локальних мереж мали надію, що з упровадженням технологій віртуальних локальних мереж буде набагато простіше управляти мережею. Наскільки ж віртуальні LAN відповідають даним обіцянкам? Цілком, коли справа стосується об'єднання в логічні групи розрізнених кінцевих користувачів в комутованих мережах, управління доменами широкого розповсюдження і спрощення адміністративних задач.

Однак, якщо йдеться про здійснення більш складних функцій управління мережею (наприклад, про визначення віртуальних локальних мереж по типу прикладної

програми), то виявляється, що технологія не виправдала очікування користувачів. Крім того, деякі експерти заявляють, що управляти VLAN настільки складно, що організаціям, в яких число користувачів перевищує 500, навіть і думати про застосування цієї технології не варто.

Віртуальні LAN мають ряд технічних переваг — в цьому згодні і експерти, і користувачі. Вони підходять як найкраще, якщо задача полягає в мінімізації багатоадресного трафіку або трафіку широкого розповсюдження. Іншою важливою перевагою VLAN є спрощення переміщень, додавань і змін. Вони не випускають за свої межі кадри широкого розповсюдження, передаючи їх лише користувачам своїх підмереж, а це звільняє "незацікавлені" хости інших віртуальних мереж від обробки не їхніх кадрів. Також VLAN дадуть додаткову гнучкість в здійсненні додавань, переміщень і змін, завдяки спрощенню процесу перерозподілу членства користувачів при їх переміщеннях в межах мережі.

Інша важлива перевага — можливість централізації серверів. Віртуальна LAN дозволяє адміністраторам мереж встановити сервери в одному місці, що спрощує управління ними і дозволяє користувачам, що знаходяться в різних місцях, одержувати доступ до серверів через VLAN.

9.6. Недоліки віртуальних мереж

На думку галузевих експертів, VLAN не відповідають своїому призначенню, коли справа стосується більш складних функцій. Наприклад, ті замовники, які хочуть інтегрувати логічні підмережі з прикладними програмами, що використовуються, цього не отримають.

В теперішній час більша частина комутаторів визначає VLAN за портами, MAC-адресами, мережевими адресами або протоколами. Якщо потрібен зв'язок VLAN з прикладним ПЗ (Lotus Notes, Web-браузерами, електронною поштою і т. ін.), то тут без допомоги постачальника програм не обійтись. Виробники комутаторів потребують співробітництва з такими компаніями, як наприклад Lotus Development та інші.

Недолік потужних функцій зумовлює складність управління комутованими віртуальними мережами. Однак інститут IEEE повинен прийняти відповідний стандарт, щоб постачальники почали створювати взаємодіючі програми для управління VLAN. Інструменти управління локальними мережами повинні уміти зіставляти події на різних маршрутизаторах, щоб адміністратору не треба було думати про те, що деяка інформація дублює іншу.

Стандарт IEEE 802.1 q VLAN буде визначати формати кадрів VLAN, правила членства і процедури управління. Віртуальні локальні мережі надають лише міст до ATM. Не викликає сумнівів, що виробники будуть пропонувати комутатори з каналами ATM. В них VLAN визначені у відповідності з існуючими системами, і вони транслюються комутатором у віртуальні мережі ATM.

9.7. Комплексний підхід до реалізації VLAN

Отримання максимального виграшу від якісної зміни традиційних мереж можливе лише при прийнятті комплексних заходів та обґрунтованої, стратегічної програми,

яка регулює процес переходу до більш досконалих мережевих рішень. Саме такою стратегічною програмою є, наприклад, Synthesis фірми Cabletron Systems.

Це комплексне рішення дозволяє послідовно звільнитись від основних недоліків класичних мережевих структур, не відмовляючись від стандартних мережевих технологій, але наділяючи їх властивостями найновішої технології ATM. Його основи були закладені вже досить давно і враховувались при створенні нового обладнання Cabletron Systems, яке початкове було орієнтоване не тільки на виконання класичних функцій, але і на реалізацію технологій майбутнього, що забезпечує довгу тривалість життя пристрою та ефективність виробничих витрат.

Synthesis — це стратегічна програма, яка охоплює сукупність продуктів та технологій, засобів автоматизованого управління та послуг по підтримці. Вона слугує основою для перетворення сьогоднішніх мереж, основаних на використанні маршрутизаторів, в гомогенні комутовані структури.

9.7.1. PLUS-архітектура

Фізична побудова та функціонування всіх інтелектуальних пристроїв Cabletron Systems заснована на єдиних принципах, відомих під назвою ***PLUS-архітектура***. Логіка всіх найновіших продуктів виконується на базі комплекту спеціалізованих інтегрованих мікросхем (ASIC — Application Specific Integrated Circuits), що керується RISC-процесорами. Обладнання має гнучку модульну структуру, яка легко трансформується для обслуговування будь-яких мережевих технологій і типів використаних фізичних засобів.

9.7.2. SecureFast Switching

За допомогою спеціального програмного продукту фірми Cabletron Systems досягається якісна зміна принципу обміну даними у традиційних LAN. Реалізацією Virtual Network Services на платформі PLUS-архітектури є SecureFast Switching (SFS). Ця технологія орієнтована на попереднє встановлення з'єднань в комутованій системі, побудованій на базі комутаторів ATM-комірок та/або пакетів. Вона бере у ATM процедури управління з'єднаннями на MAC-рівні та реалізує маршрутизацію протоколів третього рівня.

Технологія SFS підтримує всі транспортні протоколи та функціонально повністю сумісна з традиційними мостами та маршрутизаторами, для яких, як і для своїх клієнтів, виступає традиційною мережею, що складається із сегментів, з'єднаних маршрутизаторами, хоча насправді її внутрішнє функціонування досить відрізняється від традиційного.

SFS-мережа (рис. 9.1) містить два основні компоненти: пакетний комутатор (Packed Switch) та віртуальний мережевий сервер (Virtual Network Server). Логічна структура мережі — організація логічних сегментів (робочих груп) та взаємодії між ними — встановлюється програмно та підтримується Virtual Network Server. Це може бути реалізовано як засобами локального менеджменту, так і системою мережевого управління Spectrum.

SecureFast Packet Switch — це звичайний комутатор, у який завантажений SFS-агент, що являє собою дуже невеликий код для організації взаємодії з Virtual Network

Server. Функції Virtual Network Server реалізуються окремим пристроєм, а також можуть бути покладені на один із комутаторів або достатньо потужну станцію мережі.

Комутатори SecureFast динамічно вивчають інформацію, яка надходить із мережевого оточення, поновлюють свої власні таблиці відповідностей та передають цю інформацію до Virtual Network Server, який на цій основі поновлює свою базу даних і, спілкуючись з іншими такими ж серверами, має інформацію про всю мережу.

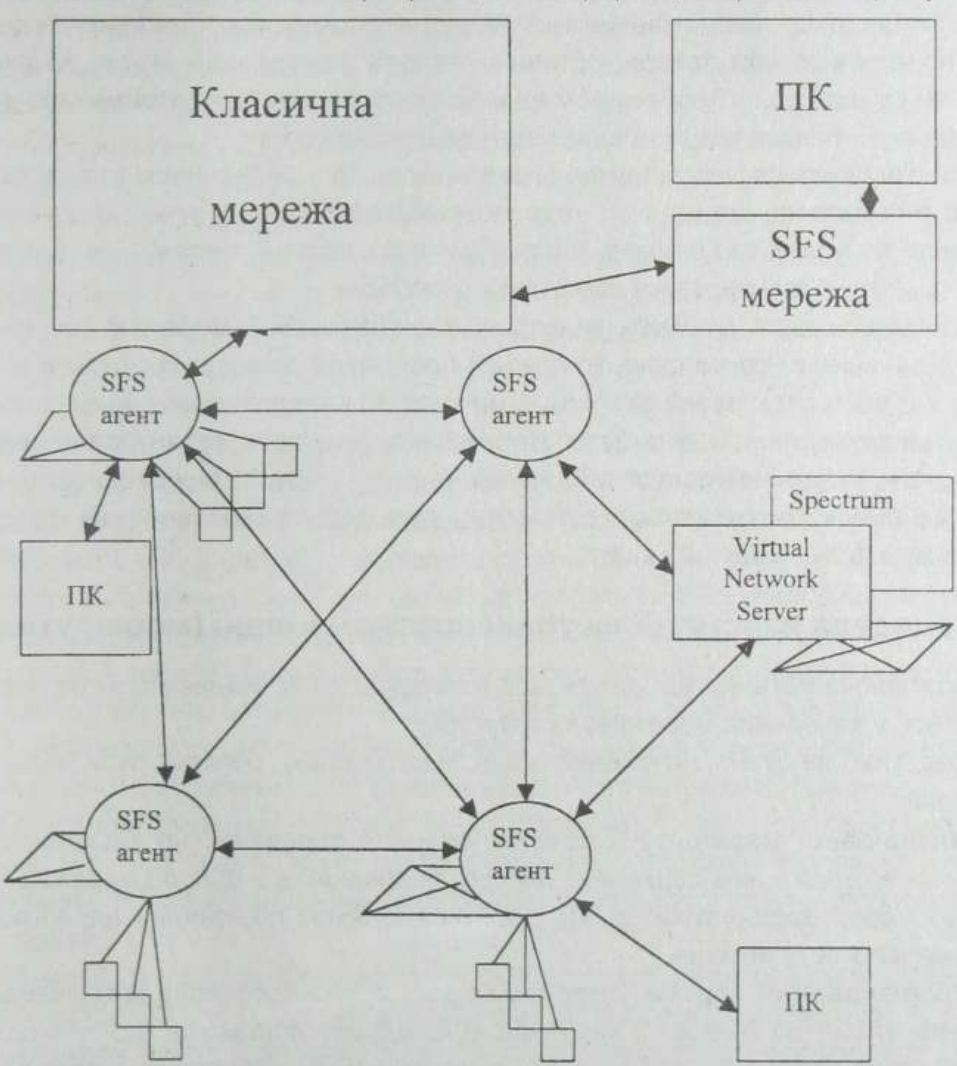


Рис. 9.1. Структура SFS-мережі

Процедура взаємодії на MAC-рівні

Отримавши від джерела перший пакет, SFS-агент на комутаторі визначає, чи підключений приймач до порту цього ж комутатора, і посилає запит до Virtual Network Server про те, чи не заборонено з'єднання джерела з приймачем. В останньому випадку джерелу повідомляється про відмову в передачі.

Якщо приймач підключено до цього комутатора, то комутатор встановлює віртуальне з'єднання між портами джерела і приймача, і наступні пакети пройдуть по цьому з'єднанню без затримки. Як і ATM, по закінченню передачі віртуальне з'єднання існує ще деякий час, протягом якого про нього зберігається запис в таблиці з'єд-

нань комутатора, і якщо виникне потреба зв'язку між цими двома хост-машинами, то запит до Virtual Network Server робити непотрібно. По закінченні цього часу процедура віртуального з'єднання повторюється.

Virtual Network Server встановлює всі віртуальні з'єднання в межах SFS-мережі, програмуючи комутатори на шляху від джерела до приймача, після чого передається повідомлення (типу "з'єднання встановлено") комутатору, до якого підключене джерело. Отримавши це повідомлення та MAC-адресу комутатора, напряму з'єднаного з портом початкового комутатора, останній створює віртуальний канал, по якому передаються всі інші дані. Таким чином весь потік проходить без затримки від джерела до приймача по раніше встановленим віртуальним каналам.

Коли з'ясовується, що приймач знаходиться зовні SFS-мережі (підключений до класичного пристрою, що не підтримує технологію SFS), віртуальне з'єднання встановлюється до порта комутатора, що виходить в класичну мережу, де доставка пакетів до приймача здійснюється класичним способом.

Якщо джерело та приймач знаходяться в різних SFS-мережах, що не мають прямого з'єднання одна з іншою, то описані процедури реалізуються лише в цих мережах, а в класичному "прошарку" між ними трафік проходить звичайним чином.

При цьому одним із атрибутів віртуальних каналів є гарантована пропускна спроможність, яка визначається або на підставі адміністративних постанов, або рахується динамічно, в залежності від потреб джерела та можливостей комутаторів, що підтримують віртуальний канал.

Процедура взаємодії на транспортному рівні (маршрутизація)

Розглядаючи питання маршрутизації в мережах SFS, згадаємо, як ця процедура реалізується в класичних мережевих структурах.

Перед тим, як стане можливим обмін інформацією, повинні бути виконані наступні кроки:

- **Визначення маршруту**. Спочатку джерело перевіряє, чи знаходиться приймач в одній з ним підмережі, або розташований в інших підмережах, доступних через маршрутизатор. Це робиться шляхом порівняння адрес підмереж джерела та приймача.
- **Розпізнавання адреси**. Джерело за допомогою протоколу розпізнавання адреси (ARP) по IP-адресі визначає MAC-адресу приймача, що знаходиться в тій самій підмережі, або MAC-адресу порта маршрутизатора, через який проходить найбільш короткий шлях до підмережі приймача. Всі станції, що знаходяться в тій самій області, отримують запити, і та з них, чия IP-адреса відповідає цьому запиту, посилає джерелу відповідь, що містить IP- і MAC-адресу. Потім джерело направляє пакети по отриманій MAC-адресі, і якщо це MAC-адреса порта маршрутизатора, то подальшим переміщенням пакету до місця призначення займається маршрутизатор за допомогою таблиці маршрутизації, яка до цього моменту вже повинна бути налаштована.

Для налаштування своїх таблиць маршрутизатор використовує відповідні протоколи, такі як RIP (Routing Information Protocol) і OSPF (Open Shortest Path First). Для отримання інформації про топологію мережі всі маршрутизуючі протоколи досить інтенсивно використовують повідомлення, які обробляються всіма станціями

мережі. Часте інформування та передавання таких повідомлень створює додатковий трафік, що практично не контролюється і досить помітно знижує корисну пропускну спроможність мережі.

В SFS-мережах зовсім немає неконтрольованого трафіку. Параметр "default gateway" ("шлюз по замовчуванню") на кожній хост-машині повинен бути налаштований на її власну IP-адресу. Таким чином, всі IP-призначення доступні напряму і знаходяться в тій самій підмережі, що і сама хост-машина.

Коли хост-машина посилає ARP-запит, його приймає SFS-комутатор і адресує лише до Virtual Network Server. Virtual Network Server, користуючись вмістом своєї бази даних, визначає MAC-адресу приймача, після чого реалізується процес встановлення віртуального з'єднання між джерелом і приймачем.

Сигнал типу "з'єднання встановлено", що повертається комутатору, містить MAC-адресу приймача, яка відповідає наданій IP-адресі. Початковий комутатор, немов за дорученням приймача, передає джерелу ARP-відповідь, що містить запитану джерелом MAC-адресу приймача. Джерело оголошує таблицю відповідності IP-MAC-адрес, а початковий комутатор тепер, немов за дорученням джерела, передає приймачу по встановленому віртуальному з'єднанню ARP-запит.

Приймач визначає, що джерело запитує його MAC-адресу, і поновлює свою таблицю відповідності IP-MAC-адрес, розміщуючи туди запис про відповідність IP-MAC-адрес джерела, інформація про що міститься в ARP-запиті. Потім він реалізує процес визначення маршруту, знаходить співпадання параметру "default gateway", адресу порта приймача і вирішує, що може направити ARP-відповідь на запитувану станцію напряму, після чого посилає її по тому ж віртуальному з'єднанню.

В процесі проходження цієї відповіді кожний комутатор, через який проходить віртуальне з'єднання, знаходить відповідний запис в таблиці з'єднань і направляє пакет в потрібний порт. Після цього джерело та приймач можуть обмінюватись інформацією вільно, без затримки, по встановленому віртуальному з'єднанню.

При відмові фізичної лінії або комутатора, через який проходить віртуальне з'єднання, Virtual Network Server автоматично перепрограмує відповідні комутатори та встановить новий віртуальний канал. Цей процес абсолютно непомітний для кінцевих станцій.

SFS-мережа підтримує режим "multicast" (передача повідомлень одночасно від однієї станції до кількох), котрий в класичних структурах реалізується при застосуванні протоколу IGMP. Для цього Virtual Network Server веде відповідні списки учасників multicast-груп. Коли який-небудь учасник multicast-групи передає перший пакет по відповідній multicast-адресі, із всіма станціями, що входять у відповідну multicast-групу, встановлюються віртуальні з'єднання. Як і у випадку unicast-з'єднань (типу "точка-точка"), встановлені multicast віртуальні з'єднання існують деякий заданий адміністратором час після останнього їх використання та руйнуються після його спливу.

9.7.3. Підтримка віртуальних мереж

Віртуальна мережа складається з віртуальних сегментів, належність до яких визначається не точкою фізичного підключення, а деякою логічною умовою. Віртуальні сегменти реально існують тоді, коли є механізм розділення трафіку між ними, тобто

локалізація трафіку сегмента. В той же час, якщо необхідно, повинен бути організований обмін даними між віртуальними сегментами — для об'єднання фізичних сегментів в таких випадках використовують фізичні мости та маршрутизатори.

Розподіл трафіку між віртуальними сегментами може бути виконано різними засобами в залежності від того, інформація якого рівня моделі OSI для цього використовується. При цьому віртуальні мережі будуть мати індекс цього рівня.

➢ Модель OSI розглядається в пункті 1.3 першого розділу книги.

Рівень 1 моделі OSI

Приналежність до віртуальної мережі рівня 1 визначається тим, до якого порту концентратора або комутатора підключений користувач. Об'єднання груп портів в межах одного концентратора або комутатора дозволяє організувати кілька сегментів, що мають свій внутрішній трафік. Більшість виробників, говорячи про те, що їх обладнання дозволяє створювати віртуальні мережі, мають на увазі саме таку організацію. Однак територіальне розміщення клієнтів таких "віртуальних" мереж залежить від розташування комутатора, до порту якого вони підключені, отже, ці сегменти фактично не є віртуальними. Тут більш правильно було б говорити про мікросегментацію.

Рівень 2 моделі OSI

Ознакою приналежності до віртуальної мережі рівня 2 слугує MAC-адреса клієнта, унікальна для кожного мережевого адаптера. Де б не був підключений клієнт таким чином організованої віртуальної мережі, він завжди автоматично буде асоційований лише з тою віртуальною LAN, при конфігуруванні якої була вказана його MAC-адреса.

Мережі 2 рівня мають високий рівень секретності (низьку вірогідність несанкціонованого доступу). Поява невідомих MAC-адрес (не включених в список клієнтів мережі) одразу ж виявляється. Технологія, орієнтована на попереднє встановлення з'єднань, вже сама по собі зводить до мінімуму можливість "прослуховування" інформації, що передається. У відповідальних випадках ця властивість може бути ще більше жорсткою, якщо заборонити встановлення віртуальних з'єднань, а використовувати постійні з'єднання, сконфігуровані адміністратором вручну.

Рівень 3 моделі OSI

Віртуальна мережа рівня 3 основана на використанні структури адреси (IP, IPX) мережевого рівня моделі OSI. Адреса мережевого рівня визначається адміністратором мережі та містить дві або кілька секцій, котрі в сукупності являють собою повну адресу клієнта, але, якщо їх розглянути окремо, також мають досить визначений зміст (наприклад, вони можуть бути адресою області, до якої належить клієнт). Класична модель побудови мереж потребує абсолютного визначення правил, згідно яким адреса 3-го рівня повинна відповідати логічній та фізичній структурі мережі.

Підтримка віртуальних мереж в SFS

Технологія комутації SecureFast та система мережевого менеджменту Spectrum складають основу SecureFast Virtual Networking (VNet) — механізму підтримки віртуальних мереж.

Організувати VNet рівня 2 можна або засобами локального менеджменту на Virtual Network Server, або за допомогою Spectrum. Створюється перелік MAC-адрес станцій — клієнтів VNet, який зберігається на Virtual Network Server. Члени VNet можуть бути розташовувані в у будь-якому місці мережі і довільно по ній пересуватись.

Належність до VNet рівня 3 дозволяє адреса протоколу транспортного рівня, що назначається при конфігуруванні станції. Станції в SFS-мережі розташовуються також довільно.

За допомогою адміністративних засобів Policy Management можуть бути означені права і можливості окремих хост-машин і VNet в їх взаємодії: загальний дозвіл на з'єднання, пріоритет клієнтів у відношенні доступу до того чи іншого мережевого ресурсу і т. ін.

9.7.4. Автоматизоване управління

Гомогенізація фізичної структури мережі переносить задачу організації її логічної структури, що реалізує функціональне призначення мережі, із області фізичної в область інтелектуальну. Функціонування мережі все менше залежить від того, як виконані фізичні з'єднання, і все в більшому ступені визначається програмними засобами налагодження та управління. Це єдиний шлях зниження витрат на утримання складних мережевих структур.

Попередні системи мережевого менеджменту не були активним елементом управління мережею в реальному часі, а використовувались адміністратором для контролю та дистанційного конфігурування пристроїв. Таким чином ефективність функціонування мережі попадала в пряму залежність від рівня кваліфікації та фізичних можливостей оператора.

Система управління мережами, що орієнтована на встановлення віртуальних з'єднань, повинна реалізовувати автоматичну реконфігурацію мережі на основі аналізу подій, що відбуваються, та з урахуванням виконання стратегічних умов, що визначаються зовні адміністратором мережі. Інакше кажучи, вона повинна бути автоматизованою системою управління в повному розумінні цього слова.

Такою є, наприклад, розподілена експертна система Spectrum, яка забезпечує повнофункціональне управління локальними та глобальними комунікаційними системами, що складаються з обладнання різних виробників та використовують різні мережеві технології. За допомогою технології індуктивного моделювання (Inductive Modeling Technology) створюється адаптивна модель об'єкта управління — мережі. Математичний апарат Spectrum дозволяє не тільки ініціювати рішення проблем, але й самостійно розв'язувати їх.

Spectrum є відкритою системою, що має великі можливості для власного розвитку і надає засоби розробки різних рівнів, навіть до найнижчого рівня, коли можна на мові C++ написати свою власну специфічну програму, яка стане невід'ємною частиною Spectrum.

Частина IV

Захист мереж

Розділ 10

Засоби захисту у мережах

"Маленький хлопчик дискету знайшов,
З нею він швидко до батька прийшов,
Всунув в комп'ютер, натиснув Reset...
Більше програм на вінчестері нєт!"
(З дитячого фольклору)

10.1. Загальні положення

Захист даних від несанкціонованого доступу до всієї інформації визначають наступні основні чинники:

- необхідність гарантії відносно необережної роботи окремих користувачів з файловою структурою серверу, яка може призвести до втрат інформації;
- вимога конфіденційності даних, їх призначення та напрямків обміну ними;
- необхідність захисту інформації;
- захист від умисного руйнування інформаційного середовища мережі:
- необхідність постійного виконання загальних вимог — до захисту інформаційного простору від зараження вірусами, недоцільного накопичення зайвих та шкідливих даних.

Адміністратор вживає наступних заходів:

- встановлює системи паролів входу в мережу, періодично їх змінює, наглядає за дотриманням всіма користувачами правил збереження таємниці;
- визначає пріоритети доступу користувачів до конкретних каталогів файлових структур сервера, встановлює право відносно використання файлів в них (наприклад, встановлення атрибуту "тільки для читання" захищає інформацію від її модифікації та знищення);
- встановлює порядок створення резервних копій на визначених носіях та наглядає за якісним їх збереженням, визначає наявний вільний дисковий простір цих носіїв, періодичну ревізію та знищення зайвої інформації на них;

- організує "дзеркальне" відображення всієї інформації на спеціальній системі носіїв (надмірно дублює всю роботу мережі) з наступним визначенням необхідності збереження її в резервних копіях;
- організує захист від випадкових чи спрямованих електронних перешкод роботі мережі та системі безперебійного живлення тощо.

Крім перелічених заходів адміністрування, додатково організується захищений порядок використання принтерів. Розподілений доступ до принтерів може здійснюватись за рахунок використання окремого сервера друку (спулера) або перемикача принтера (рис. 10.1). Адміністратор при цьому встановлює пріоритети та обмеження з доступу до принтера відповідно до прав та обов'язків користувачів мережі.

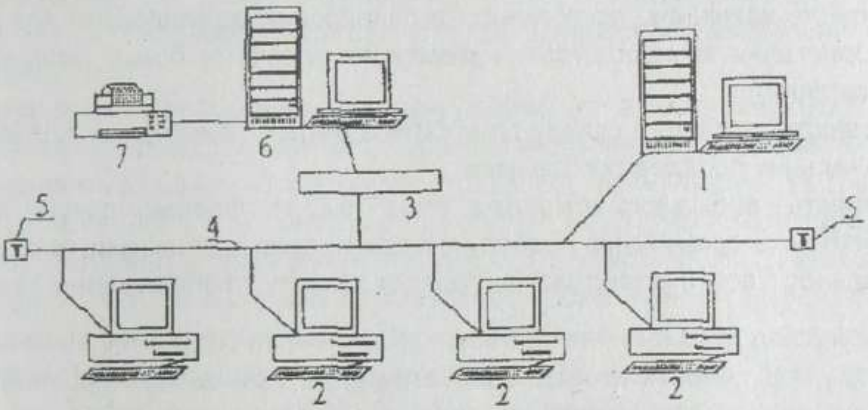


Рис. 10.1. Топологія мережі з використанням спулера:
1 — сервер; 2 — робоча станція; 3 — концентратор; 4 — канали передачі даних;
5 — термінатор; 6 — сервер друку (спулер); 7 — принтер

Робота спулера зводиться до виконання двох, основних функцій:

- створення тимчасового файлу за заявкою на друк інформації;
- передача створеного файлу в буфер принтера для його друку та наступне знищення цього файлу в пам'яті спулера.

Таким чином, черга на друк організується в спулері за часом створення тимчасових файлів, які надходять для друку. При цьому зберігається можливість відмови від друку користувачем до його початку та під час його виконання.

Аналогічно здійснюється захищене використання модемів для роботи окремих РС мережі із зовнішнім середовищем або через мости. Доступ до них, як і до принтера, встановлює адміністратор. Для роботи електронної пошти адміністратор заносить до серверу адреси та імена користувачів мережі, необхідні атрибути адресного звернення до користувачів зовнішнього середовища та можливі маршрути каналів передачі даних.

До обов'язків системного адміністратора ("сисадміна") належать також визначення доцільного ступеню використання ресурсів мережі, її продуктивності, визначення елементів та чинників, які стримують швидкодію мережі в цілому, та впровадження комплексних заходів щодо подолання їх негативного впливу.

У випадку призначення адміністратором групи фахівців, обов'язки кожного чітко регламентуються і визначаються їх права.

10.1.1. Пропозиції з організації роботи в LAN

Для впровадження локальних мереж в роботу будь-якого об'єкту практичні рекомендації можна звести до пропозіцій відносно планування та створення локальної мережі об'єкту, організації роботи в ній та нарощування її можливостей.

Для створення мережі в будь-якій структурі керівному складу об'єкту необхідно визначити:

- її функціональне призначення та обсяг задач, які необхідно виконувати на даному рівні управління;
- структуру мережі та ступінь залучення елементів її структури для виконання задач;
- наявність технічних, програмних та фінансових можливостей для створення, використання та нарощування мережі як елементу більш складної системи управління;
- готовність особового складу до роботи в мережі і виконання відповідних функціональних посадових обов'язків;
- наявність необхідного комплексу службових та правових документів, які регламентують організацію і роботу в мережі, права, обов'язки та ступінь відповідальності всіх її користувачів, порядок захисту та збереження даних в ній.

Функціональне призначення системи управління та задачі, які вона виконує, визначають структуру мережі та її складові елементи. При цьому першочергову увагу необхідно звернути на ті елементи системи, робота яких визначає якість виконання задач. Це стосується, наприклад, для військового об'єкту — оперативного відділу, штабу артилерії, апарату начальника розвідки. Крім того, на структуру мережі значно впливає рівень органу управління та його місце в загальній структурі СУВ.

Архітектура мережі може передбачати різні варіанти топології, але з урахуванням подальшої розбудови мережі слід додержуватися наступних рекомендацій.

Мережу необхідно створювати на базі виділених серверів. Одноранговa мережа значно дешевша, але її можливості дуже обмежені, особливо відносно створення потужного банку даних, знань з розподіленим доступом для колективу користувачів. Колективна робота з таким банком в одноранговій мережі дуже проблематична.

Мережа із зіркоподібною топологією має відносно більшу надійність. Вихід з ладу будь-якого променю зіркоподібної топології не призводить до відмови всієї системи. Необхідно передбачити застосування багатопортового концентратора як розподіляючого елемента, що надасть у майбутньому можливість простого переходу від зіркоподібної топології до деревоподібної.

Для головних елементів мережі — серверів — необхідно передбачати застосування набільш потужних комп'ютерів (робочих та графічних станцій). Це зумовлено необхідністю забезпечення достатнього рівня загальних показників ефективності мережі: швидкодії, обсягу оперативної нам'яті та загального обсягу зовнішніх носіїв пам'яті. При організації роботи з віддаленими робочими станціями через модем необхідно передбачити застосування закритих виділених ліній зв'язку.

З точки зору якості передачі даних, кращі показники мають оптоволоконні кабелі, але, враховуючи їх вартість та порівняльні характеристики ліній зв'язку, більш придатними є екрановані витi пари. Вони забезпечують досить високу швидкість пе-

редачі даних, роздільне використання каналів обміну (в різних парах), значну захищеність і надійність передачі інформації.

Досвід експлуатації локальних мереж показує, що для них більш доцільним є використання ОС мережі NetWare різних версій, переваги якої наведені в пункті 2.3 книги. Організація роботи LAN залежить від багатьох чинників, але її ефективність визначається, в першу чергу, рівнем використання загального ресурсу мережі, рівнем кваліфікації адміністратора, користувачів та суворим дотриманням всіма користувачами дисципліни та встановленого порядку роботи в мережі.

В організаційних заходах необхідно передбачити:

- визначення відповідальних осіб за планування, створення та налаштування локальної мережі, термінів виконання етапів, передбачення необхідного матеріального, технічного, програмного та фінансового забезпечення всього комплексу робіт;
- встановлення та затвердження порядку приймання та перевірки на функціональну придатність створеної мережі для виконання визначених задач управління;
- призначення адміністратора (адміністративної групи) мережі та групи технічного забезпечення для організації її роботи та підтримки відповідної функціональної готовності, встановлення порядку та періодичності технічного обслуговування і заходів матеріально-технічного забезпечення поточної роботи;
- організацію робіт з визначення необхідного ступеню захисту інформації, встановлення категорії мережі та рівня обмежень при виконанні робіт в ній;
- розробку перспективного плану розвитку та визначення основних напрямків подальшого технічного, програмного і комунікаційного удосконалення мережі, планування заходів щодо підвищення професійної підготовки користувачів мережі.

Організація управління мережею має базуватись на плані, в якому відображаються відомості про кабельну мережу, схемне рішення її сполучень, її протяжність, стандарти протоколів та обладнання, дані про робочі станції, технології збору, обробки та подання інформації в межах мережі та при обміні із зовнішнім середовищем і т.ін. До плану залучають, крім того, інформацію про засоби та методи контролю і управління мережею.

Управління мережею включає:

- аналіз роботи мережі в цілому та окремих її складових щодо продуктивності, обсягів та швидкості обробки інформації, що циркулює в мережі, кількості та характеру помилок в роботі, які свідчать про ступінь сумісності технічного та програмного рівня елементів мережі;
- своєчасне внесення коректив при залученні в мережу нових елементів та відповідне до цього оновлення програмних засобів;
- постійний контроль за роботою користувачів, періодична зміна порядку доступу до інформаційних ресурсів, аналіз стану та удосконалення системи заходів щодо захисту інформації та запобігання втрати даних;
- систематичний облік помилок роботи мережі, їх аналіз та впровадження заходів щодо запобігання їх появи або зменшення їх впливу на функціональні можливості мережі;
- контроль розподілу та підтримка потрібного рівня використання ресурсів мережі за рахунок нарощування технічних можливостей, видалення зайвої та надмірної інформації.

Розглянутий перелік заходів та задач зі створення та організації роботи мережі на об'єктах та в органах управління різних рівнів потребує від посадових осіб глибоких знань у вказаній предметній сфері та цілеспрямованої роботі з обгрунтування вимог, широкого впровадження та ефективного використання інформаційно-розрахункових систем на основі локальних мереж.

10.2. Заходи забезпечення безпеки обробки інформації

Впровадження LAN в практику роботи будь-якого об'єкта (органів управління), створення в перспективі розгалужених автоматизованих систем управління об'єктом та необхідність забезпечення колективного доступу користувачів до інформаційних ресурсів невпинно веде до значного підвищення ступеню вразливості інформаційних систем і висуває на передній план розв'язання проблеми забезпечення безпеки інформації.

З визначенням на сучасному етапі нової функції конкуруючих об'єктів — ведення інформаційної боротьби — та створення LAN на базі технічних засобів і програмного забезпечення як вітчизняного, так і закордонного виробництва, проблема набуває значної гостроти.

Безпека інформації — це захист її від несанкціонованого доступу, зміни або руйнування. Вона досягається шляхом проведення та виконання в строгій відповідності до законодавчих актів та керівних документів комплексу організаційних, технічних та програмних заходів щодо отримання, обробки, збереження, використання всього програмно-інформаційного простору інформаційно-розрахункової системи (ІРС) об'єктів та контролю їх виконання.

У загальній проблемі забезпечення безпеки інформації можна виділити наступні основні напрямки проведення заходів:

- захист об'єктів ІРС;
- захист процесів і процедур обробки інформації;
- захист каналів передачі даних;
- зменшення електромагнітного випромінювання всіх елементів системи;
- керівництво захистом тощо.

Всі заходи являють собою складну систему захисту, визначаються та проводяться в тісному взаємозв'язку на всіх етапах розробки, створення та функціонування ІРС.

Основою для проведення заходів є класифікація інформації за рівнями її важливості (таємності) та, у відповідності до цього, встановлення ступенів захисту об'єктів і елементів технічного і програмного забезпечення ІРС (файлів, програм, пакетів дисків, АРМ, LAN в цілому).

У відповідності до проведеної класифікації в першу чергу проводиться визначення всієї множини потенціально можливих загроз та каналів доступу до інформації, визначаються вимоги до системи захисту, здійснюється вибір заходів, методів і засобів захисту інформації та їх впровадження і організація використання. В подальшому здійснюється безперервний контроль цілісності та управління системою захисту.

10.2.1. Загроза безпеки та можливі канали витоку інформації

Основними джерелами витоку інформації можуть бути:

- діяльність розвідувальних та спеціальних служб;
- навмисне викрадення, знищення викривлення, підробка, копіювання інформації внаслідок несанкціонованного доступу до носіїв чи інформаційних ресурсів;
- недосконалість технічних каналів;
- помилкове знищення або псування інформації особовим складом органів управління;
- втрата інформації в результаті впливів стихійних лих, перебоїв живлення тощо;
- втрата інформації в результаті дії комп'ютерних вірусів.

Найбільшу небезпеку з точки зору викрадення інформації являються собою побічні електромагнітні випромінювання технічних засобів, які беруть участь в процесі обробки та передачі таємної інформації, а також контрольно-вимірювальна апаратура, елементи заземлення, живлення і т. ін. Рівень цих випромінювань може бути достатнім для приймання сигналів та добування інформації за допомогою спеціальної апаратури.

Для організації захисту інформації від її витоку через випромінювання здійснюється категорування всіх елементів ОТ, окремих примішень та всього об'єкта органу управління в цілому, монтаж обчислювальної мережі у відповідності до спеціальних вимог, обстеження об'єктів та встановлення (за необхідності) спеціальних атестованих засобів захисту.

Категорування елементів ОТ здійснюється в спеціальних лабораторіях з метою встановлення рівня їх електромагнітного випромінювання. Категорування приміщень і всього об'єкту в цілому проводиться після завершення монтажу мережі.

За результатами категорування приймається рішення на проведення організаційно-технічних заходів. До основних з них належать:

- захист об'єктів від проникнення ззовні;
- встановлення засобів сигналізації;
- визначення певної охоронної зони навколо об'єкту (зони безпеки);
- заміна звичайних технічних засобів на захищені;
- проведення часткового або повного екранування об'єкту і елементів електронно-обчислювальної техніки;
- застосування перешкодопридушуючих фільтрів та просторового електромагнітного зашумлення об'єктів;
- гальванічна та електромагнітна розв'язка мереж електроживлення;
- передбачення захисту від оптоелектронних способів перехоплення інформації та ін.

Особлива увага приділяється захисту приймально-передавальної апаратури та ліній передачі даних.

Організація допуску передбачає охорону меж встановленої зони безпеки, самого об'єкту, входів до нього і деяких критичних зон та здійснюється у відповідності до керівних документів та діючих статутів.

Розмежування доступу до інформаційного простору пов'язане з процесом встановлення повноважень та відповідальності за використання, зміну та збереження інформції. Повноваження встановлюються у відповідності до посадового призначення осіб об'єкту (органу управління) шляхом встановлення для кожного АРМ функціонального спеціаліста відповідного ПЗ, яке з точки зору забезпечення безпеки інформації дозволяє вирішувати наступні завдання:

- встановлення контролю за доступом (ідентифікація користувача та підтвердження його достовірності) — здійснюється шляхом контролю деяких фізичних або фізіологічних характеристик користувача (голос, фотографія, відбитки пальців, райдужна оболонка ока і т. ін,), встановлення паролей (кодів, спеціальних функцій, спеціальних жетонів тощо) та, можливо, контролем за доступом до інформаційних ресурсів за часом;
- встановлення повноважень санкціонованого доступу до інформації — передбачає чітке визначення інформаційного обсягу та операцій над інформацією, які можна здійснювати з конкретного АРМ фахівця, з встановленням певних обмежень діапазонів додаваємих змін (або часових) чи без них;
- виявлення та оповіщення про порушення і спроби порушень та обмеження корисності несанкціоновано здобутої інформації.

При цьому кожна посадова особа несе повну відповідальність за обсяг інформації, в якому їй надане право внесення змін. У повному обсязі доступ та право внесення змін має тільки адміністратор LAN, який несе повну відповідальність за все програмне забезпечення ІРС.

Одночасно з розмежуванням доступу до інформаційних ресурсів проводяться заходи згідно з можливими видами порушень: розкриттям інформації, несанкціонованою зміною її змісту та руйнуванням.

Розкриття інформації означає, що її зміст став відомий особам, для яких вона не призначалась (підслуховування, підглядання екранів терміналів, перехоплення при обміні інформацією, крадіжка з архівів і т. ін.). В основному вищенаведеними заходами передбачений захист від такого виду порушень. Захист від перехоплення при обміні інформацією здійснюється застосуванням пристроїв криптографічного перетворення та кодування даних під час їх передачі в канали зв'язку.

Несанкціонована зміна змісту інформації може здійснюватися помилково або з певною метою. Основними заходами від внесення помилкових змін є встановлення допустимих діапазонів та контроль вносимих змін, дублювання інформації, застосування програмних механізмів відновлення початкової інформації при внесенні помилкових змін.

Руйнування інформації — це її втрата при неправильному використанні, проведенні цілеспрямованих акцій, що роблять інформацію недоступною або неправильною, а також при пошкодженні носіїв (накопичувачів), в тому числі при виникненні стихійних лих. Особлива увага тут приділяється захисту операційної системи мережі та бази даних.

Забезпечення захисту інформації від її руйнування досягається, перш за все, за рахунок точного виконання всіми посадовими особами ІРС об'єкту вимог керівних документів відносно організації роботи на ОТ та використання програмно-інформаційного середовища, жорсткої системи доступу до найбільш важливих елементів програмного забезпечення LAN, розробка та впровадження операційних систем з програмними засобами захисту та ізоляції порушених ділянок, збереження інформації (дублювання, копіювання — апаратними та програмними засобами), захисту програмних продуктів за встановленими атрибутами.

Окремим питанням, яке потребує постійної уваги, нарощування та використання програмних засобів, є боротьба з комп'ютерними вірусами в програмному середовищі ІРС. Питання потребує окремого розгляду та пильної уваги.

Для забезпечення безперебійної роботи та виключення можливостей втрати інформації здійснюється резервування "гарячих" джерел живлення ОТ та вживаються загальновстановлені заходи щодо захисту ІРС від стихійни лих.

10.3. Засоби захисту у мережах

Проблеми інформаційної безпеки постійно посилюється процесами проникнення практично у всі сфери діяльності суспільства технічних засобів обробки і передачі даних, і передусім — обчислювальних систем. Це дає підставу підняти проблему комп'ютерного права, одним з основних аспектів якої є так звані комп'ютерні посягання. Про актуальність проблеми свідчить простий перелік можливих засобів комп'ютерних злочинів.

Об'єктами посягань можуть бути як самі технічні засоби (комп'ютери і периферія), матеріальні об'єкти, так і програмне забезпечення і бази даних, для яких технічні є оточенням.

В цьому сенсі комп'ютер може виступати і як предмет посягань, і як інструментарій. Якщо розділяти два останніх поняття, то термін комп'ютерний злочин як юридична категорія не має особливого сенсу. Якщо комп'ютер — лише об'єкт посягання, то кваліфікація правопорушення може бути здійснена по існуючим нормам права. Якщо ж — лише інструмент, то достатня тільки така ознака, як застосування технічних засобів.

Можливо об'єднання означених понять, коли комп'ютер водночас і інструмент і предмет. Частково до цієї ситуації відноситься факт викрадення машинної інформації. Якщо викрадення інформації пов'язане з втратою матеріальних і фінансових цінностей, то цей факт можна кваліфікувати як злочин. Також якщо з даним фактом зв'язуються порушення інтересів національної безпеки, авторства, то кримінальна відповідальність прямо передбачена у відповідності із законами України.

Кожне псування роботи комп'ютерної мережі — це не тільки "моральна" шкода для робітників об'єкту і мережевих адміністраторів. З розвитком технологій електронних платежів, "без паперового" документообороту і інше, серйозне псування LAN може просто паралізувати роботу цілих корпорацій і банків, що призводить до відчутних матеріальних втрат. Не випадково, що захист даних в комп'ютерних мережах стає однією з самих гострих проблем в сучасній інформатиці.

На сьогоднішній день сформульовано три базових принципи інформаційної безпеки, що повинна забезпечувати:

- цілісність даних — захист від збоїв, що ведуть до втрати інформації, а також неавторизованого створення або знищення даних;
- конфіденційність інформації;
- доступність для всіх авторизованих користувачів.

Слід також відзначити, що окремі сфери діяльності (банківські і фінансові інститути, інформаційні мережі спецпризначення, системи державного управління, оборонні і спеціальні структури) вимагають спеціальних заходів безпеки даних і висувають підвищені вимоги до надійності функціонування інформаційних систем, в відповідності з характером і важливістю задач, що вирішуються.

10.3.1. Комп'ютерна злочинність

Комп'ютерні злочини умовно можна розділити на дві великі категорії: злочини, пов'язані із втручанням в роботу комп'ютерів, і, злочини, що використовують комп'ютери як необхідні технічні засоби. Перерахуємо основні види злочинів, зв'язаних із втручанням в роботу комп'ютерів.

Несанкціонований доступ до комп'ютерної інформації

Несанкціонований доступ здійснюється, як правило, з використанням чужого імені, зміною фізичних адрес технічних приладів, використанням інформації після рішення задач, модифікацією програмного і інформаційного забезпечення, розкраденням інформації з підключенням апаратури запису до каналів передачі даних.

Хакери, "електронні корсари", "комп'ютерні пірати" — так називають людей, які здійснюють несанкціонований доступ в чужі інформаційні мережі задля забави або користі. Набираючи один номер за іншим, вони терпляче чекають, доки на іншому кінці дроту не відгукнеться чужий комп'ютер. Після цього телефон підключається до приймача сигналів у власній ЕОМ, і зв'язок встановлений. Якщо тепер вгадати код (а слова, що служать паролем часто тривіальні), то можна використовувати чужу комп'ютерну систему.

Несанкціонований доступ до файлів законного користувача здійснюється також через знаходження слабких місць в захисті системи. Виявивши їх, порушник може досліджувати інформацію в системі, копіювати її, звертатися до неї багато раз тощо.

Програмісти інколи допускають помилки в програмах, що не вдається виявити в процесі відлагодження. Автори більш складних програм можуть не помітити деяких хибностей логіки. Вразливі місця інколи виявляються і в електронних мережах. Всі ці помилки дають підстави для комп'ютерної злочинності.

Буває, що дехто використовує комп'ютерну систему, видаючи себе за законного користувача. Системи, що не володіють засобами автентичної ідентифікації (наприклад по фізіологічним характеристикам: по відбиткам пальців, по малюнку райдужної оболонки ока, голосу тощо), виявляються без захисту проти цього прийому. Самий найпростіший шлях для комп'ютерної злочинності — це отримати коди і інші шифри, що ідентифікують законних користувачів.

Це можна зробити:

- придбанням (зазвичай підкупом персоналу) списку користувачів зі всієї необхідною інформацією;
- викраденням такого документу в організаціях, де не налагоджений достатній контроль за їхнім зберіганням;
- підслуховуванням через телефоні лінії тощо.

Це трапляється, наприклад, як з помилковими телефонними дзвінками, коли користувач з терміналу підключається до чиєїсь системи, абсолютно впевнений, що він працює з тієї системою, що і думав. Власник системи, до якої відбулося фактичне підключення, формуючи конкретні відгуки, може підтримувати цю оману впродовж певного часу і таким чином передати деяку інформацію злочинцю.

В будь-якому комп'ютерному центрі є особлива спеціальна програма, що застосовується як системний інструмент у випадку виникнення збоїв або інших відхилень в роботі ЕОМ — своєрідний аналог пристроїв, розміщених в транспорті під надписом "розбити скло в випадку аварії". Така програма — потужний і небезпечний інструмент в руках зловмисника. Несанкціонований доступ може здійснюватись як результат системного збою.

Використання "логічних бомб"

Інколи можливе введення в програмне забезпечення "логічних бомб", що спрацьовують при виконанні певних умов і частково або повністю виводять з ладу комп'ютерну систему. Розглянемо деяки приклади таких засобів.

"Тимчасова бомба" — різновид "логічної бомби", що спрацьовує по досягненні певного моменту часу.

Засіб "троянський кінь" — це введення в чужу програму таких команд, що дозволяють здійснювати нові, не заплановані власником програми функції, але водночас зберігати і колишню дієздатність. За допомогою "троянского коня" злочинці, наприклад, відраховують на свій рахунок певну суму з кожної операції.

Комп'ютерні програмні тексти зазвича надзвичайно складні. Вони складаються з сотень тисяч, а інколи і мільйонів команд. Тому "троянский кінь" з декількох десятків команд навряд чи може бути виявлений, якщо немає підозр відносно цього. Але і в останньому випадку експертам-програмістам необхідно буде багато днів і тижнів, щоб знайти його.

Є ще один різновид "троянського коня". Його особливість полягає в тому, що в будь-яку частину програми вставляються не вірусні команди, а команди, що формують ці вірусні команди. В цьому випадку програмісту, що намагається знайти "троянського коня", необхідно шукати не його самого, а команди, що його формують

В США отримала розповсюдження форма комп'ютерного вандалізму, при якій "троянський кінь" руйнує через деякий проміжок часу всі програми, що зберігаються в пам'яті машини.

Розробка і розповсюдження комп'ютерних вірусів.

"Троянські коні" типу "зруйнуй всі дані цієї програми, перейди в наступну і зроби теж саме" володіють властивостями переходити через комунікаційні мережі з однієї системи в іншу, розповсюджуючись як вірусне захворювання. Виявляється вірус не

відразу: перший час комп'ютер "виношує інфекцію", оскільки для маскування вірус нерідко використовується в комбінації з "логічною бомбою" або "тимчасовою бомбою". Вірус "спостерігає" за всією інформацією, що обробляється, і може переміщатись, використовуючи процеси пересилки цієї інформації.

Починаючи діяти (перехоплювати управління), вірус дає команду комп'ютеру, щоб той записав заражену версію програми. Після цього він повертає програмі управління. Користувач нічого не помітить, бо його комп'ютер знаходиться в стані "здорового носію вірусу". Виявити цей вірус можна, лише володіючи надзвичайно розвиненою програмницькою інтуїцією, оскільки жодних порушень в роботі ЕОМ в даний момент не виявляється.

За оцінкою фахівців, в обігу знаходиться понад 100 типів вірусів, але всіх їх можна поділити на два різновиди: "вульгарний вірус" і "роздроблений вірус". Програма "вульгарного вірусу" написана єдиним блоком, і при виникненні підозр в зараженні ЕОМ експерти можуть виявити її в самому початку епідемії (розмноження). Ця операція вимагає, однак, вкрай ретельного аналізу всіх компонентів ОС.

Програма "роздробленого вірусу" поділена на частини. Ці частини містять інструкції, що вказують комп'ютеру, як зібрати їх щоб відтворити і, отже, розмножити вірус. Таким чином, він майже весь час знаходиться в "розподіленому" стані і лише на короткий час своєї роботи збирається в єдине ціле. Як правило автори вірусу вказують йому число репродукцій, після досягнення якого він стає агресивним.

Віруси можуть бути впроваджені в операційну систему, прикладну програму або в мережевий драйвер. Варіанти вірусів залежать від мети, що переслідуються їх автором. Ознаки їх можуть бути відносно доброякісними, наприклад, уповільнення в виконанні програм або поява "миготящої" крапки на екрані дисплею (т. зв. "італійський стрибунець"). Ознаки можуть бути еволюційними, і "хвороба" буде загострюватись з часом. Так, по незрозумілим причинам, програми починають переповнювати магнітні диски, в результаті чого істотно збільшується обсяг програмних файлів. Нарешті, ці прояви можуть бути катастрофічними і призвести до знищення файлів і програмного забезпечення.

Які засоби розповсюдження комп'ютерного вірусу? Вони ґрунтуються на спроможності вірусу використати будь-який носій даних, що передаються в якості "засобів пересування".

Зручними для розповсюдження "епідемій" є телекомунікаційні мережі. Достатньо одного контакту, щоб персональний комп'ютер був заражений або заразив той, з яким контактував. Однак самий частий засіб зараження — це копіювання програм, що є звичайною практикою у користувачів персональних ЕОМ.

Захисні програми поділяються на три види:

- фільтруючи — що перешкоджають проникненню вірусу;
- протиінфекційні — постійні процеси, що контролюють систему;
- противірусні — налаштовані на виявлення окремих вірусів.

Однак розвиток цих програм поки не встигає за розвитком комп'ютерної "епідемії".

Помітимо, що побажання обмежити використання неперевіреного програмного забезпечення скоріше усього так і залишиться практично не виконаним. Це пов'язано з тим, що фірмові програми на "стерильних" ліцензійних носіях коштують великих грошей. Тому уникнути їхнього неконтрольованого копіювання майже неможливо.

Але слід зазначити, що розповсюдження комп'ютерних вірусів має і деякі позитивні сторони. Так вони є кращим захистом від крадіїв програмного забезпечення. Інколи розробники свідомо заражують свої дискети яким-небудь безпечним вірусом, що добре виявляється будь-яким антивірусом-тестом. Це слугує деякою гарантією проти копіювання такої дискети.

Підробка комп'ютерної інформації

Цей вигляд комп'ютерної злочинності є різновидом несанкціонованого доступу з тієї різницею, що використовує її як правило, не сторонній користувач, а сам розробник. Ідея злочину полягає у підробці вихідної інформації комп'ютерів з метою імітації дієздатності великих систем.

До підробки інформації можна віднести підтасовку результатів виборів, голосувань, референдумів тощо.

Розкрадення комп'ютерної інформації

Якщо "звичайні" розкрадення підпадають під чинність існуючого кримінального закону, то проблема розкрадення інформації значно складніша. Привласнення машинної інформації, в тому числі програмного забезпечення, шляхом несанкціонованого копіювання не кваліфікується як майнове розкрадання. Не дуже далекий від істини жарт, що у нас програмне забезпечення розповсюджується тільки шляхом крадіжок і обміну краденим. При неправомірному звертанні до комп'ютера машинна інформація може не вилучатися з фондів, а копіюватись. Отже, як вже відзначалось вище, машинна інформація повинна бути виділена як самостійний предмет кримінально-правової охорони.

Розглянемо тепер другу категорію злочинів, в якій комп'ютер є "засобом" досягнення мети. Тут можна виділити розробку складних математичних моделей, вхідними даними в яких є можливі умови скоєння злочину, а вихідними даними — рекомендації по вибору оптимального варіанту дій злочинця.

Інший вигляд злочинів з використанням комп'ютерів отримав назву "повітряний змій". В найпростішому випадку робиться спроба відкрити в двох банках по невеликому рахунку. Далі гроші переводяться з одного банку в інший і зворотно з сумами, що поступово підвищуються. Ідея полягає в тому, щоб до того, як в банку з'ясується, що доручення про переведення не забезпечене необхідною сумою, приходило б сповіщення про переведення в цей банк, так щоб загальна сума покривала вимоги про перше переведення.

Цей цикл повторюється велике число раз ("повітряний змій" піднімається все вище і вище) до тих пір, доки на рахунку не виявляється достатня сума (фактично вона постійно "перестрибує" з одного рахунку на інший, збільшуючи свої розміри). Тоді гроші швидко знімаються, а власник рахунку зникає. Цей засіб вимагає дуже точного розрахунку, але для двох банків його можна зробити і без комп'ютера. На практиці в таку гру включають велику кількість банків: так сума нагромаджується швидше і число доручень про переведення не сягає підозрілої частоти. Але управляти цим процесом можна лише з допомогою комп'ютера.

Можна уявити собі створення спеціалізованого комп'ютера-шпигуна, що будучи підключеним до будь-якої мережі, генерує всілякі запитання, фіксує і аналізує отримані відповіді. Поставити перепону перед таким хакером практично неможливо.

10.3.2. Попередження комп'ютерних злочинів

При розробці комп'ютерних систем, вихід з ладу або помилки в роботі яких можуть призвести до важких наслідків, питання комп'ютерної безпеки стають першочерговими. Відомо багато заходів, направлених на попередження злочину. Виділимо з них технічні, організаційні і правові.

До технічних заходів можна віднести захист від несанкціонованого доступу до системи, резервування особливо важливих комп'ютерних підсистем, організацію обчислювальних мереж з можливістю перерозподілу ресурсів у випадку порушення дієздатності окремих ланок, тощо.

До організаційних заходів віднесемо:

- охорону комп'ютерного центру;
- ретельний підбір персоналу;
- виключення випадків виконання особливо важливих робіт лише однією людиною;
- наявність плану відновлення дієздатності центру після виходу його з ладу;
- організацію обслуговування комп'ютерного центру сторонньою організацією або особами, незацікавленими в приховуванні фактів порушення роботи центру;
- універсальність засобів захисту від всіх користувачів (включаючи вище керівництво);
- покладення відповідальності на осіб, що повинні забезпечити безпеку центру;
- вибір місця розташування центру і т. ін.

До правових заходів слід віднести розробку норм, встановлення відповідальності за комп'ютерні злочини, захист авторських прав програмістів, вдосконалення кримінального і громадянського законодавства, а також судочинства. До правових заходів відносяться також питання громадського контролю за розробниками комп'ютерних систем і прийняття міжнародних договорів про їхні обмеження, якщо вони впливають або можуть вплинути на військові, економічні і соціальні аспекти життя країн, укладання угоди тощо.

10.3.3. Захист даних в комп'ютерних мережах

При розгляді проблем захисту даних в мережі передусім виникає питання про класифікацію збоїв і порушень прав доступу, що можуть призвести до знищення або небажаної модифікації даних. Серед потенційних загроз можна виділити наступні:

- Збої обладнання:
  - збої кабельної системи;
  - перебої електроживлення;
  - збої дискових систем;
  - збої систем архівації даних;
  - збої роботи серверів, робочих станцій, мережевих карт тощо.

- Втрата інформації з причини некоректної роботи ПЗ:
  - втрата або зміна даних при помилках ПЗ;
  - втрати при зараженні системи комп'ютерними вірусами.
- Втрати, пов'язані з несанкціонованим доступом:
  - несанкціоноване копіювання, знищення або підробка інформації;
  - ознайомлення з конфіденційною інформацією, що складає таємницю, сторонніх осіб.
- Втрата інформації, пов'язана з неправильним зберіганням архівних даних.
- Помилки персоналу, що обслуговує мережу:
  - випадкове знищення або зміна даних;
  - некоректне використання програмного і апаратного забезпечення.

В залежності від можливих порушень роботи мережі, можливості щодо захисту інформації об'єднуються в три основні класи:

- **засоби фізичного захисту,** включаючи засоби захисту кабельної системи, систем електроживлення, засобів архівації, дискових масивів тощо;
- **програмні засоби захисту,** в тому числі антивірусні програми, системи розмежування повноважень, програмні засоби контролю доступ і т. ін.;
- **адміністративні міри захисту**, включаючи контроль доступу в приміщення, розробку стратегії безпеки фірми, планів дій в надзвичайних ситуаціях тощо.

Треба відзначити, що подібне ділення достатньо умовно, оскільки сучасні технології розвиваються в направленні поєднання програмних і апаратних засобів захисту. Найбільше розповсюдження такі програмно-апаратні засоби отримали в сфері контролю доступу, захисту від вірусів тощо.

Концентрація інформації в комп'ютерах — подібно до концентрації готівки грошей в банках — змушує все більше посилювати контроль з метою захисту інформації. Юридичні питання, персональна таємниця, національна безпека — всі ці міркування вимагають підсилення внутрішнього контролю в комерційних і урядових організаціях. Роботи в цьому напрямку призвели до появі нової дисципліни: безпека інформації або науки криптології.

Фахівець у галузі безпеки інформації відповідає за розробку, реалізацію і експлуатацію системи забезпечення інформаційної безпеки, направленої на підтримку цілісності, придатності і конфіденційності накопиченої в організації інформації. В його функції входить забезпечення фізичної (технічні засоби, лінії зв'язку і віддалені комп'ютери) і логічної (дані, прикладні програми, операційна система) системи захисту ресурсів мережі.

Складність створення системи захисту інформації визначається тим, що дані можуть бути викрадені з комп'ютера і водночас залишатись на місці.

Забезпечення безпеки інформації — дорога справа, і не стільки із-за витрат на закупівлю або установку засобів, скільки по тій причині, що важко кваліфіковано визначити межі розумної безпеки і відповідного підтримання системи в дієздатному стані.

Якщо локальна мережа розроблялась з метою спільного використання ліцензійних програмних засобів, коштовних кольорових універсальних або фото прин-

терів, великих файлів загальнодоступної інформації, то немає ніякий потреби навіть в мінімальних системах шифрування/дешифрування інформації.

Засоби захисту інформації не можна проектувати, купувати або встановлювати до тих пір, доки не зроблений відповідний аналіз. Аналіз ризику повинен дати об'єктивну оцінку багатьох факторів (прогноз порушення роботи, імовірність появи порушення роботи, комерційні витрати, зниження коефіцієнту готовності системи, суспільні відношення, юридичні проблеми тощо) і надати інформацію для визначення схожих типів і рівнів безпеки.

Комерційні організації часто переносять секретну корпоративну інформацію з великих комп'ютерних систем у середовище відкритих систем і зустрічаються з новими і складними проблемами при реалізації і експлуатації системи безпеки. Сьогодні все більше організацій розгортають потужні розподілені бази даних і прикладні програми типу "клієнт-сервер" для управління комерційними даними. При збільшенні розподілу даних зростає також і ризик неавторизованого доступу до даних і їхнього псування.

Шифрування даних традиційно використовувалося урядовими і оборонними департаментами, але у зв'язку зі зміною потреб і багато фірм і компаній починають використовувати можливості, що надаються шифруванням для забезпечення конфіденційності інформації.

10.3.4. Шифрування

Шифрування даних може здійснюватися в режимах "on-line" і "off-line". Зупинимося на першому типі, що має велике розповсюдження. Найбільш відомі два алгоритми.

Стандарт шифрування даних DES (Data Encryption Standart) був розроблений фірмою IBM на початку 70-х років і в теперішній час є урядовим стандартом для шифрування цифрової інформації. Він рекомендований Асоціацією американських банкірів. Складний алгоритм DES використовує ключ довжиною 56 байтів і 8 бітів перевірки на парність та вимагає від зловмисника перебору 72 квадриліонів можливих ключових комбінацій, забезпечуючи високий ступінь захисту при невеликих видатках. При частій зміні ключів алгоритм задовільно вирішує проблему захисту конфіденційної інформації.

Алгоритм RSA був створений Ривестом, Шамиром і Альдеманом в 1976 році і являє собою значний крок в криптографії. Цей алгоритм також був прийнятий в якості стандарту Національним бюро стандартів.

DES технічно є симетричним алгоритмом, а RSA — асиметричним, тобто він використовує різні ключі при шифруванні і дешифруванні. Користувачі мають два ключі і один з них — відкритий ключ. Відкритий ключ використовується для шифрування повідомлення користувачем, але тільки певний одержувач може дешифрувати його своїм секретним ключем. Відкритий ключ не може бути використаний для дешифрування. Це робить непотрібними секретні угоди про передачу ключів між кореспондентами.

DES обмежує довжину даних і ключа в бітах, а RSA може бути реалізований при будь-якій довжині ключа. Чим довший ключ, тим вище рівень безпеки (але стає довшим і процес шифрування і дешифрування). Якщо ключі DES можна згенерувати за

мікросекунди, то зразковий час генерації ключа RSA — десятки секунд. Тому відкритим ключам RSA віддають перевагу розробники ПЗ, а секретним ключам DES — розробники апаратури.

10.3.5. Фізичний захист даних

Кабельна система

Кабельна система залишається головною "ахілесовою п'ятою" більшості LAN: по даним різноманітних досліджень, саме кабельна система є причиною більш ніж половини всіх відмов та збоїв в мережі. В зв'язку з цим кабельній системі повинна приділятись особлива увага з самого початку проектування мережі.

Найкращим чином позбавити себе проблем з приводу неправильної прокладки кабелю є використання розповсюджених в останній час так званих структурованих кабельних систем що отримали широке розповсюдження. До структурованих кабельних систем відносяться, наприклад, SYSTIMAX SCS фірми AT&T, OPEN DECconnect компанії Digital, кабельна система корпорації IBM.

Поняття "структурованість" означає, що кабельну систему будинку можна поділити на декілька рівнів в залежності від призначення і місця розташування компонентів кабельної системи.

Наприклад, кабельна система SYSTIMAX SCS складається із:

- зовнішньої підсистеми (campus subsystem);
- апаратних засобів (equipment room);
- адміністративної підсистеми (administrative subsystem);
- магістралі (backbone cabling);
- горизонтальної підсистеми (horizontal subsystem);
- робочих місць (work location subsystem).

Зовнішня підсистема складається з мідного оптоволоконного кабелю, приладів електричного захисту і заземлення і зв'язує комунікаційну систему і апаратуру. Крім того, в цю підсистему входять прилади з'єднання зовнішніх кабельних ліній з внутрішніми. Апаратні засоби використовуються для розміщення різноманітного комунікаційного обладнання, призначеного для забезпечення роботи адміністративної підсистеми.

Адміністративна підсистема призначена для швидкого і ефективно управління кабельною системою SYSTIMAX SCS при зміні планів розміщення персоналу і відділів. В її склад входять: кабельна система (неекранована вита пара і оптоволокно), прилади комутації і з'єднання магістралі та горизонтальної підсистеми, з'єднувальні шнури, маркірувальні засоби тощо.

Магістраль складається з мідного кабелю або комбінації мідного і оптоволоконного кабелю і додаткового обладнання. Вона зв'язує між собою поверхи будинку або більші площі одного і того ж поверху.

Горизонтальна система на базі витого мідного кабелю розширює основну магістраль від вхідних вузлів адміністративної системи поверху до розеток на робочому місці.

І, нарешті, обладнання робочих місць включає в себе з'єднувальні шнури, адаптери, з'єднувачі і забезпечує механічне і електричне з'єднання обладнання робочого місця з горизонтальною кабельною підсистемою.

Найкращим засобом захисту кабелю від фізичних (а інколи температурних і хімічних дій, наприклад, в виробничих цехах) є прокладка кабелів з використанням захищених коробів. При прокладці мережевого кабелю поблизу джерел електромагнітного опромінення необхідно виконувати наступні вимоги:

- неекранована вита пара повинна розташовуватись мінімум на відстані 15–30 см від електричного кабелю, розеток, трансформаторів тощо;
- у коаксіального кабелю відстань до електричної лінії або електроприборів повинна бути не менше 10–15 см.

Важлива проблема правильної інсталяції і безвідмовної роботи кабельної системи — відповідність всіх її компонентів вимогам міжнародних стандартів. Найбільше розповсюдження в нинішній час отримали наступні стандарти кабельних систем:

- **специфікації корпорації IBM,** що передбачають дев'ять різноманітних типів кабелів, найбільш розповсюдженим серед них є кабель IBM type 1 — екранована вита пара (STP) для мереж Token Ring;
- **система категорій Underwriters Labs (UL)** створена цією лабораторією спільно з корпорацією Anixter; система включає п'ять рівнів кабелів. В нинішній час система UL доведена в відповідність з системою категорій EIA/TIA.

Стандарт EIA/TIA 568 був розроблений спільними зусиллями UL, ANSI (American National Standarts Institute) і Electronic Industry Association / Telecommunications Industry Association, підгрупою TR41. 8.1 для кабельних систем на витій парі (UTP).

В доповнення до стандарту EIA/TIA 568 існує документ DIS 11801, розроблений організацією ISO (International Standarts Organization) і IEC (International Electrotechnical Commission). Даний стандарт використовує термін "категорія" для окремих кабелів і термін "клас" для кабельних систем.

Необхідно також відзначити, що вимоги стандарту EIA/TIA 568 відносяться лише до мережевого кабелю, але реальні системи, окрім цього, включають також з'єднальні роз'єми, розетки, розподільні панелі і інші елементи. Використання тільки кабелю категорії 5 не гарантує створення кабельної системи цієї категорії. В зв'язку з цим все вище перераховане обладнання повинно бути також сертифіковане на відповідність даної категорії кабельної системи.

Системи електрозабезпечення

Найбільш надійним засобом запобігання втрат інформації при короткочасному відключенні електроенергії в нинішній час є встановлення джерел безперебійного живлення. Різноманітні по своїм технічним і споживчим характеристикам, подібні прилади можуть забезпечити живлення всієї локальної мережі або окремого комп'ютера впродовж проміжку часу, достатнього для відновлення подачі напруги або для збереження інформації на магнітному носії. Більшість джерел безперебійного живлення водночас виконують функції і стабілізатора напруги, що є додатковим захис-

том від стрибків напруги в мережі. Сучасні мережеві прилади — сервери, концентратори, мости тощо — оснащені власними дубльованими системами електроживлення.

За кордоном корпорації мають власні аварійні електрогенератори або резервні лінії електроживлення. Ці лінії підключені до різних підстанцій, і при виході з ладу однієї з них електрозабезпечення здійснюється з резервної підстанції.

Системи архівування і дублювання інформації

Організація надійної і ефективної системи архівації даних є однією з найважливіших задач по забезпеченню захисту інформації в мережі. В невеликих мережах, де встановлені один-два сервери, найчастіше застосовується установка системи архівації безпосередньо у вільні слоти серверів. В крупних корпоративних мережах більш прийнято організовувати виділений спеціалізований архіваційний сервер. Зберігання архівної інформації повинно бути організоване в спеціальному приміщенні. Фахівці рекомендують зберігати дублікати архіву найбільш цінних даних в іншому будинку, на випадок пожежі або стихійної лиха.

Захист від стихійних лих

Основний і найбільш розповсюджений засіб захисту інформації і обладнання від різноманітних стихійних лих — пожеж, землетрусів, повені тощо — це зберігання архівних копій інформації або розміщення деяких мережевих приладів, наприклад, серверів баз даних, в спеціальних захищених приміщеннях, розташованих, як правило, в інших будинках або, навіть в іншому районі міста чи в іншому місті.

10.3.6. Програмні і програмно-апаратні засоби захисту

Захист від комп'ютерних вірусів

По даним дослідження, проведеного фірмою Creative Strategies Research, 64% з 451 опитаного фахівця випробували "на собі" дію вірусів. На сьогоднішній день додатково до тисяч вже відомих вірусів з'являється 100–150 нових штамів щомісячно. Найбільш розповсюдженими засобами захисту від вірусів на сьогоднішній день залишаються різноманітні антивірусні програми.

Однак в якості перспективного підходу до захисту від комп'ютерних вірусів в останні роки все частіше застосовується поєднання програмних і апаратних засобів захисту. Серед апаратних засобів можна відзначити спеціальні антивірусні плати, що вставляються в стандартні слоти розширення комп'ютера. Корпорація Intel ще в 1994 році запропонувала перспективну технологію захисту від вірусів в комп'ютерних мережах. Flash-пам'ять мережевих адаптерів Intel EtherExpress PRO/10 містить антивірусну програму, скануючи всі системи комп'ютера ще до його завантаження.

Захист від несанкціонованого доступу

Проблема захисту інформації від несанкціонованого доступу особливо загострилась з широким розповсюдженням локальних і, особливо, глобальних комп'ютерних мереж. Необхідно також відзначити, що інколи шкоду завдають самі користувачі, що випадково псують або видаляють важливі дані. В зв'язку з цим, окрім контролю доступу, не-

обхідним елементом захисту інформації в комп'ютерних мережах є розмежування повноважень користувачів.

В комп'ютерних мережах при організації контролю доступу і розмежування повноважень користувачів найчастіше використовуються вбудовані засоби мережевих операційних систем. Так, найбільший виробник мережевих ОС — корпорація Novell — у продукті NetWare передбачила окрім стандартних засобів обмеження доступу, таких, як система паролів і розмежування повноважень, ряд нових можливостей, що забезпечують перший клас захисту даних. Нова версія NetWare передбачає можливість кодування даних по принципу "відкритого ключа" (алгоритм RSA) з формуванням електронного підпису для пакетів, що передаються по мережі.

В той же час в такій системі організації захисту все одно залишається слабке місце: рівень доступу і можливість входу в систему визначається паролем. Не секрет, що пароль можна вкрасти або підібрати. Для виключення можливості неавторизованого входу в комп'ютерну мережу в останній час використовується підхід, що комбінує пароль плюс ідентифікація користувача по персональному "ключу". В якості "ключа" може використовуватись пластикова картка (магнітна або з вбудованою мікросхемою — *smart-card*) чи різноманітні прилади для ідентифікації особи по біометричній інформації (по райдужній оболонці ока або відбиткам пальців, розмірам кисті руки і т. ін).

Якщо сервер або мережеві робочі станції, наприклад, облаштовані пристроєм читання смарт-карток і спеціальним ПЗ, то можна значно підвищити ступінь захисту від несанкціонованого доступу. У цьому випадку для доступу до комп'ютера користувач повинен вставити смарт-карту в спецпристрій для читання і ввести свій персональний код. Програмне забезпечення дозволяє встановити декілька рівнів безпеки, що керуються системним адміністратором. Можливий і підхід, що комбінується з введенням додаткового паролю. При цьому приймаються спеціальні міри проти "перехоплення" паролю з клавіатури. Цей підхід значно надійніший за застосування паролів, оскільки, якщо пароль підгледіти, та користувач про це може не знати, якщо ж зникла картка, то можна прийняти міри негайно.

Смарт-карти управління доступом дозволяють реалізувати такі функції, як контроль входу, доступ до приладів персонального комп'ютера, доступ до програм, файлів і команд тощо. Крім того, можливо також здійснення контрольних функцій, таких як, реєстрація спроб порушення доступу до ресурсів, використання заборонених утиліт, програм, команд DOS і т. ін.

Одним з ефективних прикладів створення комплексного рішення для контролю доступу у відкритих системах, заснованого як на програмних, так і на апаратних засобах захисту, стала система Kerberos. В основі цієї схеми авторизації лежать три компоненти:

- **База даних,** що містить інформацію по всім мережевим ресурсам, користувачам, паролям, шифрувальним ключам і т. ін.
- **Авторизаційний сервер**, що обробляє всі запити користувачів на предмет отримання того або іншого типу мережевих послуг. Авторизаційний сервер, одержуючи запитання від користувача, звертається до бази даних і визначає, чи має користувач право на проведення даної операції. Паролі користувачів по мережі не передаються, що також підвищує ступінь захисту інформації.
- **Сервер видачі дозволу** одержує від авторизаційного сервера "перепустку", що містить ім'я користувача і його мережеву адресу, час вводу запиту і ряд інших параметрів, а також унікальний серійний ключ. Пакет, що містить "пе-

репустку", передається також в зашифрованому по алгоритму DES вигляді. Після отримання і розшифровки "перепустки" сервер видачі дозволу перевіряє запит і порівнює ключі, а після цього дає дозвіл чи заперечення на використання мережевої апаратури або програм.

Серед інших подібних комплексних схем можна відзначити розроблену Європейською асоціацією виробників комп'ютерів (ECMA) систему Sesame (Secure European System for Applications in Multivendor Environment), призначену для використання у великих гетерогенних мережах.

Захист інформації при віддаленому доступі

По мірі розширення діяльності підприємств, зростання чисельності персоналу і появи нових філіалів, виникає необхідність доступу віддалених користувачів (або груп користувачів) до обчислювальних і інформаційних ресурсів головного офісу компанії. Компанія Datepro свідчить, що вже у 1995 році тільки в США число робітників, що постійно або тимчасово використовують віддалений доступ до комп'ютерних мереж, складало 25 мільйонів чоловік. Найчастіше для організації віддаленого доступу використовуються кабельні лінії, звичайні телефонні або виділені і радіоканали. В зв'язку з цим захист інформації, що передається по каналам віддаленого доступу, вимагає особливого підходу.

Так, в мостах і маршрутизаторах віддаленого доступу застосовується сегментація пакетів — їх розподіл і передача паралельно по двом лініям, що робить неможливим "перехоплення" даних при незаконному підключенні хакера до однієї з ліній. До того ж при передачі даних процедура, що використовується для ущільнення пакетів, що передаються, гарантує неможливість розшифровки "перехоплених" даних. Крім того, мости і маршрутизатори віддаленого доступу можуть бути запрограмовані таким чином, що користувачі будуть обмежені в доступі до окремих ресурсів мережі головного офісу.

Розроблені і спеціальні прилади контролю доступу до комп'ютерних мереж по лініям, що комутуютья. Наприклад, фірмою AT&T пропонується модуль Remote Port Security Devise (PRSD), що являє собою два блоки розміром як звичайний модем: RPSD Lock ("замок, що подає"), в центральному офісі і RPSD Key ("ключ, що встановлюється"), що підключається до модему віддаленого користувача. KPIB Key і Lock дозволяють встановити декілька рівнів захисту і контролю доступу, а саме:

- **шифрування даних**, що передаються по лінії за допомогою цифрових ключів, що генеруються;
- **контроль доступу** в залежності від дня тижня або часу доби (усього 14 обмежень).

Широке розповсюдження радіо- та супутникових мереж в останні роки поставило розробників радіосистем перед необхідністю захисту інформації від хакерів, озброєних різноманітними скануючими приладами. Були застосовані різноманітні технічні рішення. Наприклад, в радіомережі компанії RAM Mobil Date інформаційні пакети передаються через різні канали і базові станції, що робить практично неможливим для сторонніх зібрати всю інформацію, що передається, воєдино. Активно використовуються в радіомережах і технології шифрування даних за допомогою алгоритмів DES і RSA.

Розділ 11

Системи захисту в Internet

"Чим більше знаємо, тим ширший обрій нашого невігластва"
(Народна мудрість)

Інформаційна безпека та Internet несумісні за самою природою Internet. Інформаційна безпека була спочатку заснована і використовувалась виключно для корпоративних мереж, однак нині, за допомогою єдиного протоколу TCP/IP та єдиного адресного простору, реалізується не лише в корпоративних та відомчих мережах (державні, комерційні, воєнні тощо), які за своєю природою є мережами з обмеженим доступом, а й в мережах для рядових користувачів, які мають можливість отримати прямий доступ до Internet через свої домашні комп'ютери, за допомогою модемів та телефонної мережі загального користування.

Як відомо, чим легший доступ до мережі, тим слабша її інформаційна захищеність, тому можна казати про те, що початкова простота доступу до Internet є дуже небезпечна, тому що користувач може навіть і не мати уявлення про те, що файли з його інформацією були скопійовані, не кажучі вже про можливість їх псування та коригування.

Що ж визначає це швидке зростання Internet, яке подвоюється кожного року? Відповідь дуже проста — низька вартість ПЗ, легкий да дешевий доступ до Internet (чи за допомогою IP-адреси, чи за допомогою провайдера), тобто до всіх світових інформаційних ресурсів.

Розплатою за користування Internet є загальне зниження інформаційної безпеки в Internet. Тому для запобігання несанкціонованому доступу до своїх інформаційних ресурсів, всі корпоративні та відомчі мережі, а також підприємства, які використовують технологію Internet, ставлять брандмауери (*firewall*) між внутрішньою мережею та Internet, що фактично означає вихід з єдиного адресного простору. Ще більшу безпеку надає відхилення від протоколу TCP/IP та доступ в Internet через шлюзи.

Цей процес можна виконати разом з побудовою всесвітньої інформаційної мережі загального використання, на базі мережевих комп'ютерів, які за допомогою мережевої карти та кабельного модему, забезпечують швидкісний доступ до Internet (10 Мбіт/с і більше) через мережу кабельного телебачення.

Для вирішення цих та інших питань, при переході на нову архітектуру, необхідно передбачити наступне:

- по-перше — ліквідувати фізичний зв'язок між Internet (всесвітньою інформаційною мережею загального використання) та корпоративними і відомчими мережами, забезпечивши їм лише інформаційний зв'язок через систему WWW;
- по-друге — замінити маршрутизатори на комунікатори, виключивши обробку в вузлах IP-протоколу, замінивши її на режим трансляції кадрів Internet, при якому процес комунікації зводиться до простого порівняння MAC-адрес;

- по-третє — перейти в новий єдиний адресний простір на базі фізичних адрес доступу до середовища (MAC-рівень), прив'язаний до географічного розміщення мережі, і який дозволяє в рамках 48-біт створювати адреси для більш, ніж 64 триліонів незалежних вузлів.

Безпека даних — це одна з основних проблема Internet. Трапляється все більше і більше випадків, коли комп'ютерні злочинці, які використовують все більш досконалі методи, проникнення в чужі бази даних. Зрозуміло, що це не сприяє популярності Internet в ділових колах. Одна тільки думка, про те, що конкуренти зможуть отримати доступ до архівів комерційних даних, змушує керівництво корпорацій відмовлятися від використання відкритих інформаційних систем. Проте, спеціалісти стверджують, що у таких міркувань не має підстав. Так, у компаній, що мають доступ до відкритих інформаційних систем, і у приватних мереж однакові шанси стати жертвами комп'ютерного терору.

Кожна організація, яка має справу з якими-небудь інформаційними цінностями, рано чи пізно виявляє спроби зазіхання на них. Передбачливі починають планувати захист заздалегідь, а непередбачливі — після першого значного "проколу". Так чи інакше, доводиться вирішувати, що і від кого захищати.

Як правило, перша реакція на погрозу — сховати "цінності" (важливу інформацію) в недосяжне місце і приставити до них охорону. Це відносно нескладно, якщо мова йде про такі "цінності", які довго не знадобляться, але досить складно, якщо планується постійно використовувати їх. Кожне звертання в сховище за "цінностями" потребує виконання особливої процедури, віднімає час та створює додаткові незручності.

В цьому полягає дилема безпеки: доводиться робити вибір між захищеністю інформації та її доступністю.

11.1. Використання брандмауерів

Зараз не потрібно доводити той факт, що при підключенні до Internet будь-яка локальна мережа піддається ризику у сенсі порушення конфеденційності її інформації. Одним з найбільш розповсюджених механізмів захисту від Internet-бандитів є використання міжмережевих екранів — брандмауерів (*firewalls*).

Але треба зауважити, що через непрофесіоналізм адміністраторів та недоліки деяких типів брандмауерів, приблизно 30% руйнувань сталось після встановлення захисних ситем.

Незважаючи на невпорядкованість в цій області, будь-яка діяльність по розробці, впровадженню та використанню засобів захисту інформації регулюється, а всі системи, що використовуються при цьому, підлягають обов'язковій сертифікації.

В наш час, багато уваги приділяється питанням безпеки даних в розподілених системах. Розроблено багато засобів для забезпечення інформаційної безпеки, призначених працювати на різноманітних комп'ютерах, з різними ОС. Як один з напрямків, можна виділити брандмауери, призначені контролювати доступ до інформації з боку користувачів зовнішніх мереж.

Розглянемо основні поняття екранних систем, а також вимоги, що висуваються до них. На прикладі пакету Solstice FireWall-1 розглянемо декілька типових випадків

використання таких систем, особливо при забезпеченні безпеки Internet-підключень. Проаналізуємо декілька унікальних властивостей системи Solstice FireWall-1.

11.2. Призначення екрануючих систем

Проблема міжмережевого екранування формулюється наступним чином. Нехай ми маємо дві інформаційні системи чи дві множини інформаційних систем. Екран (брандмауер) — це засіб розмежування доступу клієнтів однієї множини до інформації, яка зберігається в іншій множині (рис. 11.1).

Рис. 11.1. Екран (брандмауер)

Екран виконує свої функції, контролюючи всі інформаційні потоки між цими двома множинами інформаційних систем, працюючи як "інформаційна мембрана". В цьому розумінні екран можна уявити собі як певний набір фільтрів, що аналізують інформацію, яка проходить через них, та на основі закладених в них алгоритмів вирішують, пропускати цю інформацію чи ні. До того ж, така система котнролює та фіксує всі спроби незаконного вторгнення і додатково сигналізує про ситуації, що потребують негайного втручання.

Взагалі, екрануючі системи роблять несиметричними. Для екрану визначаються поняття внутрішнього (*inside*) та зовнішнього (*outside*) середовища. Задача екрану полягає в захисті внутрішнього середовища мережі від ворожого оточення. Прикладом суперворожого оточення є Internet.

Розглянемо більш детально проблеми, які виникають при побудові екрануючих систем. При цьому, будемо розглядати не лише проблему безпечного підключення до Internet, а й відмінності доступу в самій корпоративній мережі:

- Основна вимога до таких систем — це забезпечення безпеки внутрішній мережі та повний контроль над зовнішніми підключеннями та сеансами зв'язку.
- Екрануюча система повинна мати потужні та гнучкі засоби для простого та повного втілення в життя політики безпеки організації, просту реконфігурацію системи при зміні структури мережі.
- Екрануюча система повинна працювати непомітно від користувача локальної мережі та ускладнювати виконання нелегальних дій.
- Екрануюча система повинна працювати достатньо ефективно та забезпечувати обробку всього вхідного та вихідного трафіку в період "пікових" режимів.
- Система безпеки повинна бути надійно захищена від несанкціонованого доступу, оскільки вона є ключем до конфіденційної інформації організації.
- Якщо у організації є декілька зовнішніх підключень, в тому числі і у віддалених філіалах, екрануюча система повинна мати можливість централізовано забезпечувати єдину політику безпеки.
- Екрануюча система повинна мати засоби авторизації доступу користувачів через зовнішнє підключення.

11.3. Структура системи Solstice FireWall-1

Класичним прикладом, на якому можна проілюструвати всі ці принципи, є програмний комплекс Solstice FireWall-1 компанії Sun Microsystem. Цей пакет неодноразово нагороджувався на різних конкурсах. Він має багато корисних особливостей, що й виділяє його серед продуктів аналогічного типу. Розглянемо рис. 11.2.

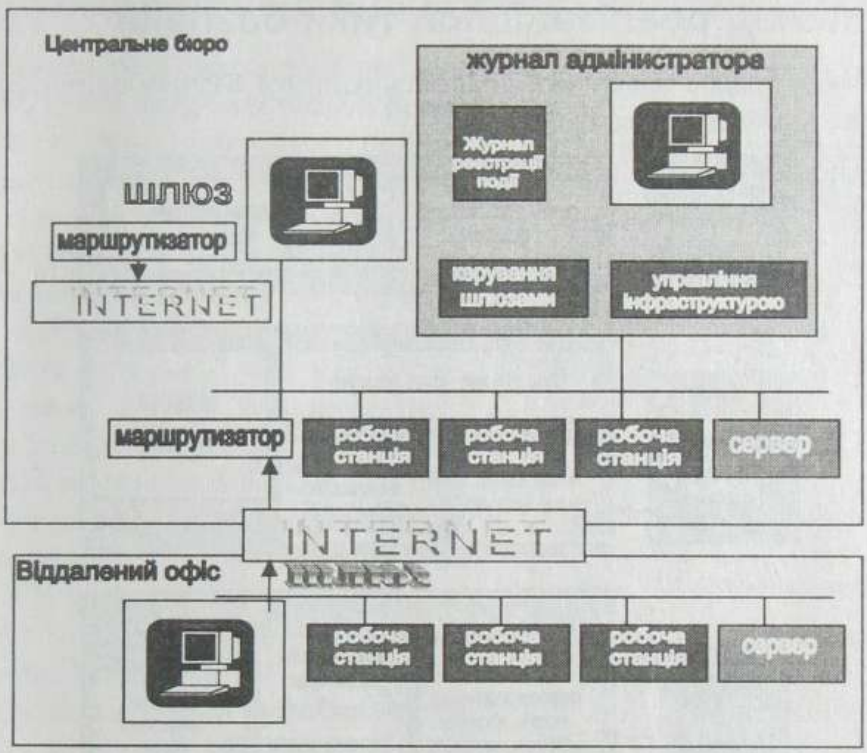


Рис. 11.2. Основні компоненти Solstice FireWall-1

Центральним для системи FireWall-1 є модуль керування усіма компонентами. З цим модулем працює адміністратор безпеки мережі.

Адміністратору безпеки мережі для конфігурації комплексу необхідно виконати наступні дії:

- визначити об'єкти, що приймають участь в процесі обробки інформації;
- описати мережеві протоколи та сервіси, з якими мають працювати прикладні програми;
- за допомогою введених понять, описати політику розмежування прав доступу в наступних термінах: "Групі користувачів А дозволений доступ до ресурсу В за допомогою сервісу С, але про це треба зробити помітку у регістраційному журналі".

Сукупність таких записів компілюється у виконавчу форму блоком керування і далі передається на виконання в модулі функцій.

Модулі фільтрації можуть бути розташовані на комп'ютерах-шлюзах, віддалених серверах чи в маршрутизаторах, як частина конфігураційної інформації. На сьогодні підтримуються два типи маршрутизаторів: Cisco IOS 9.X, 10.X, а також BayNetwork (WellFleet) os v.8.

Модулі фільтрації проглядають усі пакети, що поступають в мережеві інтерфейси, і, в залежності від заданих правил, пропускають чи відкидають ці пакети з відповідним записом у реєстраційному журналі. Треба відмітити, що працюючи безпосередньо з драйверами мережевих інтерфейсів, вони обробляють весь пакет даних і мають повну інформацію про пакети, що передаються.

11.4. Приклад реалізації політики безпеки

Розглянемо процес практичної реалізації політики безпеки організації за допомогою Solstice FireWall-1 (рис. 11.3).

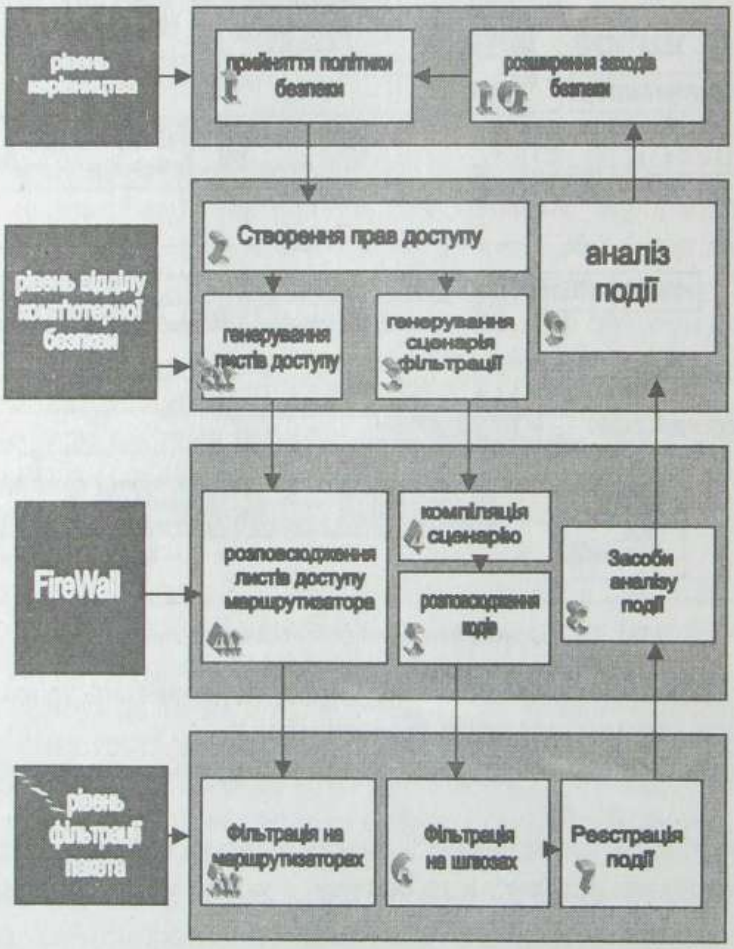


Рис. 11.3. Реалізація політики безпеки організації за допомогою Solstice FireWall-1

Важливо відзначити, що правила політики безпеки розробляються та затверджуються на рівні керівництва організації. Після затвердження цих правил, їх треба втілити у життя. Для цього їх треба перевести в структуру типу бази правил системи FireWall-1.

На основі цієї бази правил формуються списки доступу до маршрутизаторів та сценарії роботи фільтрів на мережевих шлюзах. Списки та сценарії далі переносяться на фізичні компоненти мережі, після чого правила політики безпеки вступають в дію.

В процесі роботи фільтри пакетів на шлюзах та серверах генерують записи про всі події, які було наказано відслідковувати, а також запускають механізми безпеки, які потребують від адміністратора негайної реакції.

На основі аналізу записів, зроблених системою, відділ комп'ютерної безпеки може розробити пропозиції щодо зміни та подальшого розвитку політики безпеки.

11.5. Ще один приклад реалізації політики безпеки

Розглянемо тепер випадок, коли початкова конфігурація мережі змінюється, а разом з нею зазнає змін і політика безпеки.

Нехай є рішення встановити в організації декілька загальнодоступних серверів для надання інформаційних послуг. Це можуть бути, наприклад, сервери WWW, FTP чи інші інформаційні сервери. Оскільки такі системи відокремлені від роботи решти мережі організації, для них часто виділяють свою власну підмережу, яка має вихід в Inernet через шлюз (рис. 11.4).

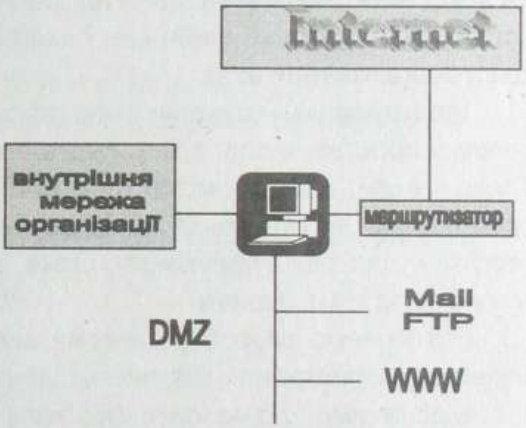


Рис. 11.4. Схема шлюзу Inernet

Оскільки в попередньому варіанті локальна мережа вже була захищеною, то все, що потрібно зробити — це дозволити відповідний доступ у виділену підмережу. Це робиться за допомогою одного рядка в редакторі правил. Така ситуація є стандартною при зміні конфігурації FireWall-1. За звичайних умов, для цього треба змінити один чи декілька рядків в наборі правил доступу, що ілюструє потужність засобу конфігурації та загальну продуманість архітектури FireWall-1.

11.6. Аутентифікація користувачів при роботі з FTP

Solstice FireWall-1 дозволяє адміністратору встановлювати різні режими роботи з інтерактивними засобами FTP та Telenet для різних користувачів та для груп користувачів. При встановленому режимі аутентифікації, Solstice FireWall-1 переводить стандартні FTP- і Telenet-процеси UNIX на свої власні, розташовуючи їх на шлюзі, закритому за допомогою модулів фільтрації пакетів. Користувач, що бажає розпочати інтерактивну сесію по FTP (це має бути "дозволений" користувач, в дозволений для нього час), може це зробити лише через вхід на такий шлюз, де виконується вся процедура аутентифікації. Вона задається при дефініції користувачів чи груп користувачів і може здійснюватись наступними способами:

- UNIX-пароль;
- програма S/Key генерування одноразових паролів;
- картки SecurID з апаратною генерацією одноразових паролів.

Розділ 12

Криптосистеми

"Спочатку подумай, а потім помовч"
(Народна мудрість)

Найбільш широко відомою і ефективною криптосистемою з відкритим ключем, що була запропонована Ривестом, Шаміром і Адлеманом (Rivest, Shamir, Adleman), є система RSA. Вона заснована на дивовижно простій теоретико-числовій (можна сказати, навіть арифметичній) ідеї і широко використовується для захисту від багатьох крипто-аналітичних атак.

Ідея полягає у використанні того простого факту, що легко перемножити два великих простих числа, проте надзвичайно важко розкласти на множники їх добуток. Таким чином, добуток може бути використано в якості ключа шифрування. Вихідні прості числа не можуть бути відновлені з їх добутку. З іншого боку, ці прості числа необхідні для розшифрування. Отже, ми маємо ефективний каркас для криптосистеми з відкритим ключем.

Розглянемо розробку системи шифрування і розшифрування, а також обидва аспекти використання: таємність і ідентифікацію.

Відзначимо, що не існує формального доказу фактів: факторизація важкообчислювана в спеціальному змісті, використовуваному в RSA, і факторизація необхідна для криптоаналізу RSA, тобто не існує криптоаналітичних методів, що не використовують факторизацію. Є множина емпіричних підтверджень для обох фактів.

12.1. Постановка задачі

Нехай p та q — два різних великих випадково обраних простих числа (які мають 100 розрядів у їхньому десятковому представленні).

Позначимо

$$n = pq \text{ і } \varphi(n) = (p - 1)(q - 1),$$

(тут φ — функція Ейлера). Випадково виберемо велике число $d > 1$, таке, що $(d, \varphi(n)) = 1$, і обчислимо e, $1 < e < \varphi(n)$, що задовольняє рівнянню:

$$ed \equiv 1 (mod \varphi(n))$$

Числа n, e і d називаються ***модулем експонентого шифрування*** і ***розшифрування*** відповідно. Числа n і e утворять ***відкритий ключ шифрування*** тоді як числа p, q, n і d, що залишились, формують ***секретну вказівку***. Вочевидь, що секретна вказівка містить у собі взаємозалежні числа. Наприклад, знаючи p, неважко обчислити q, n і d.

При шифруванні вихідний текст підноситься в ступінь e по модулю n. При розшифруванні, криптотекст підноситься в ступінь d по модулю n.

Більш детально: припустимо, що вихідний текст кодувався десятковим числом (аналогічно можна використовувати і двійкове представлення). Дане число потім ділиться на блоки відповідного розміру, які шифруються окремо.

Розмір блоків визначається як єдине ціле число i, що задовольняє нерівностям $10^{i-1} < n < 10^i$. У деяких випадках розмір блоків можна вибрати рівним $i-1$, проте, якщо важлива однозначність розшифрування, потрібно бути впевненим у тому, що кожному блоку відповідає число, менше за n.

Якщо w є блоком вхідного тексту, а c — відповідним блоком криптотексту, то шифрування може бути описане в термінах наступного рівняння:

$$c = (w^e, \ mod\ n).$$

Тепер покажемо коректність розшифрування.

Лема. Для w та c, визначених вище,

$$w = c^d\ (mod\ n).$$

Отже, якщо розшифрування однозначне, то $w = c^d\ (mod\ n)$.

Доведення. Виходячи з вибору d, існує позитивне ціле число j, таке, що

$$e^d = j\varphi(n) + 1.$$

Припустимо спочатку, що ні p, ні q не ділять w. За теоремою Ейлера

$$w^{\varphi(n)} \equiv 1\ (mod\ n),$$

звідки

$$w^{ed-1} \equiv 1\ (mod\ n).$$

Отже,

$$c^d \equiv (w^e)^d \equiv \mathrm{w}\ (mod\ n).$$

Якщо в точності одне з чисел p і q, нехай p, ділить w, то

$$w^{q-1} \equiv 1\ (mod\ n).$$

Тоді

$$w^{\varphi(n)} \equiv 1\ (mod\ q);$$
$$wj^{\varphi(n)} \equiv 1\ (mod\ q);$$
$$w^{ed} \equiv w\ (mod\ q).$$

Оскільки, останнє рівняння є вірним, то по модулю p одержуємо:

$$w \equiv c^d\ (mod\ n).$$

Якщо і p, і q ділять w, то маємо

$$w^{ed} \equiv w\ (mod\ n),$$

звідки, як і раніше, випливає

$$w \equiv c^d\ (mod\ n).$$

12.2. Аналіз задачі

Розглянемо розробку криптосистеми, тобто процес генерування різних її частин. В цілому, коли говорять, що взяте випадкове число або ж що-небудь вибиране випадково, то використовується генератор випадкових чисел, наприклад комп'ютерна

програма, що генерує таку послідовність розрядів, яка має якнайбільше статистичних властивостей випадкової послідовності.

Для визначення двох величезних випадкових простих чисел p і q довільним чином обирається непарне ціле число r обмеженого розміру (скажімо, 100 розрядів) і перевіряється на простоту. У випадку негативної відповіді перевіряється число $r + 2$ і т. ін. Згідно з теоремою про прості числа, існує приблизно

$$10^{100}/\ln 10^{100} - 10^{99}/\ln 10^{99}$$

100-розрядних простих чисел. Якщо це число порівняти з кількістю $(10^{100} - 10^{99})/2$ всіх непарних 100-розрядних чисел, то очевидно, що можливість успіху для конкретного тесту дорівнює приблизно 0,00868.

Після того як p і q обрано, "кандидати" для d перевіряються за допомогою алгоритму Евкліда. Коли d задовольняє умові $(d, \varphi(n)) = 1$, то ланцюжок рівності з алгоритму Евкліда дає відразу і e.

Операцією, необхідною при шифруванні і розшифруванні, є ***модулярне піднесення до ступеню***, тобто обчислення $(a^r, mod\ n)$. Це можна зробити набагато швидше, ніж за допомогою повторюваного множення на себе. Розглянемо метод, який називається ***методом послідовного піднесення до квадрату***. Після кожного піднесення до квадрату результат зводиться по модулю n, при цьому ніколи не виникають числа більші за n^2.

Пояснимо це більш докладно. Розглянемо двійкове представлення

$$r = \sum x_j 2^j; \quad x_j = 0,1; \quad j = \overline{1,k}; \quad k = \lfloor log2r \rfloor + 1$$

Припустимо, що нам відомі всі числа

$$(a^{2^j}, mod\, n),\ 0 \le j \le k,$$

тоді $(a^r, mod\ n)$ може бути обчислене за допомогою не більш, ніж k–1 множень і зведенням кожного добутку по модулю n. Таким чином, достатньо обчислити числа $(a^{2j}, mod\ n)$, що потребує k модульних піднесень до квадрату і додатково не більш, як k–1 модульних добутків. Це означає, що обчислюється не більше $2k$–1 добутків з обома множниками, меншими за n, із зведенням добуків по модулю n. Якщо r є великим і відоме $\varphi(n)$, то r може бути спочатку взяте по модулю $\varphi(n)$.

Для реалізації даної криптосистеми використовуються алгоритми множення і ділення великих чисел. Розглянемо кожний з них більш детально.

12.2.1. Множення

Нехай у нас є два $2n$-розрядних числа $u = (u_{2n-1} \ldots u_0)$ і $v = (v_{2n-1} \ \ldots v_0)$. Тоді можна записати $u = 2^n U_1 + U_0$, $v = 2^n V_1 + V_0$, де $U_1 = (u_{2n-1} \ldots u_n)$ і $v_1 = (v_{2n-1} \ \ldots v_n)$ — найбільш значима половина, а $U_0 = (u_{n-1} \ldots u_0)$ і $v_0 = (v_{n-1} \ \ldots v_0)$ — найменш значима половина (для числа u і v відповідно). Тепер маємо:

$$uv = (2^{2n} + 2^n)U_1V_1 + 2^n(U_1 - U_0)(V_0 - V_1) + (2^{2n} + 1)U_0V_0.$$

Ця формула зводить задачу множення $2n$-розрядних чисел до трьох операцій множення n-розрядних чисел:

$$U_1V_1, (U_1 - U_0)(V_0 - V_1), U_0V_0,$$

та деяких простих операцій зсуву і додавання.

12.2.2. Ділення

За даними невід'ємними цілими числами $u = u_1u_1...u_{m+n}$ і $v = v_1v_1...v_n$, поданими в системі числення з основою b, з припущення, що $v_1 \neq 0$ і $n > 1$, знаходимо $u/v = (q_0q_1...q_m)$ і залишок $u \bmod v = (r_0r_1...r_n)$.

1. **Нормалізувати**. Встановити $d = b \operatorname{div} (v_1 + 1)$,
$$u_0u_1...u_{m+n} = (u_1u_2...u_{m+n})d, \quad v_0v_1...v_n = (v_1v_2...v_n)d.$$
2. **Початкова установка для** j. Встановити $j = 0$.
3. Обчислити q'. Якщо $u_j = v_1$, то встановити $q' = b - 1$, інакше $q' = (u_jb + u_{j+1}) \operatorname{div} v_1$. Перевірити, чи виконується нерівність $v_2q' > (u_j + u_{j+1} - q'v_1)b + u_{j+2}$. Якщо виконується, то зменшити q' на 1 і повторити перевірку.
4. **Помножити і відняти**. Замінити $u_j...u_{j+n}$ на $u_j...u_{j+n} - v_1...v_n \cdot q'$.
5. **Перевірити залишок**. Встановити $q_j = q'$. Якщо результат кроку 4 був від'ємний, перейти на крок 6, інакше — на крок 7.
6. **Додавання, що компенсує**. Зменшити q_j на 1 і додати $v_1v_2...v_n$ і $u_j...u_{j+n}$.
7. **Цикл по** j. Збільшити j на 1. Якщо тепер $j \leq m$, то повернутися на крок 3.
8. **Денормалізація**. Тепер $q_0q_1...q_m$ є шукана частка і для отримання шуканого залишку достатньо $u_{m+1}...u_{m+n}$ розділити на d.

При піднесенні до ступеню використовується операція модулярної редукції. Існує кілька алгоритмів для її реалізації. Розглянемо найшвидший з відомих нині алгоритмів, запропонований А. В. Анісімовим.

Нехай необхідно обчислити $x \bmod m$, де
$$x = x_{2k-1}b^{2k-1} + x_{2k-2}b^{2k-2} + \ldots + x_0,$$
$$m = m_{k-1}b^{k-1} + m_{k-2}b^{k-2} + \ldots + m_0.$$

Позначимо r_{i+1} залишок від ділення b^{i+1} на m.

1. Покладемо
$$v = x_{k-2}b^{k-2} + \ldots + x_0, \quad u = 0$$
2. Покладемо $i = 2k-1$.
3. Обчислюємо $y = r_{i+1}\, x_i$ і покладемо $u = u + y$.
4. Якщо $u > mb$, зменшуємо u на mb.
5. Зменшуємо i на 1 і, якщо $i \geq k-1$, переходимо на крок 3.
6. Обчислюємо $t = (-m_0^{-1}\, u_0) \bmod b$.
7. Покладемо $u = (u + tm) \operatorname{div} b$.
8. Якщо $u > m$, то $u = u - m$.
9. Додаємо v до u, і, якщо $u > m$, то зменшуємо u на m.
10. Тепер $u = x \bmod m$.

Система RSA також може бути розглянута відповідно до загальних принципів побудови криптосистеми з відкритим ключем. Початкова постановка тут не так зрозуміла, як наприклад, для криптосистеми на основі задачі про укладку рюкзака.

В якості складної задачі P можна вибрати факторизацію n, коли відомо, що n — це добуток двох простих чисел. "Легкою" підзадачею P_{easy} є задача P з додатково ві-

домою $\varphi(n)$. Змішаною версією P_{easy} є просто сама P. Або в якості складної задачі P можна можна взяти розв'язок рівняння:

$$X^e = c(mod\ \ n),$$

коли відома трійка (e,c,n), формуюча RSA. P_{easy} в цьому випадку — це задача P з додатково відомою однією з величин $\varphi(n)$, d, p, q.

Очевидні наступні два пункти:

- Немає протиріччя між тим, що для заданого числа n можна визначити, чи є воно складовим чи ні, і тим, що n не може бути факторизоване. Перший факт потрібен при розробці RSA-систем, тоді як факторизація n розкриває криптосистему. Дійсно, існує багато результатів у формі: "якщо n — просте, то виконується умова $C(n)$". Отже, якщо зауважити, що $C(n)$ не виконується, то n — складене, але з цього не випливає, що можна факторизувати n.
- Для функцій дійсних змінних не існує різниці з точки зору складності між піднесенням до ступеню і обчисленням логарифмів. У дискретному випадку модульне піднесення до ступеню є "легким", у той час як обчислення логарифмів обраховується "складно".

12.3. Ідентифікація і цифровий підпис

Нехай e, d, n — експоненти шифрування і розшифрування, A — модуль використання. Будемо вимагати спочатку, щоб при передачі повідомлень необхідний був тільки підпис, а не таємниця. Тоді A посилає пару:

$$(w, D_A(w)), \quad \text{де} \quad D_A(w) = (w^d, mod\ n).$$

Одержувач може перевірити підпис, застосовуючи опубліковану експоненту шифрування e. Оскільки тільки A володіє d, то ніхто інший не може підписати повідомлення w.

Проте супротивник може вибрати число c, обчислити:

$$E_A(c) = (c^e, mod\ n),$$

і з успіхом підтвердити, що c — підпис A у повідомленні $E_A(c)$. Цей метод атаки може бути використаний лише для підробки підписів випадкових повідомленнь: тільки A може підписати обране повідомлення.

Такі непередбачені повідомлення будуть мати сенс із дуже невисокою ймовірністю, наприклад, коли вихідні повідомлення утворюються числовим кодуванням деякої природної мови. Тоді надмірність у повідомленнях висока, і тільки дуже маленькі порції блоків визначеного розміру є числовими кодами частин змістовного вихідного тексту.

Крім ступеню непевності важливий також тип непевності вихідних текстів. Зокрема, ні інверсія змістовного повідомлення, ні добуток двох змістовних повідомлень не буде змістовним. Інакше перехоплювач, знаючи підпис s_i, A в повідомленні w_i, $i = 1, 2$, може підписати правильним підписом A повідомлення $(w_1w_2,\ mod\ n)$ і $(w_1{-}1,\ mod\ n)$, використовуючи $(s_1s_2,\ mod\ n)$ і $(s_1{-}1,\ mod\ n)$.

Для побудови нових підписів потрібні лише n та підпис для w_i. Цей метод підробки призведе до добутку більш, ніж двох множників. Оскільки, d — завжди непар-

не, то ми можемо підписати також $(-w_1, mod\ n)$, використовуючи $(-s_1, mod\ n)$. Вимагатимемо тепер, щоб при посиланні повідомлення були потрібні і підпис, і секретна передача: посилаючи підписане повідомлення до Y, A — спочатку підписує його, використовуючи (d,n), а потім шифрує результат, застосовуючи пару (e,n).

Y спочатку розшифровує текст, використовуючи експоненту розшифрування d, після чого оригінальне повідомлення може бути отримане застосуванням відкритого ключа шифрування e. Присутність d у повідомленні гарантує, що воно надійшло від A. Але, як і раніше, треба бути обережними, тому що можлива підробка підпису в неочікуваних повідомленнях.

Існує також і інша складність, що виникає з того, що A і B використовують різні модулі. Нехай $n > n_B$. Тоді $D_A(w)$ не обов'язково лежить в інтервалі $(1, n-1)$ і зводити по модулю n, що буде обов'язково виконуватися при легальному розшифруванні, дуже важко. Існує два шляхи подолання цієї складності.

- Всі користувачі домовляються про загальний поріг t. Кожний користувач A вибирає два ключі RSA: один для підпису, інший для шифрування. Частини позначаються s і e відповідно. Кожний користувач A піклується про те, щоб $nA^S < i < nA^e$. Складності, описані вище, не з'являються, якщо A посилає повідомлення w до B в представленні:

$$E_B^e(D_A^s(w))$$

- Можна також уникнути граничного числа, якщо повідомлення від A до Y посилаються у вигляді

$$E_B(D_A(w));$$

або

$$D_A(E_B(w));$$

в залежності від того, яка з нерівностей $n < n_B$, $n_B < n$ виконується.

12.4. Генерація простих чисел

Розглянемо спочатку вибір p та q. Вони повинні бути випадковими простими числами і не знаходитися в жодній із відомих таблиць простих чисел. При факторизації можна завжди перевірити всю таблицю або перебрати послідовність простих чисел специфічного виду. Ці два простих числа також не повинні бути близькими одне до одного. Інакше (при p та q) $(p - q)/2$ мале, а $(p + q)/2$ лише трохи більше за $\sqrt{n}$. Більше того,

$$(p+q)^{2/4} - n = (p-q)^{2/4};$$

і, отже, ліва сторона рівності є повним квадратом.

Факторизуючи n, перевіряємо цілі числа $x > \sqrt{n}$ доти, поки не знайдемо таке, що $x^2 - n$ є повним квадратом, скажімо, y^2. Тоді $p = x + y$ та $q = x - y$. Взагалі рекомендується, щоб двійкові представлення p і q відрізнялись по довжині на кілька бітів.

При виборі p і q також повинно бути розглянуте і $\varphi(n)$. Обидва числа $p-1$ і $q-1$ парні, з чого випливає, що $\varphi(n)$ ділиться на 4. Нехай $(p-1, q-1)$ буде великим, і тому найменше спільне парне $p-1$ і $q-1$ мале в порівнянні з $\varphi(n)$. Тоді будь-яка інверсія e по модулю i буде працювати як експонента розшифрування. Для того, щоб в цьому

випадку набагато легше було знайти d простою перевіркою, $p-1$ і $q-1$ не повинні мати великий спільнй дільник.

Виродженими є випадки, коли одне з $p-1$ і $q-1$, скажімо $p-1$, є дільником іншого. Тоді достатньо резглянути інверсії e по модулю $p-1$. Розробник криптосистеми також повинний уникати ситуації, коли в розкладанні $\varphi(n)$ на множники беруть участь лише дуже маленькі прості числа.

Нехай усі прості множники r $\varphi(n)$ менші за деяке ціле k. Оскільки $[\log_r n]$ є найбільшим показником ступеню множника r, який мабуть є дільником $\varphi(n)$, то можна перебрати всіх кандидатів v для $\varphi(n)$ і перевірити криптотекст піднесенням до ступеню $(v+1)/e$, забезпечуючи, щоб показник був цілим числом.

Шляхом подолання обох труднощів, що стосуються $\varphi(n)$, є розгляд тільки "безпечних" простих чисел. За визначенням, просте число p ***безпечне*** тоді і тільки тоді, коли $(p-1)/2$ також є простим числом. Вочевидь, що генерація безпечних простих чисел p і q набагато складніша, ніж генерація звичайних простих чисел.

Є інші властивості p, q $\varphi(n)$, що можуть полегшити факторизацію і розшифрування. Розробник криптосистеми RSA повинен враховувати їх, а також ті властивості, які можуть бути виявлені в майбутньому. Дійсно, найбільш відомі серед подібних властивостей враховуються в існуючому для RSA комп'ютерному забезпеченні.

Вибір p і q також важливий з погляду можливої атаки, заснованої на послідовному розшифруванні. Це означає, що процес починається з криптотексту c_0 і обчислення чисел:

$$c_i = (C^e_{i-1}, mod\ n),\ I = 1, 2, 3\ldots\ ;$$

поки не знайдуть осмисленого c_i. Неважко показати, що можливість успіху такої атаки малоймовірна, якщо $p-1$ і $q-1$ мають великі прості множники p' і q', а також $p'-1$ і $q'-1$ мають великі прості множники. Легко оцінити можливість успіху за розмірами використовуваних простих множників.

Нехай p і q вже обрані. Роглянемо вибір e і d. Він не є незалежним, тому що одне з них визначає інше. Звичайно d не повинне бути малим, інакше воно може бути знайдене перебором. Це є аргументом на користь того, чому ми при проектуванні системи спочатку фіксуємо d, а потім обчислюємо e.

Проте маленьке e також може призвести до ризику для безпеки. Якщо однакове повідомлення посилається декільком одержувачам, то криптоаналіз може стати можливим. З точки зору безпеки, бажано, щоб обидва e і d були великими. Малі e і d більш прийнятні, коли обмежено час виконання шифрування або розшифрування. Невеликі експоненти корисні, коли існує велика різниця у обчислювальних потужностях між двома сторонами, що зв'язуються.

Типовим є приклад, коли RSA використовується для зв'язку користувачів кредитних карток з великим комп'ютером. Тоді дуже корисно для користувачів мати невелике d, а для великого комп'ютера мати невелике e. У таких ситуаціях повинен бути досягнутий компроміс між безпекою і доступними обчислювальними потужностями.

Нарешті звернемо увагу на те, що в кожній криптосистемі RSA деякі блоки похідного повідомлення при шифруванні переходять самі в себе. Дійсно, існує щонайменше чотири блоки вихідного тексту, що задовольняють обом умовам $E(w)=w$ і $(w,n) = 1$. Вочевидь, що

$$(1^e, mod\ p) = (1^e, mod\ q) = 1\ ;$$

$$((p-1)^e, mod\ p) = p-1,\ \ ((q-1)^e, mod\ q) = q-1.$$

Останні рівності випливають з того, що e завжди непарне. Ми одержуємо за китайською теоремою про залишки одночасне рішення рівнянь

$$x \equiv a(mod\ p)\ \text{ і }\ x \equiv b(mod\ q).$$

Коли вимагається, щоб a і b приймали незалежні значення ± 1, то отримаємо чотири числа w, що задовольняють

$$(w^e, mod\ \ n) = w.$$

Якщо умова $(w, n) = 1$ ігнорується, то $w = 0$ також може бути вихідним текстом і існує, щонайменше, дев'ять чисел w із $E(w) = w$. Це вочевидь точно так само, як і раніше, з огляду на те, що тепер можливими значеннями для a та b є також і 0.

Аналіз показує, що деяких вихідних текстів потрібно уникати. Також повинні ігноруватись деякі експоненти шифрування e. Якщо $e - 1$ є кратним обох чисел $p-1$ і $q-1$, то кожне w задовольняє рівності $E(w) = w$. Це безпосередньо випливає з теореми Ейлера. Таким чином, $e = \varphi(n)/2 + 1$ є особливо поганим вибором, хоча і лежить у звичайному діапазоні для e.

Невідомо, чи належить ця задача до класу P. Але це несуттєва перешкода з точки зору RSA, тому що конструюється лише прості числа визначеного розміру i, крім того, цілком прийнятними є стохастичні алгоритми з низькою можливістю неуспіху.

Такі стохастичні алгоритми працюють у більшості випадків як описано далі. Розглянемо тест на складеність $C(m)$. Якщо ціле число m успішно проходить тест, то воно є складеним. Якщо m не проходить тест, то m може бути простим. Ймовірність для m бути простим числом зростає з числом невдалих тестів на складеність.

Навіть якщо m не проходить тесту на складеність, постає дуже складна задача факторизації m. Як вже згадувалось, безпека RSA заснована на припущенні, що набагато легше знайти два великих простих числа p і q, ніж розкрити їх, якщо відомо тільки їхній добуток n. Це припущення засноване лише на емпіричних даних — доведених теорем такого роду не існує.

Оскільки ймовірність того, що згенероване розробником RSA-криптосистеми число p дійсно є складеним, надзвичайно мала, досліджуємо, що може означати така помилка. Якщо $p = p1*p2$, де $p1$, $p2$, як і q, є простими числами, то розробник працює з помилковою $\varphi_1(n) = (p-1)\ (q-1)$, тоді як правильною буде функція $\varphi(n) = (p_1-1)(p_2-1)(q-1)$.

Нехай u — найменше спільне кратне чисел $p_1 - 1$, $p_2 - 1$, $q - 1$. Нехай також $(w, n) = 1$. Тоді за теоремою Ейлера справедливі рівності

$$w^{p_1-1} \equiv 1\ (mod\ \ p_1),\ \ w^{p_2-1} \equiv 1\ (mod\ \ p_2),\ \ w^{q-1} \equiv 1\ (mod\ \ q),$$

і рівності $wu \equiv 1$ вірні для всіх трьох модулів. З цього випливає

$$w^u \equiv 1\ (mod\ \ n).$$

Вочевидь, u є дільником $\varphi(n)$. Якщо u є дільником також і $\varphi_1(n)$, то

$$w^{\varphi_1(n)+1} \equiv w\ (mod\ \ n),$$

звідки випливає, що шифрування і розшифрування виконуються так, якби p було простим числом.

За функцією φ_1 можна обчислити e, для якого знову буде виконуватись рівність $D(E(w)) = w$. Це ніяк не вплине на безпеку системи крім того, що найменше спільне кратне буде значно меншим за $\varphi(n)$.

Проте якщо u не є дільником $\varphi_1(n)$, то в більшості випадків $D(E(w)) \neq$ w, і цей факт, мабуть, буде помічений розробником.

Нехай m — непарне ціле число і $(w, m) = 1$. Якщо m просте, то за теоремою Ейлера

$$w^{m-1} \equiv 1 \pmod{m}.$$

Якщо m не є простим, то виконання цієї рівності можливе, але малоймовірне. У цьому випадку m є ***псевдопростим*** за *основою* w. Це негайно дає наступниий тест на складеність: m успішно проходить тест $C(m)$ тоді і тільки тоді, коли

$$w^{m-1} \neq 1 \pmod{n},$$

для деякого w_i, із $(w, m) = 1$.

Якщо m не задовольняє тест $C(m)$ для w', тобто виконується

$$w^{m-1} \equiv 1 \pmod{m},$$

то m ще може бути складеним.

Назвемо ціле число, для якого $(w, m) = 1$ і виконується рівність

$$w^{m-1} \equiv 1 \pmod{m},$$

свідком простоти числа m.

Як можна побачити, існують також і "помилкові свідки", для яких m є тільки псевдопростим. Метод, який з високою ймовірністю свідчить, що m просте, полягає в доборі великого числа свідків простоти m. Наведемо деяке теоретичне обгрунтування цього факту.

Лема. Всі або неменше половини цілих чисел w, де $1 < w < m$ *и* $(w, m) = 1$, є свідками простоти m.

Доведення. Нехай w не є свідком. Виконується

$$w^{m-1} \equiv 1 \pmod{m}.$$

Нехай, $w_i \leq i \leq t$ — це всі свідки. Тоді числа $u_i = (w_i \bmod m)$, $1 \leq i \leq t$, попарно взаємно прості і задовольняють умовам $1 \leq u_i < m$ і $(u_i, m) = 1$. Немає числа u_i, що може бути свідком, тому що

$$1 \equiv u_i^{m-1} \equiv w^{m-1}, \quad w_i^{m-} \equiv w^{m-1} \pmod{n};$$

буде суперечити

$$w^{m-1} \neq 1 \pmod{m}.$$

Існує стільки ж чисел u_i, скільки всіх свідків.

Ймовірнісний алгоритм працює в такий спосіб. Задавши m, вибираємо випадкове w, де $1 < w < m$. Найбільший спільний дільник (w, m) знаходимо за алгоритмом Евкліда. Якщо $(w, m) > 1$, то допускаємо, що m складене. Інакше обчислюємо $u = (w^{m-1} \bmod m)$ послідовними піднесенням до квадрату. Якщо $u \neq 1$, то допускаємо, що m складене. Якщо $u = 1$, то w — свідок простоти m, і ми маємо деяке обгрунтування, що m може бути простим.

Чим більше свідків ми знайдемо, тим сильнішим буде це обгрунтування. Коли ми знайдемо k свідків, то за лемою можливість того, що m буде складеним, не пере-

вищуватиме 2^{-k} тому, що в гіршому випадку всі числа $w((w, m) = 1$ і $w < m)$ є свідками. Якщо m — просте, то всі числа є свідками, і обгрунтування призводить до правильного висновку. Проте всі числа можуть бути свідками для m, що не є простим. Такі числа m називаються ***числами Кармішеля***. За визначенням, непарне складене число m називається числом Кармішеля тоді і тільки тоді, коли виконується

$$w^{m-1} \equiv 1 \pmod{m},$$

для всіх w з $(w, m) = 1$.

Легко довести, що число Кармішеля ніколи не є квадратом іншого числа і що непарне складене число m, що не є квадратом, є числом Кармішеля тоді і тільки тоді, коли для простого p, що є дільником m, $p-1$ є дільником $m-1$.

З цього випливає, що число Кармішеля повинно бути добутком не менше трьох різних простих чисел. Всі вони мають вигляд $(6i + 1)(12i + 1)(18i + 1)$, де три множники є простими числами. Всі числа такого вигляду, де три множники є простими числами, є числами Кармішеля.

Існують також числа Кармішеля іншого виду. Невідомо, чи є нескінченною множина чисел Кармішеля. Оцінка ймовірності 2^{-k} для алгоритму, описаного вище, не вірна, якщо число m, що перевіряється є числом Кармішеля. За допомогою цього алгоритму є тільки шанс виявити, що m складене, потрапивши при випадковому виборі на число w з $(w, m) > 1$.

12.4.1. Тест перевірки простоти Соловея-Штрассена

Він дуже схожий на тест, описаний вище, за винятком того, що замість умови

$$w^{m-1} \equiv 1 \pmod{m},$$

використовується інша умова

$$w^{(m-1)/2} \equiv (w/m) \pmod{m}.$$

Проте для останньої умови немає аналогів числам Кармішеля. Таким чином, чим більше свідків ми знайдемо, тим вище буде ймовірність того, що число яке перевіряється, є простим.

Лема. Якщо m — просте, то для всіх w

$$w^{(m-1)/2} \equiv (w/m) \pmod{m},$$

Непарні складові m, що відповідають цій рівності для деякого w із $(w, m) = 1$, називаються ***псевдопростими ейлеровими числами*** по підставі w. Тому що

$$w^{(m-1)/2} \equiv (w/m) \pmod{m}$$

спричиняє

$$w^{m-1} \equiv 1 \pmod{m},$$

то ейлерові псевдопрості числа по підставі w також є і псевдопростими числами по підставі w. Обернене невірно.

Лема. Якщо m — непарне складове число, то не більше половини цілих чисел w, де $(w, m) = 1$ і $1 \le w \le m$, задовольняють умові

$$w^{(m-1)/2} \equiv (w/m) \pmod{m}.$$

Тест Соловея-Штрассена використовує цю умову точно в тому ж змісті, що і використовував попередній алгоритм

$$w^{m-1} \equiv 1 \pmod{m}.$$

Перевіряючи простоту m, спочатку вибираємо випадкове число $w < m$. Якщо $(w, m) > 1$, то m — складене. Інакше перевіряється істинність

$$w^{(m-1)/2} \equiv (w/m) \pmod{m}.$$

Це привабливо, з точки зору складності, тому що значення (w/m) може бути швидко обчислене за допомогою закону квадратичної оборотності. Якщо

$$w^{(m-1)/2} \equiv (w/m) \pmod{m},$$

не виконане, то m — складене. Інакше w є свідком простоти m.

Вибираємо інше випадкове число менше за m і повторюємо процедуру. Після перебору k свідків допускаємо, що ймовірність того, що m складене, не перевищує 2^{-k}. Отриманий результат "сильніший", ніж за допомогою попереднього алгоритму, тому що не існує аналогів чисел Кармішеля при роботі з

$$w^{(m-1)/2} \equiv (w/m) \pmod{m}.$$

Існують числа m, що є ейлеровими псевдопростими точно на половині всіх можливих основ.

Існує й інша модифікація тесту перевірки простоти, де оцінка дійсно може бути поліпшена: не більш 25% можливих чисел є (помилковими) свідками простоти для складеного числа m.

12.4.2. Тест Міллера-Рабіна

Нехай m псевдопросте за основою w, тобто виконане

$$w^{m-1} \equiv 1 \pmod{m}.$$

Ідея полягає в послідовному видобуванні квадратних коренів наведеного вище рівняння і перевірки, що перше відмінне від 1 число в правій частині рівняння дійсно дорівнює –1.

Якщо m просте, то перше таке число повинно дорівнювати –1, тому що тільки тоді ± 1 є квадратними коренями по модулю m. Отже, отримаємо інший тест перевірки на простоту. Якщо m не задовольняє цьому тесту, тобто перше число, відмінне від 1, дорівнює -1, але m складове, то m називається ***сильним псевдопростим*** числом за основою w.

Розглянемо тест більш докладно. Нехай m — непарне складене число і s — максимальний ступінь двійки, що є дільником $m - 1$, тобто $m-1 = 2^s r$, де r непарне.

Виберемо число w з $1 \le w \le m$ і $(w, m) = 1$. Виходить, m — сильне псевдопросте по основі w тоді і тільки тоді, коли виконуються наступні умови:

$$w^r \equiv 1 \pmod{m} \quad \text{або} \quad w^{r*2^{sl}} \equiv -1 \pmod{m},$$

для деякого s_1, де $0 < s_1 < s$.

Зауважимо, що формальне визначення уточнює ідею отримання квадратних коренів з порівняння

$$w^{m-1} = w^{r*2^{sl}} \equiv 1 \ (mod \ m) .$$

Не існує квадратних коренів, якщо в лівій частині залишиться w^r. Можна показати, що сильне псевдопросте число m по основі w є також ейлеровим псевдопростим числом по основі w.

У тесті Міллера-Рабіна за заданим непарним цілим числом m спочатку обчислюємо $m - 1 = 2^{sr}$, де r непарне. Як і раніше, вибираємо випадкове число w і перевіряємо

$$w^r \equiv 1 \ (mod \ m) \ \text{ або } \ w^{r*2^{sl}} \equiv -1 \ (mod \ m) .$$

Якщо тест не виконується, то m — складене. Інакше w є свідком простоти m (у цьому випадку m — просте або сильне псевдопросте число по основі w), і повторюємо процедуру для іншого w. Після перебору k свідків простоти можна допустити, що можливість того, що m — складене, не перевищує 4^{-k}. Це є результатом нижченаведеної леми.

Лема. Якщо m — непарне складене ціле число, то m є сильним псевдопростим числом по основі w для не більше 25% від усіх w, що задовольняють нерівностям

$$1 \le w \le m.$$

Щоб бути майже впевненим у тому, що m — просте, зовсім не обов'язково перевіряти велике число основ w, якщо m є сильним псевдопростим числом по кожній з цих основ. Можна сформулювати навіть більш загальне твердження, вважаючи, що вірна "узагальнена гіпотеза Римана". За цим припущенням, якщо m — непарне складене ціле число, то

$$w^r \equiv 1 \ (mod \ m) \ ; \ w^{r*2^{sl}} \equiv -1 \ (mod \ m) \ ;$$

порушується, принаймні для одного $w < 2(1n\ m)2$. Таким чином, достатньо перевірити числа w тільки до цієї межі.

На цьому шляху тест Міллера-Рабіна трансформується в детермінований алгоритм з поліноміальним часом роботи. (Звичайна гіпотеза Римана полягає в твердженні, що всі комплексні нулі Риманової дзета-функції, що лежать на "критичній смузі", де дійсна частина змінюється від 0 до 1, насправді лежать на "критичній лінії", де дійсна частина дорівнює 1/2. Узагальнена гіпотеза Римана складається з того ж самого твердження для узагальнень дзета-функцій, яки називають ***L-серіями Дирехле***).

Нехай n є модулем RSA. Якщо можна знайти w таке, що n — псевдопросте, але не сильне псевдопросте по основі w, то зможемо факторизувати n. Це вірно, тому що в цьому випадку можна знайти число $u \ne 1 \ (mod \ m)$, або $u \ne -1 \ (mod \ m)$ таке, що $u^2 = 1 \ (mod \ m)$, звідки випливає, що $(u + 1, n)$ є нетривіальний множник n. Можна уникнути цього при розробці криптосистеми, якщо бути впевненим, що $p-1$ і $q-1$ не мають великого спільного дільника.

Тільки найстарший і самий повільний тест — решето Ератосфена — дійсно дає простий множник m, який свідчить, що m складене. Решето складається з перевірок подільності m на прості числа менші за $m^{1/2}$. Усі більш швидкі тести перевірки чисел на простоту звичайно виявляють тільки те, що m складене, не визначаючи множники.

Перелік літератури

1. Антонов В.Н. Проектирование АРМ для лиц, принимающих решение в интегрированных АСУ // УСиМ. - 1989. - № 3.- с.118-121.
2. Автоматизированные рабочие места / Е.А. Карагодова, В.Н. Антонов, В.Ф. Маслов // Под общ. ред. В.Н. Антонова. - К.: Техника -1989. -128 с.
3. Антонов В.Н. АРМ в интегрированных АСУ // УСиМ - 1989. - № 4. - С. 125-126.
4. Антонов В.Н., Барышников В.Н. Сети АРМ. Методические рекомендации.- Киев: КПИ, МИПК - 1989. -20 с.
5. Антонов В.Н. Автоматизированные рабочие места в экономических и обучающих системах. - К. -Об-во "Знание" Украинской ССР. - 1990 - 36 с.
6. Антонов В.Н., Коваль С.М. Сеть АРМ предприятия // Механизация и автоматизация управления. - 1990. - № 3. - с.49-52.
7. Антонов В.Н. Опыт применения автоматизированных рабочих мест в системе материально-технического обеспечения.- К.: УкрНИИНТИ Госплана УССР, 1990. - 64 с. -(Новое в науке, технике и пр-ве: Обзор. информ. Сер. Новые информ. технологии, системы и средства автоматизир. упр.; Вып.5).
8. Антонов В.Н. Автоматизированные рабочие места в системах обработки данных . -К.: УкрНИИНТИ Госплана УССР, 1990.- 40 с. -(Новое в науке, технике и пр-ве: Обзор. информ. Сер. Новые информ. технологии, системы и средства автоматизир. упр.; Вып. 7).
9. Антонов В.Н. Локальная сеть АРМ предприятия // УСиМ. - 1991. - № 1. - С. 114-118.
10. Антонов В.Н. Автоматизированные рабочие места: Вопросы практического использования. - К.: Лыбидь. - 1992. - 164 с.
11. Антонов В.Н. Справочник по автоматизированным рабочим местам. / Киев. ун-т. - Киев, 1993. - 370 с.: ил. и библ. Деп. в ГНТБ Украины 06.12.93. № 2407 - Ук 93.
12. Антонов В.Н. Интеллектуально-экспертная система обработки документов, подсказки принятия решений: метод построения и реализации // УСиМ. - 1995. - № 3. - С. 82-85.
13. Антонов В.Н. Интеллектуально-экспертная графическая система // Інформатизація та нові технології. - 1996. - № 2 - С. 17-18.
14. Антонов В.Н. Архитектура интелектуально-экспертной системы поддержки и принятия решений // Информация и новые технологии - 1996. - № 4. с. 16-18.
15. Антонов В.Н. Проектирование объектно-ориентированых интеллектуальных АРМ // УСиМ 1997. - № 4/5. - с.102-106.
16. Антонов В.М. Сучасні комп'ютерні технології. Методичні рекомендації. - К.: РВЦ "Київський університет." - 1997. - 34 с.
17. Антонов В.М. Internet - всесвітня мережа. Методичні рекомендації. - К.: РВЦ "Київський університет." - 1997. - 44 с.
18. Антонов В.М. АРМ економіста, фінансиста, менеджера. Навчальний посібник.-К.: Таксон. - 1998. – 122 с.
19. Антонов В.М. Концептуальні основи роботи мережі. Методичні рекомендації. - К.: РВЦ "Київський університет." - 1998. - 30 с.
20. Антонов В.М. Системи підтримки прийняття рішень. Методичні вказівки. - К.: РВЦ "Київський університет" - 1998. - 61 с.

21. Антонов В.М., Москаленко М.А. Автоматизовані банківські системи в Internet. Методична розробка. - К.: ВЦ "Київський університет" - 1999. – 69 с.
22. Антонов В.М. Інтелектуальні АРМ: Навчальний посібник. - К.: ВЦ "Київський університет" - 2000. – 158 с.
23. Антонов В.М. Військові ситуаційно-дорадчі системи з "розмитою" логікою: концептуальний підхід // Вісник Київського ун – ту: Військово-спеціальні науки. – вип. 5. – 2002. - с. 52-57.
24. Антонов В.М., Бондарчук Ю.В. Інформаційні технології в банківській системі та електронній комерції.: Навч.пос. – К.: ВПЦ "Київський ун – т." – 2002. – 69 с.
25. Антонов В.М., Бондарчук Ю.В Методи забезпечення безпеки е-комерції та банківських систем // http: // www.vant.unicyb.kiev.ua/~vant/ua/i-Bank/i_Bank_4 - 2003
26. Антонов В.М., Бондарчук Ю.В Словник термінів, літературні та електронні джерела інформації з питань інформаційних банківських систем та е-комерції // http: // www.vant.unicyb.kiev.ua/~vant/ua/i-Bank/i_Bank_5 - 2003
27. Бовбель Е.И., Паршин В.В. Нейронные сети в системах автоматического распознавания речи // Заруб. Радиоэлектроника. – 1998. - № 4. – С. 49 -65.
28. В. Беляєва "Безпека в розподільчих системах. Відкриті системи", Москва, 1995, № 3, 36-40 с.
29. Ведєєв Д. "Захист даних в комп'ютерних мережах. Відкриті Системи", Москва, 1995, № 3, 12-18 с.
30. Воройский Ф.С. Информатика. Новый систематизированный толковый словарь-справочник. – М.: Либерия – 2001 – 536 с.
31. Векслер (J. Wexler) "Нарешті надійно забезпечений захист даних в радіомережах", Компьютеруорлд Москва, 1994, № 17.- 13-14 с.
32. Гусев В.С. Освоение Интернет: Краткое руководство. – К.: Диалектика - 2004. -288 с.
33. Гусев В.С. Поиск в Интернет: Самоучитель. К.: Диалектика–2004 -336с.
34. Гринь В. Локальные сети, модемыЮ Интернет. – К.: 2004. – 351 с.
35. Джейн А.К., Мао Ж., Моиуддин К.М. Введение в искусственные нейронные сети // Открытые системы. – 1997.- С.16 – 24.
36. Дилип Н."Стандарти и протоколы Интернет", М.:Русская радакция, 1999.- 453 с.
37. Кааба М. My SQL и Perl: Коммерческие приложения для Интернета + СД-РОМ. – СПб: Питер – 2002. – 658 с.
38. Карпов В.Е., Коньков К.А. Основы ОС: курс лекций. –М. Интернет-университет ИТ. – 2004. -632 с.
39. Комашинский В.И., Смирнов Д.А. Нейронные сети и их применение в СУ и связи. – М.: Гор. Линия.- Телеком.-2003.- 94 с.
40. Савилл Дж. Windows XP / 2000 (Вопросы и ответы) – М.: Вильямс – 2004 - 1120 с.
41. 42. С.В. Сухова "Система безпеки NetWare. Мережі", 1997, № 4, 60-70 с.
42. Толковый словарь сетевых терминов и аббревиатур – М. Cisco Sys – 2004
43. Уоссерман Ф. Нейрокомпьютерная техника. – М.: Мир. – 1990. – 235 с.
44. Хонникат Дж. Использование Интернет. –М.: Вильямс – 2004. -456 с.

Додаток А

Український словник мережевих термінів

А

Абонент мережі

Пристрій, підключений до мережі.

Автономна система

Система, яка не підключена до мережі, обладнана власним диском і не потребує для роботи підтримки будь-якої іншої системи.

Агент користувача

В еталонній моделі взаємодії відкритих систем (OSI) прикладний процес, який представляє користувача (персону або організацію) в системі передавання повідомлень X.400. Створює, представляє та розсилає повідомлення від імені користувача.

Адаптер мережевий

Електронна плата (карта) для сполучення комп'ютера із середовищем передачі інформації.

Адреса підмережі

Число, що додається до основної частини IP-адреси мережі для ідентифікування підмережі. Наприклад, 186.122.37.45/12 та 186.122.37.45/13 є адресами підмереж 12 і 13 мережі з адресою 186.122.37.45.

Адреса рівня представлення даних

В еталонній моделі взаємодії відкритих систем (OSI) адреса, що використовується для локалізації прикладного компонента. Містить мережеву адресу та до трьох селекторів — для транспортного та сеансового рівнів, а також для рівня представлення даних.

Алгоритм маршрутизації

Правила прийняття рішень транзитним вузлом по вибору адреси наступного вузла при русі повідомлення (пакета) від джерела до одержувача в мережах передачі даних. Класифікація алгоритмів маршрутизації проводиться по різним ознакам:

- ступінь централізації — централізовані алгоритми (рішення про вибір маршруту приймається центром управління мережою та повідомляється всім вузлам даного маршруту), розподілені (кожний вузол автономно приймає рішення про вибір маршруту), змішані (рішення приймається у вузлах комутації, але на нього впливають рекомендації центру управління);
- ступінь обліку інформації про стан мережі для вибору маршруту — без обліку (продовження маршруту реалізується або по всіх можливих напрямках, або

по випадково обраному); з врахуванням локальної інформації (інформація про стан інших вузлів не використовується); з врахуванням глобальної інформації (використовується інформація про стан інших вузлів);

- залежність маршрутів від інтенсивності вхідних потоків
- статистичні алгоритми (маршрут, що використовується кожною парою відправник-адресат, фіксований і не залежить від коливання трафіку); адаптивні (динамічні) алгоритми (вибір маршруту залежить від динамічного стану вузла або мережі).

Наведемо деякі приклади алгоритмів маршрутизації:

- Хвильова маршрутизація — реалізується децентралізовано і баз врахування якої-небудь інформації про мережу. Пакет, який надійшов у вузол, передається по всім вихідним трактам крім того, із якого він надійшов.
- Маршрутизація з фіксованими шляхами — розподілена з використанням інформації про топологію мережі. Для кожної пари вузлів "джерело-адресат" з'єднання завжди реалізується по одному й тому ж шляху.
- Маршрутизація з альтернативними шляхами — розподілена з урахуванням локальної інформації про топологію мережі. Для кожного адресата можуть існувати кілька маршрутів.
- Локальна адаптивна маршрутизація — розподілена з урахуванням локальної інформації про стан мережі. Заснована на використанні маршрутних таблиць, в яких для кожного вузла вказано кілька варіантів вихідних трактів, впорядкованих по ступеню їх переваги.
- Глобальна адаптивна маргрутизація — розподілена або змішана з урахуванням інформації про поточний стан інших вузлів та трактів мережі.

Аплет

Відкомпільований код прикладної програми, написаної на мові Java, призначений для виконання WWW-браузером. Тому, що аплет являє собою бінарний код, одержаний браузером із WWW-серверу, теоретично в нього можуть бути закладені деякі, неприємні для користувача тіньові функції. Тому WWW-браузери, що підтримують Java, звичайно дозволяють відключити цю можливість. Фактично це означає встановлення заборони на виконання аплетів браузером. У цьому випадку браузер також не буде витрачати час на одержання коду аплетів із віддалених серверів.

Б

Байт-орієнтований протокол

Клас комунікаційних протоколів канального рівня, що використовують який-небудь символ із набору користувача для визначення меж інформації, що передається. Вони були витіснені біт-орієнованими протоколами.

Біт парності

Службовий біт, який додається до групи інформаційних бітів, щоб забезпечити парну (або непарну) сумарну кількість одиниць в групі. В типовому випадку комунікацій за допомогою моделі OSI парність є одним з параметрів, які повинні бути

узгоджені на передаючому та приймальному кінцях каналу перед початком обміну інформацією.

Блокування запису

Особливість багатозадачних операційних систем, зокрема, мережевих, яка полягає у відведенні спроби одночасного модифікації кількома задачами (користувачами) одного й того ж запису.

Буфер

Область пам'яті, яка використовується для обробки даних при передаванні. Буфери часто використовуються для компенсування різниць в швидкодії мережевих пристроїв. Пакет даних може зберігатись в буфері до тих пір, доки його не обробить більш повільний пристрій.

В

Виклик віддаленої процедури

Одна із реалізацій моделі розподілених обчислень в архітектурі клієнт-сервер. В загальному випадку виклик посилається на віддалену систему (сервер), яка виконує вказану процедуру, використовуючи отримані аргументи, та повертає результат клієнту.

Вита пара

Середовище передачі інформації із двох перекручених між собою електричних дротів, яке характеризується найбільш простим монтажем та низькою вартістю. Розрізняють екрановану виту пару (Shieled Twisted Pair — STP) та неекрановану виту пару (Unshielded Twisted Pair — UTP). Найбільше поширення в мережах в якості середовища передачі інформації отримали категорії 3, 4 та 5. Вони розрізняються між собою рядом залежних від регулярності скручення електричних характеристик, таких як допустима тактова частота, коефіцієнт затухання сигналу та ступінь придушення перехресних наводок на ближньому кінці лінії. Чим вища категорія кабелю, тим краще його електричні характеристики. Фізично кабелі категорії 3 та 5 розрізняються лише тим, що в останньому кількість скручень на одиницю довжини є постійним (6 витків на дюйм).

Віддалена реєстрація

Метод активізації процедури регістрації в мережі, що дозволяє віддаленому користувачу отримати доступ до файлів та інших мережевих служб так, якби він був підключений локально.

Віддалене управління

Таке управління роботою комп'ютера по каналу зв'язку, при якому управляючий комп'ютер переводиться в режим емуляції терміналу хоста.

Віддалений доступ

Доступ до ресурсів мейнфрейму або локальної мережі по телефонним каналам зв'язку. Розрізняють два методи віддаленого доступу. Перший, метод віддаленого клієнта (вузла), передбачає підключення комп'ютера користувача як рівноправної

робочої станції. В цьому випадку модем розглядається як повільний мережевий адаптер, і вся інформація направляється через послідовний порт ПК. Єдиною перевагою цього поїдходу є надання віддаленому комп'ютеру повного набору дисководів та можливості використовувати в прикладних задачах стандартні шляхи передачі файлів, програм та даних. Другий метод відомий як метод віддаленого управління, при якому віддалений комп'ютер підключається до робочої станції мережі в режимі емуляції терміналу, перетворюючи дану станцію в сервер доступу. При цьому стандартне введення/виведення реалізується на віддаленому ПК, а програми, що викликаються, завантажуються та виконуються на сервері доступу.

Віддалений зворотний виклик

Метод забезпечення безпеки, при якому у відповідь на спробу під'єднатись до віддаленого серверу виконується виклик віддаленого клієнта.

Віддалений комп'ютер

Комп'ютер, в якому для зв'язку з хост-комп'ютером та доступу в мережу виконується програма дистанційного управління (емуляції терміналу) або віддаленого вузла.

Відкрита система

В контексті комп'ютерних (обчислювальних) систем — середовище, яке складається з апаратних, програмних продуктів та технологій, розроблених у відповідності з загальнодоступними та загальноприйнятими стандартами (індустріальними або встановленими de facto). Відкрита система повинна мати (за визначенням) наступні властивості:

- переносимість (portability) — операційні системи та програми використовуються на різних комп'ютерних платформах;
- інтероперабільність (interoperability) — різні апаратні та програмні платформи здатні взаємодіяти;
- масштабованість (scalability) — програмне забезпечення коректно працює як на малих, так і на великих системах з ефективністю, яка зростає пропорційно обчислювальній потужності системи;
- доступність (availability) — програмне та апаратне забезпечення є доступним для розвитку та реструктуризації.

Віртуальна локальна мережа

Об'єднання користувачів (кінцевих станцій), під'єднаних до фізично різних сегментів LAN в логічні робочі групи за допомогою програмних засобів таким чином, що комунікації між ними стають "прозорими", ніби вони працюють в одному мережевому сегменті. Для побудови VLAN можуть бути використані будь-які характеристики кінцевих станцій: MAC-адреса, IP-адреса, адреса підмережі, тип протоколу і т. ін. Однак найбільше розповсюдження отримали три моделі, або способи конфігурації віртуальних локальних мереж:

- **За допомогою портів**. Логічний сегмент в цьому випадку створюється шляхом закріплення необхідної кількості портів в маршрутизаторах та комутаторах за конкретною VLAN. Повідомлення широкого віщання розповсюджують-

Захоплення мережі

Отримання абонентом мережі права на передачу пакету.

Збирання

Процес відновлення раніше фрагментованої IP-дейтаграми перед її передачею на транспортний рівень.

Зворотна відповідь

В архітектурі клієнт-сервер передача клієнту результатів виклику віддаленої процедури.

Зірка

Вид топології локальної мережі, в якому до одного центрального абонента (концентратора) підключаються декілька периферійних абонентів; при цьому все управління мережею і (або) передачу всієї информації в ній здійснює центральний абонент.

І

Ізохронна передача

Асинхронна (старт-стопна) передача інформації по синхронним каналам. В телефонії термін "ізохронний" означає постійну частоту квантування сигналу та має протилежне значення в порівнянні з терміном "асинхронне передавання".

Індивідуальний режим

В технології 100VG-AnyLAN режим роботи концентратора, при якому кожний порт отримує пакети, адресовані лише йому.

Інтервал

В телекомунікаціях відсутність сигналу. Еквівалентний двійковому нулю.

Інтерфейс

З'єднання між двома системами чи пристроями. В термінах маршрутизації — мережеве з'єднання. Крім того, це межа між сусідніми рівнями в мережевій моделі OSI. В телефонії — загальна межа, що визначає параметри фізичного з'єднання, характеристики сигналів та значення сигналів обміну.

К

Кабель

Дротове або волоконно-оптичне середовище для передавання даних, що оснащене захисною оболонкою.

Кадр

Пакет, одиниця інформації, що передається по мережі.

Канал

Шлях передавання інформації. Один канал може бути розділений на кілька каналів (мультиплексування), що пердаються по по одному кабелю. Термін також ви-

користовується для опису зв'язку між потужними комп'ютерами та підключеною до нього периферією.

Канальне підключення

Відноситься до підключення пристроїв безпосередньо до інформаційних каналів комп'ютера.

Кільце

Вид топології локальної мережі, в якому всі абоненти послідовно передають інформацію один одному по ланцюгу, замкнутому в кільце.

Клієнт

Абонент, який не віддає свій ресурс в мережу, але який має доступ до ресурсів мережі. Інколи клієнти називаються також робочими станціями в протилежність серверу. Цей термін також відносять до прикладного програмного забезпечення, що дозволяє користувачам одержувати інформаційні послуги від серверів, розташованих десь у мережі. Програма-клієнт установлює з'єднання із сервером за допомогою визначеного протоколу передачі даних і дозволяє користувачу одержати доступ до ресурсів цього серверу. Як приклад програми-клієнта можна навести Web-браузер.

Клієнт-сервер

Термін використовується для опису мережевих систем розподілення обробки (обчислень), в яких виконання транзакцій розподілено на дві частини: частина клієнта (front end) та сервера (back end). Обидва терміни (клієнт та сервер) можуть застосовуватись як до програмного забезпечення, так і до мережевих пристроїв.

Коаксіальний кабель

Середовище передачі інформації, кабель, що складається із внутрішнього провідника та зовнішнього металевого екрану, розділений ізоляційним матеріалом. Зараз для локальних мереж використовують два типи коаксіального кабелю: кабель, що застосовується для передавання цифрових сигналів з хвильовим опором 50 Ом; кабель для передавання аналогових сигналів та високошвидкісного передавання цифрових сигналів з хвильовим опором 75 Ом.

Колізія

Ситуація, при якій в мережу передаються декілька пакетів, що визиває спотворення інформації.

Комбінований маршрутизатор

Пристрій (комп'ютер), який є комбінацією мосту і маршрутизатора.

Комірка

Базова одиниця інформації ATM-комутації та мультиплексування. Кожна комірка складається з 5-байтного заголовку та 48 байт корисної інформації.

Комунікаційна лінія

Фізичний канал зв'язку (наприклад, дріт або телефонний канал).

Комунікаційний контролер

Вузол підобласті, що використовує NCP (NetWare Core Protocol) в технології SNA.

Комункаційний сервер

Забезпечує підключення асинхронних пристроїв до локальних та глобальних мереж, використовуючи мережеве обладнання та програмне забезпечення емуляції терміналу.

Комунікація

Передавання інформації.

Комутатор

Комутатор, який передає на інші сегменти лише ті пакети, які адресовані ним, з метою зниження навантаження на мережу.

Комутація каналів

Система комутації, в якій віддалена фізична лінія зв'язку повинна існувати між пристроями передавання та приймання під час всього сеансу зв'язку. Застосовується, як правило, в мережах телефонних компаній. Крім того, застосовуються системи, в яких в якості методу доступу до каналів використовується режим конкуренції та передавання маркеру.

Комутація пакетів

Концепція, у відповідності до якої по комунікаційній мережі передаються певні порції даних по каналам, що становлюються лише на час передавання. В мережах з комутацією пакетів довгі повідомлення розділяються на невеликі частини — пакети. Кожний з них, крім даних, містить адресну та управляючу інформацію. Це дозволяє направляти пакети отримувачу різними маршрутами і в довільному порядку. Вибір оптимального для заданого критерію маршрута реалізовується маршрутизаторами, які по суті є комп'ютерами, що здатні розпізнати адресу і вибрати відповідний канал доставки пакету. В пункті призначення пакети збираються у вірному порядку пристроєм збирання пакетів. В мережіх з комутацією пакетів між вузлом-відправником та вузлом-одержувачем не створюється реальне фізичне з'єднання (його не можна "продзвонити"), а створюються так звані віртуальні канали, емулюючі фізичний канал. Найбільш широке розповсюдження отримали мережі в стандартах X.25, Frame Relay (ретрансляція кадрів), Asynchronous Transfer Mode (режим асинхронної передачі).

Комутація портів

Властивість модульних концентраторів ставити у відповідність програмними засобами один або кілька портів з різними сегментами мережі, якщо кожний сегмент представлений власним модулем. Фактично комутація портів полягає в статичному перепризначенні портів сегментам мережі.

Конвергенція

Здатність групи пристроїв, що забезпечують міжмережеву взаємодію, визначати топологію мережі після її змінення, використовуючи певний протокол маршрутизації.

Конвертер протоколу

Пристрій чи програма, що перетворює коди даних, які передаються, та/або протокол однієї мережі або пристрою у відповідні коди та/або протокол іншої мережі або пристрою, не сумісних за цими характеристиками.

Конкуренція

Метод доступу, при якому мережеві пристрої змагаються за право доступу до фізичного середовища передавання на базі певних правил, прийнятих у мережевій технології, яка використовується.

Консоль

Термінальне обладнання передавання даних, за допомогою якого команди передаються на хост-комп'ютер.

Контролер кластера

Інтелектуальний пристрій, який забезпечує з'єднання з терміналами кластера для передавання даних.

Контрольна сума

Метод перевірки цілісності даних, що передаються. Контрольна сума — це ціле число, обраховане за допомогою послідовних арифметичних операцій над послідовністю байтів. Значення знову обраховується на станції приймача та порівнюється з початковим для перевірки.

Концентратор

Пристрій, який служить для об'єднання декількох сегментів єдиної мережі і не перетворює інформацію, що передається.

Конфлікт

Див. "Колізія".

Круговий арбітраж

В технології 100VG-AnyLAN алгоритм приорітетного кругового арбітражу, при якому черговий термінальний пристрій для обслуговування вибирається у відповідності з приорітетом запиту і фізичним порядком портів концентратора.

Л

Локальна мережа

Комп'ютери або інші пристрої, з'єднані лініями зв'язку для передачі інформації між ними.

М

Маркер

Унікальна комбінація бітів або пакет спеціального виду, який використовується для процедури захоплення мережі.

Маршрутизатор

Пристрій, що забезпечує трафік між двома логічно різними (тими, що мають різні логічні адреси) локальними мережами. Функціонує на третьому, мережевому рівні еталонної моделі OSI та відповідає за вибір маршруту передачі пакетів між вузлами. Вибір маршруту виконується на основі протоколів маршрутизації, які містять інформацію про топологію мережі, та алгоритмів маршрутизації, що базуються на певних критеріях вибору, відомих як "метрика маршрутизації".

Маршрутизація

Процес вибору відповідного інтерфейсу та наступного транзитного вузла при русі пакета між мережами від джерела до одержувача.

Маршрутизація від джерела

Метод маршрутизації, що використовується в мережах Token Ring, при якому інформація про маршрут між відправником та отримувачем (послідовність мостів на шляху кадра) міститься в самому кадрі. Цей маршрут визначається робочою станцією в процесі попереднього тестування, під час якого вона в пошуках потрібної станції призначення передає тестовий кадр всім вузлам мережі. Отримавши тестовий кадр, станція призначення надсилає відповідь, яка надходить на станцію-відправника кількома шляхами. Остання на основі аналізу цієї інформації визначає маршрут кадрів до станції призначення.

Маршрутизований протокол

Протокол, у відповідності з яким пакет містить інформацію, необхідну для його пересилання між пристроями, що знаходяться в логічно різних мережах. З точки зору еталонної моделі взаємодії відкритих систем (OSI), ця інформація забезпечується мережевим рівнем. Найбільш розповсюджені маршрутизовані протоколи — TCP/IP, IPX, AppleTalk, VINES і SNA. Прикладами немаршрутизованих протоколів є NetBEUI і LAT.

Мейнфрейм

Велика обчислювальна машина (наприклад, комп'ютери компаній Burroughs, Control Data, IBM, Univac та ін.), що постачається з повним оригінальним програмно-математичним забезпеченням та периферією.

Мережа-заглушка

Локальна мережа, по якій передаються дані лише між локальними хостами, навіть якщо є підключення до інших мереж.

Мережа середнього рівня

Комп'ютерна (обчислювальна) мережа, що займає другий рівень ієрархії Internet. Є транзитною мережою, що зв'язує магістральну мережу з периферійною.

Мережева карта

Електронна плата, яка з'єднує апаратуру абонента мережі і лінії зв'язку мережі.

Мережевий адаптер

Див. "Мережева карта"

Мережевий контролер

Див. "Мережева карта"

Метод доступу

Спосіб визначення, який із абонентів мережі може захопити мережу і почати передачу свого пакета.

Метропольна маршрутизація

Алгоритм маршрутизації в мережах з комірковою топологією, в якій порядок визначається вісями. При посилці повідомлення із всіх можливих вибирається маршрут, що проходить по "основних" вісях.

Міст

Пристрій (комп'ютер), який служить для об'єднання в єдину мережу декількох мереж різних типів (наприклад, Ethernet і Arcnet), а також для зниження навантаження в мережі.

Монітор порта

Програма, яка безперервно відслідковує запити на реєстрацію, доступ до файлів або принтерів. Як тільки монітор порту знаходить запит, він встановлює необхідні параметри для організації комунікаційного каналу між операційною системою та пристроєм, що запитує, а потім передає управління іншим процесам (наприклад, програмі реєстрації), що забезпечують виконання служб.

Н

Напівдуплексна передача

Режим передачі, при якому інформація може передаватись по лінії зв'язку в двох напрямках але не одночасно.

Незбалансована лінія

Лінія, в якій для передачі сигналу використовується без врахування заземленої шини один провідник (наприклад, коаксіальний кабель).

Несуча

Гармонічні коливання, характеристика яких змінюється в залежності від сигналу, який містить інформацію, що передається.

О

Одномодове оптоволокно

Тип оптоволоконного кабелю (світловоду), по якому може розповсюджуватись світловий промінь тільки однієї частоти (одна мода). Складається із волокна (серцевини), що знаходиться всередині оболонки, яка в свою чергу покрита захисним шаром. Показник опору серцевини дещо вище, ніж оболонки, тому промінь, який розповсаджується, витримує майже повне відбиття на межі двох середовищ. Як правило, діаметр одномодового волокна складає 8,5 мкм, відбиваючої оболонки —

125 мкм. В разі використання лазерних передавачів відстань між вузлами, з'єднаними цим типом кабелю, може досягати 50 км.

Одноранговa мережа

Мережа, яка складається із вузлів, кожний із яких може виконувати функції як клієнта, так і сервера, і в якій зв'язок та розділення ресурсів реалізується безпосередньо на рівні вузлів, а не за допомогою допоміжного арбітру. В одноранговій мережі всі вузли працюють під управлінням однієї (одноранговоі) операційної системи, котра і наділяє їх вказаними властивостями. Прикладами одноранговых ОС можуть слугувати LANtestic фірми Artisoft, Windows for Workgroup, Windows 95/98 і Windows NT/2000/2003 компанії Microsoft, Personal NetWare компанії Nowell.

Оптоволоконний кабель

Середовище передачі інформації, яке являє собою скляне або пластикове волокно в оболочці, по якому поширюється світовий сигнал.

П

Пакет

Одиниця інформації, яка передається по мережі. Можуть бути короткими (порядку десятків байт і навіть одиниць байт), а також довгими (порядку декількох кілобайт). Вміщують в себе дані (необов'язково), адреси і управляючі коди.

Пароль

Засіб безпеки, що застосовується для обмеження доступу до обчислювальної системи в цілому або до певної інформації. Як правило, пароль — це унікальний набір символів, який вводиться користувачем як ідентифікаційний код. Система порівнює введений код з допустимими для даного користувача і, у випадку співпадання, дозволяє доступ у відповідності з обумовленими паролем правами.

Перешкоди

Небажані шуми в комунікаційному каналі.

Підлеглий сервер

Сервер, на якому зберігається копія NIS (Network Information Service — мережева інформаційна служба), ключової інформації про системи та користувачів мережі. Такий сервер обладнаний жорстким диском та повною копією операційної системи.

Підмережа

Частина мережі (це може бути фізично незалежний сегмент), який має ту саму мережеву адресу, що і мережа, і ідентифікується номером підмережі. Підмережа також співвідноситься із мережою, як мережа з об'єднанням мереж.

В контексті моделі взаємодії відкритих систем (OSI) підмережа — термін, що означає сукупність кінцевих та проміжних систем, які з точки зору управління являють єдиний адміністративний домен та мають єдиний мережевий протокол доступу. Прикладами можуть служити корпоративні мережі X.25 або об'єднані мостами локальні мережі.

Повнодуплексна передача

Режим передачі, при якій інформація може передаватись по лінії зв'язку в двох напрямках одночасно.

Повторювач

Пристрій для відновлення та підсилення сигналів в мережі, який служить для збільшення її довжини.

Порт

Число, що ідентифікує конкретну прикладну програму Internet. Отримані з мережі дані повинні бути розділені між різними програмами, запущеними на даному комп'ютері. Номер порту, що знаходиться у переданих по мережі пакетах даних поряд з іншою службовою інформацією, саме і вказує, якій програмі повинні бути доставлені ці дані.

У даному контексті поняття "порт" не має нічого спільного з фізичними портами вводу-виводу комп'ютера. Він є логічною конструкцією, яка використовується з протоколами передачі даних для ідентифікації кінцевих програм.

Постійне підключення

Підключення комп'ютера через виділену лінію до хосту, що безпосередньо працює з Internet та має мережеву адресу.

Потоково-орієнтована служба

Тип транспортної служби, який дозволяє клієнтам посилати дані неперервним потоком. Потоково-орієнтована транспортна служба гарантує, що всі дані будуть доставлені отримувачеві в тому ж порядку, в якому вони відправлені без дублювання.

Пошта

Здатність комп'ютерної системи посилати та отримувати повідомлення.

Поштова скринька

Область дискового простору, в якій зберігаються повідомлення електронної пошти, адресовані конкретному мережевому користувачу.

Поштовий клієнт

Система, яка не забезпечує підкачку пошти для своїх користувачів (підкачка пошти виконується на поштових серверах).

Поштовий міст

Поштовий шлюз, який розподіляє електронну пошту між двома чи більше мережами за умови, що повідомлення задовольняє певним адміністративним умовам. Взагалі, поштовий міст є спеціалізованим поштовим шлюзом, який реалізує ту чи іншу адміністративну політику по відношенню до електронних повідомлень.

Поштовий сервер

Система, в якій зберігаються поштові скринькі. Поштовий сервер може також виконувати функції поштового хосту.

Поштовий хост

Одна з основних компонент поштової системи в мережі, яка приймає та розподіляє пошту за межі мережі або домену. Може бути також поштовим сервером.

Поштовий шлюз

Машина, яка зв'язує дві та більше систем електронної пошти (в тому числі і різнорідні) та передає повідомлення між ними. Шлюзи можна розподілити на дві категорії: асинхронні одноформатні та ті, що конвертують формат. Одноформатні шлюзи дозволяють системі електронної пошти обмінюватись повідомленнями з іншою LAN, на якій встановлена така сама програма електронної пошти. Вони можуть зв'язуватись майже через будь-яке середовище передачі, включаючи виділені лінії та ISDN. Але коли говорять про шлюзи електроннної пошти, частіше за все мають на увазі шлюзи формату, що конвертуються. Основна робота шлюзів цього типу полягає в перекладі повідомлень з одного формату і інший, що дозволяє програмі електронної пошти обмінюватись повідомленнями з системами інших типів або напряму, або через проміжні стандартні формати (наприклад, MHS). Ці шлюзи звичайно називають по імені формату, в який вони перетворюють повідомлення. В принципі, будь-яка система електронної пошти може мати окремі шлюзи з кожною з інших систем.

Приватна мережа

Ліцензована приватна компанія, яка надає лінії зв'язку та комунікаційні послуги користувачам.

Пріоритет виклику

Пріоритет, призначений кожному визначеному порту в системах з комутацією каналів. Приорітет визначає порядок комутації викликів, а також те, які виклики можна допустити при резервуванні полоси пропускання, а які ні.

Проміжна буферизація

Метод передачі даних в мережах з комутацією повідомлень (або пакетів), при якій повідомлення (пакет) повністю приймається транзитним вузлом, перш ніж відправити його наступному.

Пропускна спроможність

Місткість інформаційного каналу або середовища передачі. Здебільшого вимірюють в Мбіт/с або МГц.

Протокол

Сукупність правил, відповідно до яких комп'ютери взаємодіють між собою. Зона дії протоколів простирається від регламентації процесу передачі даних по фізичному середовищу до визначення форматів повідомлень електронної пошти. Використання стандартних протоколів дозволяє взаємодіяти між собою комп'ютерам, виготовленим різними виробниками устаткування. На цих комп'ютерах можуть бути встановлені будь-які операційні системи і програмне забезпечення, єдиною вимогою до якого, з метою забезпечення можливості роботи з ресурсами Internet, є підтримка стандартних протоколів передачі даних.

Протокол маршрутизації

Протокол, що визначає метод вибору оптимального (за заданим критерієм) маршруту для різних пар "відправник-отримувач" та забезпечує правильність доставлення повідомлень їх адресатам після того, як ці маршрути обрані. Більшість протоколів маршрутизації реалізується через взаємодію маршрутизаторів. Протокол може працювати лише тоді, коли формат пакетів відповідає одному з протоколів, який маршрутизується. Найбільш розповсюдженими є протокол маршрутної інформації (RIP — Routing Information Protocol), протокол переваги найкоротшого шляху (OSPF — Open Shortest Path First), транзитна система — транзитна система (IS-IS — Intermediate System - Intermediate System).

Протокол рівня користувача

Протокол, що забезпечує виконання команд користувача та запуск прикладних програм. Прикладами можуть слугувати протоколи Telnet, FTP і SMTP.

Р

Реальний час

Режим роботи обчислювальної системи, при якому час відгуку на подію не перевищує наперед визначеної величини. В цьому режимі як правило додатковою є вимога, щоб час відгуку слабо залежав від параметрів ефективності системи, наприклад від ступеню завантаження процесора.

Репітер

Див. "Повторювач".

Реплікація

Приведення у відповідність (синхронізація) даних, що зберігаються на двох або більше комп'ютерах.

Рівень представлення даних

Шостий з семи рівнів еталонної моделі взаємодії відкритих систем (OSI) для стандартизації міжкомп'ютерних комунікацій. Визначає синтаксис інформації, що передається, тобто набір знаків, які є зрозумілими для всіх взаємодіючих відчинених систем. Сам процес встановлення загальних правил взаємодії визначається протоколом рівня представлення, за яким системи "домовляються" про форму, в якій буде передаватись інформація. Основні функції рівня: перетворення даних в потрібний формат для наступного виведення на дисплей або принтер; забезпечення управління процедурами взаємодії з транспортною мережею; селекція (вибір) прикладних процесів.

Робоча станція

Інша назва абонента мережі, клієнта мережі (в протилежність серверу) або спеціального комп'ютера, орієнтованого на роботу в мережі.

Розповсюджувач пошти

Частина поштової системи, яка забезпечує доставку повідомлень по списку поштової розсилки. Користувачі посилають повідомлення по одній адресі, а розповсю-

джувач пошти турбується про те, щоб розподілити їх по індивідуальним поштовим скринькам у відповідності до списку.

С

Сеанс

Логічне з'єднання між абонентами мережі для обміну інформацією. Включає в себе передачу декількох пакетів.

Сервер

Абонент мережі, який віддає в мережу свій ресурс і який має або не має доступу до ресурсів мережі. Також сервером називають спеціалізований комп'ютер, призначений для роботи в мережі (має швидкодіючі диски великого об'єму, швидкий процесор, велику пам'ять). Крім того, сервером називають програмне забезпечення, що дозволяє комп'ютеру, на якому воно встановлено, давати інформаційні послуги користувачам мережі. Для одержання інформації із серверу звичайно використовуються спеціальні програми-клієнти (див. "Клієнт").

Середовище передачі інформації

Електричний кабель (коаксіальний, вита пара), волоконно-оптичний кабель, радіоканал, інфрачервоний канал, — тобто те, що використовується в даній мережі для зв'язку абонентів. Характеризується вартістю, зручністю підключення, пропускною спроможністю (тобто граничною швидкістю передачі), граничною довжиною лінії зв'язку (затуханням сигналу з відстанню на даній частоті), завадостійкістю, секретністю (можливістю підслуховування) даних, що передаються, складністю адаптерів абонентів, а також рядом специфічних параметрів.

Симплексна передача

Передача даних по фізичному з'єднанню лише в одному раніше вибраному напрямку.

Скрізьний канал

Метафора, за допомогою якої описується спосіб проходження даних по каналу зв'язку, який комутується або орендується. При такому з'єднанні дані йдуть неперервним потоком: вони не формуються в пакети та пересилаються по одному й тому ж маршруту.

Список поштової розсилки

Список e-mail адрес, що використовуються розповсюджувачем пошти для перенаправлення повідомлень групам адресатів. Як правило, список розсилки використовується для організації телеконференцій — кожний список зв'язаний з певною темою.

Спулер друку

Програма, що перехоплює виведення на друк та посилає його на диск або в пам'ять, де завдання на друк зберігається до тих пір, доки не звільниться принтер.

Статистична маршрутизація

Метод маршрутизації в мережах з комутацією пакетів, при якому дані передаються по заздалегідь визначеному шляху і затримуються, якщо цей шлях блоковано.

При цьому методі маршрутна інформація як правило вводиться в маршрутні таблиці вручну.

Стек

Група мережевих пристроїв (наприклад, комутатори), які логічно інтегровані в єдину систему.

Стек протоколів

Розділений на рівні (шари) набір протоколів, які працюють сумісно, реалізуючи певну комунікаційну архітектуру. Всі сучасні комунікаційні протоколи є насправді стеками протоколів: так, родина протоколів TCP/IP скаладається з чотирьох рівнів, а IPX — з п'яти.

Стискання

Обробка масиву даних за допомогою алгоритму ущільнення, що зменшує об'єм масиву інформації, а отже, і полосу пропускання.

Сумісність

Здатність комп'ютерного обладнання різних виробників вдало взаємодіяти в одній мережі.

Сценарій

Командний текстовий файл, призначений для виконання під керуванням програми-інтерпретатора. Крім того, сценаріями часто називають програми, що взаємодіють із Web-серверами по протоколу CGI і виконують різні службові функції. Такі програми можуть бути написані на мові Perl і дійсно бути сценаріями, але також можуть бути написані і на мові C и називатися сценаріями в силу звички, сформованої завдяки широкому поширенню англійського слова "script".

T

Термінатор

Узгоджуючий пристрій, який виконує електричне погодження кабелю на обох його кінцях. Являє собою резистор з опором, який дорівнює хвильовому опору застосованого кабелю. Приєднується до кабелю за допомогою роз'єму.

Тип повідомлення

Асоційована з повідомленням інформація, що ідентифікує його вміст. Більшість систем передачі повідомлень автоматично передають отримувачу інформацію про відправника. Часто зустрічається вимога, щоб відправник вказував тип повідомлення, надаючи тим самим можливість отримувачу їх фільтрувати.

Топологія

Метод поєднання, структура зв'язків абонентів мережі. Основні топології — це "зірка", "шина" і "кільце" рідше зустрічаються топології "ланцюжок" і "дерево". Топології різняться довжиною з'єднувального кабелю, зручністю з'єднання, можливістю підключення додаткових абонентів, відмовостійкістю, можливостями управління обміном тощо.

Трансівер

Прийомопередавач мережі, який служить для збільшення потужності сигналів або для перетворення фізичної природи сигналів (наприклад, електричних сигналів в світлові і навпаки).

"Троянський кінь"

Комп'ютерна програма, яка містить засоби, що забезпечують доступ до виконавчої системи.

Тунелювання

Інкапсуляція протоколу А в протокол В таким чином, що В слугує для А протоколом канального рівня. Використовується при передачі пакетів між двома мережами за допомогою третьої транспортної, яка не підтримує протоколів цих мереж.

У

Управління конфігурацією

Одна з п'яти категорій мережевого управління. Підсистема управління конфігурацією відповідає за пошук мережевих пристроїв та визначення станів мережі.

Ф

Фізичне середовище

В контексті комунікацій будь-який засіб в фізичному світі, що слугує середовищем передавання фізичних сигналів.

Фізичний рівень

Перший, найнижчий рівень в ієрархічній семирівневій еталонній моделі взаємодії відкритих систем (OSI), що визначає дисципліну міжкомп'ютерних комунікацій. Цей рівень є повністю апаратно-орієнтованим та забезпечує такі функції, як встановлення та управління фізичним каналом, реалізуючи механічні, електричні, функціональні та процедурні аспекти взаємодії з фізичними засобами передавання даних.

Х

Хост

Позначає деякий комп'ютер, підключений до Internet, який надає для користувачів деяку інформацію. Про будь-який FTP, WWW або Gopher-сервер можна сказати, що вони є хостами Internet. Також хостами Internet є комп'ютери, що вирішують різні службові задачі в мережі, наприклад, задачу маршрутизації або доставлення електронної пошти. Нарешті, хостами Internet можна назвати робочі станції або просто комп'ютери, підключені до Internet на постійній основі.

Хост-комутатор

Комп'ютер, який обмінюється даними із зовнішніми по відношенню до даної мережі або домену джерелами, використовуючи ті ж самі комунікаційні протоколи, що і у джерел.

Ц

Централізоване управління обміном

Метод управління обміном в мережі, при якому один комп'ютер або один спеціальний пристрій управляє всім обміном в мережі.

Ч

Час встановлення з'єднання

Час, який потрібен для встановлення комутованого з'єднання між DTE-пристроями (Data Transmission Equipment — обладнання для передавання даних).

Час доступу

Часовий проміжок між виникненням заявки на передачу від абонента і отриманням права на передачу.

Ш

Швидкість порту

Максимальна швидкість передавання сигналів по каналу цифрового доступу.

Шина

Вид топології локальної мережі, в якому всі абоненти паралельно підключені до лінійного відрізку кабелю, погодженого на кінцях.

Шлюз

Пристрій (комп'ютер), який служить для об'єднання мереж з цілком різними протоколами обміну.

Шлях

Канал між двома мережевими вузлами.

Додаток Б

Англійський словник мережевих термінів

1000BASE-CX

Стандарт сегменту мережі Gigabit Ethernet на екранованій витій парі.

1000BASE-T

Стандарт сегменту мережі Gigabit Ethernet на неекранованій витій парі.

4B/5B

Самосинхронізуючий код для передачі даних, що застосовується в мережі FDDI, в якому 4 біта даних перетворюються в 5 біт, що передаються в мережу.

5B6B

Самосинхронізуючий код передачі даних, який застосовується в мережі 100VG-AnyLAN, в якому 5 біт даних перетворюються на 6 біт,що передаються в мережу.

8B/10B

Код передачі даних, який буде використовуватись в мережі Gigabit Ethernet.

A

ANSI

American National Standards Institute — Національний інститут стандартів США.

Archie

Система Archie була розроблена для пошуку файлів на FTP-серверах. Принцип дії Archie досить простий: Archie-сервер опитує відомі йому FTP-сервери і створює на основі отриманих даних індексовану базу даних, у якій зберігається інформація про те, які файли можна знайти на опитаних FTP-серверах.

При одержанні запиту від користувача системи Archie у цій базі даних робиться пошук по зазначеним користувачем ключовим словам. Ключем пошуку в базі даних системи Archie є маска імені файлу. Сервер Archie у відповідь на запит повертає список імен хост-комп'ютерів, на яких були знайдені відповідні файли (якщо такі файли удалося виявити), із вказівкою точних шляхів до них. Таким чином, після цього можна за допомогою FTP-клієнта установити з'єднання з відповідним FTP-сервером і скопіювати необхідні файли на свій локальний комп'ютер.

Arcnet

Attached Resourse Computer Net — локальна мережа, розроблена фірмою Datapoint Corporation (швидкість передачі — 2,5 Мбіт/с, метод доступу — маркерний).

ATM

Asynchronous Transfer Mode — технологія передачі інформації, при якій по мережі одночасно передаються дані, аудио- та відео-сигнали, а також відповідні технічні засоби одноіменної мережі, які забезпечують обмін інформації.

AUI

Тип роз'єму кабелю для підключення мережевого адаптеру Ethernet до трансивера (MAU) "товстого" коаксіального кабелю.

Auto-Negotiation

Протокол автодіалогу для автоматичного визначення можливостей абонента в мережі Fast Ethernet.

B

Backbone network

Стрижнева мережа — лінія зв'язку або апаратура з високою пропускною спроможністю, яка з'єднує окремі частини єдиної локальної мережі.

Bandwidth

Пропускна спроможність (місткість) інформаційного каналу або середовища передачі. Здебільшого виміряють в Мбіт/с або МГц.

BFOC/2.5

Стандарт оптоволоконного байонетного роз'єму.

BNC

Bayonet Neill Concelnan — роз'єм байонетного типу, який застосовується в мережі Ethernet для з'єднання адаптера з "тонким" коаксіальним кабелем.

Broadcast

Повідомлення широкого "віщання", яке розсилається всім адресатам в мережі.

Broadcast address

Адреса широкого "віщання" — адреса, зарезервована для одночасного розсилання повідомлень всім станціям в мережі.

Broadcast search

Пошук широкого "віщання" — розповсюдженя пошукових запитів по всім мережевим вузлам, якщо місце розташування ресурсу невідомо стороні, яка робить запит. Див. також "Directed search".

Broadcast storm

"Шторм" широкого "віщання" — небажана подія в мережі, при якій багато повідомлень широкого віщання розсилаються одночасно, суттєво знижуючи пропускну спроможність мережі та викликаючи затримки в передаванні інформації.

BSC

Binary Synchronuos Communication — символьно-орієнтований канальний протокол для напівдуплексних застосувань. Для срощення часто називається bisync.

Bus and tag channel

Оригінальна розробка IBM, виконана в 60-х роках. Канал використовує шинну технологію. Максимальна швидкість передавання — 4,5 Мбіт/с на відстань 125 м. Див. також "Parallel channel".

Bypass mode

Обхідний режим — режим роботи в мережах FDDI та Token Ring, при якому деякі пристрої підключені паралельно мережевому кільцю.

Byte reversal

Процес зберігання цифрових даних, при якому найменш важливі байти записуються першими. Використовується для цілих чисел та адрес у пристроях з мікропроцесорами Intel.

C

Catenet

Сукупність мереж, зв'язок між якими реалізується за допомогою маршрутизаторів та хост-комп'ютерів. Internet є найбільш яскравим прикладом мережі catenet.

CATV

Cable Television — кабельне телебачення (перед цим називалось Community Antenna Television) — система передавання інформації, в якій багато TV-каналів або програмних матеріалів передається користувачам за допомогою широкополосного коаксіального кабелю.

CBDS

Connectionless Broadband Dada Service. Синонім — SMDS (Switched Megabit Data Service). Високошвидкісна WAN-технологія з комутацією пакетів; що базується на методі використання дейтаграм.

CCITT

Consultative Commitee for International Telegraph and Telephone — міжнародна організація, що розробила комунікаційні стандарти, такі як Recommendation X.25

CCS

Common Challing Signalling — система передавання сигналів, що використовується в багатьох телефонних мережах, яка відокремлює сигнальну інформацію від даних користувача.

Cell

Комірка — базова одиниця інформації ATM-комутації та мультиплексування. Кожна комірка складається з 5-байтного заголовку та 48 байт корисної інформації.

Cell relay

Комутація комірок — мережева технологія, заснована на використанні комірок або невеликіх пакетів фіксованої довжини. Комірки містять ідентифікатори, які визначають потік даних. Оскільки комірки мають фіксовану довжину, їх обробка та комутація відбувається з великою швидкістю. Комутація комірок є основною для бага-

тьох високошвидкісних мережевих протоколів, наприклад, IEEE 802.6, DQDB (Distributed Queue Dual Bus), SMDS Interface Protocol і ATM.

CERFnet

California Education and Research Foundation Network — TCP/IP-мережа у Південній Каліфорнії, яка з'єднує університетські центри та призначена для розповсюдження наукових знань за допомогою засобів комунікації.

CGI

Common Gateway Interface — набір угод, що використовується Web-сервером для обміну інформацією з іншими прикладними програмами, які часто називають шлюзами. Шлюзи використовуються системою WWW для доступу до тієї інформації, що не може оброблятись безпосередньо Web-сервером і, у більшості випадків, недоступна для інших видів сервісу Internet. Як приклад, можна навести інформацію з баз даних. При цьому, для кінцевого користувача, робота шлюзів "прозора", тобто продивляючись ресурси Web у своєму браузері, недосвідчений користувач навіть не помітить, що деяка інформація була подана йому за допомогою системи шлюзів.

Channel

Канал — шлях передавання інформації. Один канал може бути розділений на кілька каналів (мультиплексування), що передаються по одному кабелю. Термін також використовується для опису зв'язку між потужними комп'ютерами та підключеною до нього периферією.

CHAOSnet

Мережевий протокол, розроблений MIT (Massachusets Institute of Technology). Спочатку використовувався спеціалістами по системах штучного інтелекту.

CHAP

Challenge Handshake Authentification Protocol — функція системи безпеки, що забороняє несанкціонований доступ до пристроїв, які цю функцію використовують. CHAP використовується лише на лініях, що підтримують PPP-інкапсуляцію.

Cheapernet

Промисловий термін, що відноситься до стандарту IEEE 802.3 — 10Base2 або до кабелю, що визначається тим самим стандартом. Термін "тонкий" Ethernen, що також використовується для опису цього ж стандарту, визначає менш дорогий але більш тонкий кабель Ethernet.

Choke packed

Пакет, що надсилається пристроєм приймача пристрою передавання у випадку перевантаження каналу. Повідомлює про необхідність зменшити швидкість передавання.

CICS

Customer Information Control System — прикладна підсистема IBM, що забезпечує одночасну обробку програмами користувача транзакцій, що надходять з віддалених терміналів.

CiscoBus

Високошвидкісна шинна технологія, що забезпечує швидкість передавання даних 0,5 Гбіт/с, розроблена компанією Cisco для забезпечення високошвидкісної комутації.

CLNP/CLNS

Connectionless Network Protocol/Connectionless Network Service — протокол стандарту OSI, який не потребує підтвердження встановлення каналу зв'язку перед передаванням інформацій. CLNP — це еквівалент IP-протоколу в термінах OSI.

CMI

Coded Mark Inversion — техніка лінійного кодування CCITT, визначена для чотирирівневого мультиплексованого сигналу.

CMIP/CMIS

Common Management Information Protocol/Common Management Information Services — інтерфейс протоколу/служби управління мережню моделі OSI, розроблений для управління неоднорідними мережами.

CMNS

Connection-Mode Network Service — розширення можливостей локальної комутації мереж X.25 для різних середовищ передавання даних (Ethernet, FDDI, Token Ring).

CMOT

CMIP Over TCP — використання протоколу управління мережою (CMIP) моделі OSI за допомогою протоколів Internet (TCP/IP).

CMT

Connection Management — прийом, що управляє процесом переведення кільця мережі FDDI (Fiber Distributed Date Interface) в різні стани (off, active, connect — вимкнене, активне, приєднане і т.ін.) згідно специфікації X3T9.5.

CODEC

Coder-Decoder — кодер-декодер — пристрій, що як правило використовує імпульсно-кодову модуляцію для перетворення аналогового звукового сигналу в послідовність бітів та навпаки.

Common channel signaling

Використання певного каналу виключно для передавання управляючих сигналів.

Communication

Комунікація, передавання інформації.

Community

В протоколі SNMP — це логічна група керуємих пристроїв та NMS (Network Management Station — станція управління мережою в одному адміністративному домені).

Companding

Компандування — ущільнення (стискання) та розширення (Compressing і expansion). Частина процесу PCM (pulse-code modulation), за допомогою якого зна-

чення квантового аналогового сигналу округлюються до дискретних крокових значень по нелінійній шкалі. Потім десятковий номер кроку кодується двійковим еквівалентом для передавання. В терміналі приймання відбувається зворотний процес з використанням тієї ж нелінійної шкали.

Compression

Ущільнення (стискання) — обробка масиву даних за допомогою алгоритму, що зменшує об'єм масиву інформації, а отже, і полосу пропускання.

Configuration management

Управління конфігурацією — одна з п'яти категорій мережевого управління. Підсистема управління конфігурацією відповідає за пошук мережевих пристроїв та визначення станів мережі.

Congestion

Перевантаження, надмірний мережевий трафік. Термін використовується для опису процесу передавання даних в віртуальних мережах після встановлення віртуального з'єднання.

Connectionless

Термін використовується для опису процесу передавання даних у віртуальних мережах, але при відсутності віртуального з'єднання між пристроями.

CONP

Connection-Oriented Network Protocol — протокол мережевого рівня моделі OSI, що забезпечує передавання даних після встановлення віртуального з'єднання (connection-oriented operation) для протоколів більш високого рівня.

Contention

Конкуренція — метод доступу, при якому мережеві пристрої змагаються за право доступу до фізичного середовища передавання на базі певних правил, прийнятих у мережевій технології, яка використовується.

Convergence

Конвергенція (східність) — здатність групи пристроїв, що забезпечують міжмережеву взаємодію, визначати топологію мережі після її змінення, використовуючи певний протокол маршрутизації.

COS

Corporation for Open Systems — організація, яка визначає використання протоколів стандарту OSI за допомогою сертифікації та тестування на сумісність. Крім того, цей термін визначає клас обслуговування протоколів: яким чином протокол низького рівня повинен обробляти повідомлення протоколу більш високого рівня.

COSINE

Corporation for Open Systems Interconnection Networking in Europe — європейський проект побудови комунікаційної мережі між науковими та індустріальними об'єктами у Європі, який фінансується ЄС.

CP

Control Point — елемент, який управляє ресурсами мережевих пристроїв у мережах SNA. Він може забезпечувати обслуговування запитів від інших пристроїв.

CPE

Customer Premises Equipment — термінальне обладнання (термінали, телефони і модеми), що обслуговуються телефонною компанією. Встановлені на вузлах користувачів та підключені до телефонної мережі компанії.

CRC

Cyclic Redundancy Check — метод перевірки на наявність помилок, при якому вузол-отримувач кадра обраховує залишок від ділення вмісту кадру або простий двійковий дільник та порівнює обрахований залишок (який також часто називається CRC) із значенням, записаним в кадрі вузлом-відправником.

CREN

Corporation for REsearch snd Education Networking — результат злиття BITNET та CSNET.

Cross talk

Перехресні перешкоди — взаємні перешкоди, що наводяться одним каналом зв'язку на інший.

CSA

Canadian Standarts Association — агенство в Канаді, що сертифікує продукти по національним стандартам безпеки Канади (Canadian national safety standarts).

CSLIP

Compressed Serial Link Internet Protocol — протокол, що мінімізує трафік та прискорює передачу по SLIP-лініям.

CSMA/CD

Carrier Sense Multiple Access with Collision Detection — механізм доступу до каналів зв'язку при якому пристрої, що мають дані для передачі, попередньо перевіряють наявність несучої. Якщо несуча не знайдена протягом деякого часу, то пристрій починає передачу даних. Якщо два пристрої починають передачу даних одночасно, то виникає конфлікт, що визначається конфліктуючими пристроями, котрі припиняють передачу та відновлюють її через деякий невеликий, випадково вибраний інтервал часу. Доступ CSMA/CD використовується в технології Ethernet і IEEE 802.3.

CSNET

Computer Science Network — велика мережа, що початково з'єднувала університети, дослідницькі та комерційні організації. CSNET об'єдналась з BITNET, утворивши CREN.

CSNP

Copmlete Sequence Number PDU (protocol data unit) — протокольний блок даних, який посилається певним маршрутизатором, що працює в мережі з протоколом маршрутизації OSPF (Open Shortest Path First — "першим обирається найкоротший

шлях" — мережевий протокол маршрутизації), для забезпечення процесу синхронізації бази даних.

CSU

Channel Service Unit — цифровий інтерфейсний пристрій, що приєднує обладнання кінцевого користувача до телефонної мережі.

Cyclic Redundancy Check

Див. "CRC".

D

DIX

Об'єднання компаній DEC (Digital), Intel та Xerox, утворене для підтримки і стандартизації мереж Ethernet.

DNS

Domain Name System — розподілена база даних, що забезпечує перетворення імен комп'ютерів (наприклад, `wist.ifmo.ru`) у числові адреси Internet (наприклад, `194.85.161.24`) і навпаки. DNS дозволяє використовувати при роботі з ресурсами Internet символьні імена хостів, а не їх числові адреси, які важко запам'ятовувати.

DTE

Data Terminal Equipment — термінальне обладнання передавання даних, за допомогою якого команди передаються на хост-комп'ютер.

E

ECMA

European Computer Manufacturers Assotiation — Європейська Асоціація виробників комп'ютерів, міжнародна організація.

EIA/TIA 568

Commercial Building Telecommunicatios Cabling Standard — стандарт на кабелі із витих пар для локальних мереж, який визначає їх основні характеристики (затухання на різних частотах, відображення, кількість витків на метр довжини і т. ін.)

Ethernet

Найбільш розповсюджена в світі локальна мережа, запропонована фірмою Xerox (топологія — "шина", метод доступу — CSMA/CD, швидкість передачі — 10 Мбіт/с). Задовольняє стандарту IEEE802.3.

F

Fast Ethernet

Високошвидкісна різновидність мережі Ethernet, яка забезпечує швидкість передачі 100 Мбіт/с. Задовольняє доопрацьованому стандарту IEEE 802.3 (стандарт затверджений у 1995 році)

FCS

Frame Check Sequence — перевірочна послідовність кадру — контрольна сума (назва прийнята в мережі 100VG-AnyLAN).

FDDI

Fiber Distributed Data Interface — кільцева оптоволоконна високошвидкісна локальна мережа (метод доступу — маркерний, швидкість передачі — 100 Мбіт/с).

FLP

Fast Link Pulse — сигнали, що передаються у проміжках між пакетами в мережі Fast Ethernet в режимі автодіалогу.

FOIRL

Fiber Optic Inter-Repeater Link — стандарт оптоволоконного зв'язку між двома репітерами мережі Ethernet.

FOMAU

Fiber Optic MAU — оптоволоконні трансивери мережі Ethernet.

FTP

File Transfer Protocol — протокол, що регламентує процедуру передачі даних у процесі роботи з файловими архівами Internet. Також, загальновживана назва прикладної програми, що використовується для доступу до файлових архівів Internet.

G

Gigabit Ethernet

Надвисокошвидкісна версія Ethernet, яка забезпечує швидкість передачі 1 Гбіт/с.

Gopher

Вид сервісу Internet, що забезпечує доступ до його ресурсів на основі системи меню. Виявивши необхідний матеріал, можна його прочитати або одержати до нього доступ через Gopher, не турбуючись про точну адресу і місцезнаходження цього ресурсу. У якомусь відношенні, використання системи Gopher нагадує перегляд каталога бібліотеки, розташованої в іншому місті. При цьому, якщо ця бібліотека є частиною системи Gopher, то її точне месцезнаходження не має ніякого значення.

H

HTML

HyperText Markup Language — гіпертекстова мова міток, спеціально розроблена для створення документів системи WWW. HTML-документи являють собою звичайні текстові файли, за винятком того, що деякі рядки в них є керуючими. Ці рядки називаються тегами і визначають зовнішній вигляд документа при перегляді його в Web-браузері. Саме теги мови HTML дозволяють авторам HTML-документів включати в них ілюстрації, гіпертекстові посилання на інші ресурси Internet, а також усе те, що можна зустріти у вікні Web-браузера.

HTTP

HyperText Transfer Protocol — протокол передачі гіпертексту — регламентує процес передачі даних у системі WWW. Протокол HTTP заснований на системі запитів і відповідей. Запити відправляються клієнтом, у ролі якого звичайно виступає Web-браузер, WWW-серверу. Відповіді відправляються сервером клієнту і звичайно містять у собі запитані останнім ресурси, наприклад, HTML-документи, графічні файли і т. ін. По замовчуванні, при передачі даних HTTP використовує порт із номером 80. Для адресації ресурсів WWW у HTTP використовується універсальний локатор ресурсів — URL.

I

IAB

Internet Activites Board — рада по архітектурі Internet. Група дослідників в області мережевих технологій, яка регулярно досліджує питання, пов'язані з мережами. Саме ця група у головному визначає перспективні напрямки розвитку Internet.

ICMP

Internet Control Message Protocol — протокол управління повідомленнями в Internet. Internet-протокол мережевого рівня, що формує інформаційні пакети повідомлень про помилки та інші події, що відносяться до процесу обробки IP-пакетів. Описаний в документі RFC 792.

I-connector

З'єднувач двох кусків "тонкого" коаксіального кабелю, оснащених роз'ємами BNC на кінцях.

IDI

Initial Domain Identifier — ідентифікатор, що відповідає за формування значення Domain Specific Part (DSP) в мережевих адресах OSI (NSAP).

IDP

Initial Domain Part — частина адреси CLNS, що містить ідентифікатор походження і формату та ідентифікатор домену.

IDPR

Interdomain Policy Routing — експериментальний протокол міждоменної маршрутизації, який динамічно змінює алгоритми обміну між автономними системами. IDPR інкапсулює міждоменний системний трафік на направляє його по шляху передавання у відповідності з алгоритмами кожної з автономних систем. IDPR пропонується групою IETF.

IDRP

IS-IS Interdomain Routing Protocol — OSI-протокол, що визначає метод взаємодії маршрутизаторів у різних доменах.

IEEE

Institute of Electrical and Electronic Engineers — Інститут інженерів по електротехніці та електроніці. Організація, що встановлює мережеві стандарти. IEEE-стандарти для локальних мереж сьогодні є домінуючими, зокрема, це протоколи, аналогічні або еквівалентні Ethernet та Token Ring.

IEEE 802.12

Стандарт IEEE, якому задовольняє мережа 100VG-AnyLAN (швидкість передачі 100 Мбіт/с, топологія "зірка", централізоване управління доступу и т. ін.).

IEEE 802.2

IEEE-протокол, розроблений для локальних мереж. Реалізує підрівень управління логічним зв'язком канального рівня моделі OSI. IEEE 802.2 відповідає за безпомилкове передавання даних, формування пакетів, контроль потоку та інтерфейс із мережевим рівнем. Застосовується в локальних мережах, що використовують протоколи IEEE 802.3 та IEEE 802.5.

IEEE 802.3

IEEE-протокол, розробленний для локальний мереж. Визначає реалізацію фізичного рівня та підрівня MAC канального рівня моделі OSI. IEEE 802.3 використовує метод доступу CSMA/CD для різних швидкостей передавання даних в різних мережевих середовищах. Одна із фізичних реалізацій IEEE 802.3 — 10Base5 — практично ідентична Ethernet.

IEEE 802.3z

Стандарт IEEE, якому задовольняє мережа Gigabit Ethernet.

IEEE 802.4

IEEE-протокол, розроблений для локальних мереж. Визначає реалізацію фізичного рівня та підрівня MAC канального рівня моделі OSI. IEEE 802.4 використовує шинну технологію з передаванням маркеру.

IEEE 802.5

IEEE-протокол, розроблений для локальних мереж. Визначає реалізацію фізичного рівня та підрівня MAC канального рівня моделі OSI. IEEE 802.5 використовує кільцеву топологію на екранованій витій парі, метод доступу з передаванням маркеру та працює із швидкістю передавання даних 4 або 16 Мбіт/с. IEEE 802.5 практично ідентичний протоколу IBM Token Ring.

IEEE 802.6

Специфікація IEEE для муніципальних мереж (MAN), що базується на технології DQDB (Distributed Queue Dual Bus). IEEE 802.6 підтримує комутацію пакетів та каналів із швидкістю передавання даних від 1,5 Мбіт/с до 155 Мбіт/с.

IETF

Internet Engineering Task Force — робоча група інженерів Internet, що складається з більш ніж 40 комісій, відповідальних за розв'язок поточних проблем Internet.

IFIP

International Federation for Information Processing — Міжнародна федерація по обробці інформації. Дослідницька організація, що виконує попередні роботи по OSI-стандартизації. Серед робіт, які виконала IFIP, — формалізація базової моделі MHS (Message Handling System — система обробки повідомлень).

IGP

Interior Gateway Protocol — Internet-протокол, який використовується для обміну таблицями маршрутизації в межах однієї автономної системи. Прикладами Internet IGP-протоколів є IGRP, RIP та OSPF.

IGRP

Interior Gateway Routing Protocol — протокол внутрішньої маршрутизації між шлюзами. Розроблений компанією Cisco Systems для розв'язання проблем маршрутизації великих неоднорідних мережах.

IIH

IS-IS Hello — повідомлення, яке розсилається всім IS-IS системам для підтримки зв'язку із сусідніми системами.

ILMI

Interim Local Managenment Interface — проміжний інтерфейс локального рівня. Специфікація ATM Forum для забезпечення функцій управління об'єднаними мережами за допомогою інтерфейсу ATM UNI (User-to-Network Interface — інтерфейс "абонент-мережа").

IMP

Interface Message Processor — інтерфейсний процесор повідомлень. Раніше використовувався лише для комутаторів пакетів в мережі Internet. Зараз абревіатура IMP використовується для вузлів комутації пакетів (Packed-Switched Node), комутаторів пакетів (Packed Switch) або комутаторів (Switch).

In-band signaling

Передавача сигналів управління в полосі частот, що використовується для передачі основної інформації.

INOC

Internet Network Operations Center — група, що реалізовувала управління та контроль основних маршрутизаторів мережі Internet.

INTAP

Interoperability Technology Association for Information Processing — технічна організація, сформована для розвитку японської специфікації OSI і тестів на сумісність.

Integrated IS-IS

Протокол маршрутизації, заснований на IS-IS-протоколі маршрутизації, призначеному для мереж в архітектурі OSI та такого, що підтримує IP або інші мережі. В цьому випадку розсилається лише один набір поновлень маршрутної інформації, що робить маршрутизацію більш ефективною.

Internet

Термін, що використовується для посилання на всесвітню мережу, що об'єднує тисячі мереж у всьому світі і яка має своєрідну культуру, що побудована на простоті використання у повсякденному житті, можливості проведення різних досліджень та стандартизацій. Велика кількість нинішніх передових технологій народились в Internet-співтоваристві. Internet розвинулась з мережі ARPANET.

Internet address

Internet-адреса (або IP-адреса) — 32-бітна адреса, призначена хост-комп'ютерам, що використовують родину протоколів TCP/IP. Адреса записується у вигляді 4-8-бітових чисел, розділених крапками і складається з мережевої секції, необов'язкової секції підмережі та хост-секції.

Internetwork

Зв'язана мережа — кілька мереж, зв'язаних за допомогою маршрутизаторів та (в загальному випадку) функціонуючих як одна мережа. Іноді використовується назва internet, яке не слід змішувати з Internet.

Internetworking

Організація міжмережеової взаємодії. Узагальнений термін, який використовується для посилання на галузь промисловості, пов'язану з проблемами об'єднання мереж. Термін може відноситись до продуктів, методів та технологій.

Intra-area routing

Термін, що використовується для визначення маршрутизації в межах однієї області в мережах DECnet.

IP

Internet Protocol — протокол мережевого рівня, який містить адресну та деяку управляючу інформацію, що дозволяє виконувати маршрутизацію пакетів. Описаний в документі RFC791.

IP adress

Див. "Internet address".

IPSO

IP Security Option — частина Internet-протоколу (IP), що визначає рівні безпеки для різних типів інтерфейсів.

IPX

Internetwork Packed Exchange — протокол мережевого рівня фірми Nowell, подібний XNC та IP і використовується в мережах NetWare.

IRDP

ICMP Router Discovery Protocol — протокол, що дозволяє хост-комп'ютеру визначити адресу маршрутизатора, який може використовуватись ним як шлюз по замовчуванню. Аналогічний протоколу ES-IC, але використовується в IP-мережах. ICMP (Internet Control Message Protocol — протокол управління повідомленнями в мережі Internet). Протоколи ES-IC (End System – Intermediate System — кінцева сис-

тема – проміжна система) — дозволяють кінцевим та проміжним системам розпізнавати одна одну.

IRTF

Internet Research Task Force — група дослідників в області технологій об'єднаних мереж. IRTF управляється організацією Internet Research Steering Group (IRSG).

IS

Intermediate System — проміжна система — вузол маршрутизатора в OSI-мережах.

Isarithmic flow control

Метод управління потоком, при якому по мережі передаються спеціальні пакети-дозволи. Отримання такого пакету дає право на передавання даних.

ISDN

Integrated Services Digital Network — цифрова мережа з наданням комплексних послуг. Комунікаційний протокол, запропонований телефонними компаніями, який дозволяє передавати по телефонним мережам дані, голос та іншу інформацію. Див. також "BRI", "BISDN" та "PRI".

IS-IS

Intermediate System-to-Intermediate System (проміжна система – проміжна система) — протокол маршрутизації, що працює на основі інформації про стан каналів зв'язку та, який забезпечує маршрутизацію лише між різними проміжними системами. Протокол IS-IS був розроблений ISO спеціально для мереж в архітектурі OSI. Алгоритм маршрутизації IS-IS передбачає, що проміжні системи (маршрутизатори) обмінюються інформацією на базі єдиної метрики мережі для визначення її топології.

ISO

International Organization for Standartization — часто невірно розшифровується як International Standarts Organization. Міжнародна організація, яка відповідає за стандарти в різних галузях, в тому числі і в області мережевих технологій. ISO розробила найпопулярнішу мережеву еталонну модель — OSI-модель.

ISODE

ISO Development Envorinment — популярна реалізацію верхніх рівнів OSI-моделі в стеку TCP/IP-протоколів.

ISSI

Inter-Switching System Interface — стандартний інтерфейс між SMDS-комутаторами. SMDS (Switched Multimegabit Data Service) — технологія побудови високошвидкісних глобальних мереж з пакетною комутацією, що базується на використанні дейтаграм.

J

Java

Мова програмування, розроблена компанією Sun Microsystems. В даний час широко використовується в системі WWW.

JavaScript

Інтерпретована мова програмування, що дозволяє писати програми, запуск яких здійснюється при завантаженні і перегляді HTML-документів Web-браузером. Команди JavaScript вбудовуються безпосередньо в HTML-сторінки і дозволяють обробляти різні дії користувачів при роботі з документом: наприклад, "клікання" мишею на одному з об'єктів сторінки, позиціонування курсору на гіперпосиланні або ж коректність введення інформації в поля HTML-форми.

L

LAN

Local Area Network — локальна обчислювальна мережа, ЛОМ.

LLC

Logical Link Control — верхній підрівень другого рівня моделі OSI (рівня управління лінією передачі), який відповідає за управління логічними зв'язками.

M

MAC

Media Access Control — управління доступом до середовища — один з підрівнів канального рівня еталонної моделі взаємодії відчинених систем (Open System Inconnect — OSI), який визначає формат інформації, що передається по мережі, адресацію та спосіб яким мережевий пристрій отримує доступ до розділеного середовища передачі (наприклад, протокол CSMA/CD є частиною підрівня MAC). Управління доступом до середовища залежить від фізичного середовища передачі.

MAC address

MAC-адреса — в мережах з розділеним середовищем передачі (Ethernet, Token Ring, FDDI і т. ін.) фізична адреса, що ідентифікує мережевий вузол і тому унікальна. Як правило, розміщується на мережевому контролері та жорстко встановлюється виробником, хоча звичайно може бути змінена за допомогою відповідної програми (робити це категорично не рекомендується). Кожний виробник мережевих контролерів має деякий діапазон MAC-адрес, при цьому перші три байти визначають самого виробника. Всі пристрої, що підключені до мережі, аналізують MAC-адресу, яка міститься на другомі підрівні (Media Access Control Syblayer) канального рівня, та декодують пакет у випадку співпадання адрес.

MacBinary communications

Передавання у форматі MACBinary — спеціальний формат для зберігання на інших комп'ютерах програм, розроблених для ПК Macintosh. Для передачі на інші комп'ютери додаток чи документ перетворюється в спеціальний текстовий формат. При надходженні файлу на MAC-комп'ютер виконується обернене перетворення. Більшість комунікаційних програм роблять це автоматично.

Mail

Пошта — здатність комп'ютерної системи посилати та отримувати повідомлення.

Mailbox

Поштова скринька — область дискового простору, в якій зберігаються повідомлення електронної пошти, адресовані конкретному мережевому користувачу.

Mailer

Протокол, який специфікує стратегію та механізми, що використовуються програмою розсилання пошти.

Mailing list

Список поштової розсилки — список e-mail адрес, що використовуються розповсюджувачем пошти для перенаправлення повідомлень групам адресатів. Як правило, список розсилки використовується для організації телеконференцій — кожний список зв'язаний з певною темою.

Mainframe

Головний комп'ютер (мейнфрейм) — велика обчислювальна машина (наприклад, комп'ютери компаній Burroughs, Control Data, IBM, Univac та ін.), що постачається з повним оригінальним програмно-математичним забезпеченням та периферією.

MAN

Metropolitan Area Network — мережа масштабу міста — комп'ютерна (обчислювальна) мережа, призначена для обслуговування визначеної географічної області, яка приблизно дорівнює площі великого міста. Як середовище передачі часто використовують оптоволокно, яке прокладається в тунелях метро.

Manchester-II

Самосинхронізуючий дворівневий код передачі, який застосовується, зокрема, в мережі Ethernet.

Marshall

Програма, що виконує ущільнення значень кількох змінних, масивів чи структур в єдиний неперервний блок пам'яті. В більшості систем передачі повідомлень дані перед посиланням підлягають ущільненню.

Master server

Мастер-сервер — сервер, який містить мастер-копію бази даних мережевих інформаційних послуг.

MAU

Media Attachment Unit — пристрій підключення до середовища — в термінах стандарту IEEE 802.3 пристрій, за допомогою якого робоча станція підключається до середовища передачі. Популярна назва — трансівер (Transceiver — TRANSmitter/reCEIVER). Може бути виконано у вигляді окремого блоку і монтуватись на кабель ("товстий" Ethernet) або вбудовуватись в плату мережевого адаптера ("тонкий" Ethernet).

Mbps

Mb/s, Mbits per second — мегабіт в секунду (Мбіт/с) — одиниця виміру швидкості передачі і пропускної спроможності середовища передачі.

Mesh

Вузлова мережа — топологія, в якій вузли утворюють регулярну ациклічну D-мірну решітку, при цьому кожне ребро решітки паралельно її вісі і з'єднує два суміжних вузли вздовж цієї вісі. Архітектура багатьох суперкомп'ютерів реалізується як двох- або тривимірна решітка.

MHS

Message Handling System — система обробки повідомлень — система агентів повідомлень користувача, агентів передачі повідомлень, зберігання повідомлень та пристроїв доступу, що забезпечує роботу електронної пошти в стандарті OSI.

MIB

Management Information Base — база управляючої інформації — в системі управління мережею у відповідності до протоколу SNMP (Simple Network Management Protocol) шаблон, де перераховані та об'єднані в групи всі об'єкти, що підлягають управлінню. Протокол SNMP визначає 11 груп об'єктів для MIB-2. Ось деякі з них. Група статистики включає лічильники кадрів, повідомлень широкого розповсюдження та багатоадресних повідомлень, а також конфліктів та помилкових кадрів і містить дані, що відображують використання мережі на рівні трафіку. В групі аварійних сигналів користувач може визначати пороги подій. Подія генерується в той момент, коли виконується умова виходу за порогове значення. В групі, що містить таблицю хостів, збирається статистика трафіку через вузли. За допомогою групи фільтрів визначаються умови утотожнення пакетів (наприклад, оголошуються помилковими всі короткі пакети). Група подій веде журнал обліку подій. Існує також група історії аналізу тенденцій в показниках, що отримуються групою статистики.

MII

Media Independent Interface — інтерфейс, що не залежить від середовища, — мініатюрний 40-контактний конектор, що забезпечує електричний інтерфейс між деякими системами Sun та трансіверами мережі Ethernet стандартів 10Base-t та 100Base-t.

MIME

Multipurpose Internet Mail Extension — формат MIME був розроблений з метою забезпечення можливості пересилки по електронній пошті Internet будь-яких типів даних. Розробники MIME зрозуміли, що SMTP у більшості випадків не зможе реалізувати потреби користувачів, що зростають. Протокол SMTP навіть не може працювати з іноземними мовами, тому що перші 128 символів таблиці ASCII (7-розрядні символи) дозволяють передавати лише літери англійського алфавіту.

Стандарт MIME був розроблений у 1993 році. Вся його концепція заснована на понятті типу даних. Поточна версія MIME містить у собі сім стандартних типів, кожний із яких додатково розділений на підтипи:

- ASCII-текст;
- програми, передані в бінарній формі;

- аудіодані;
- інкапсульовані повідомлення;
- повідомлення з декількох частин;
- графічні зображення;
- відеодані.

Внаслідок того що розглянутий стандарт може використовуватись для відправлення аудіо- і відео-даних, приєднаних як доповнення до тексту і до бінарних програм, він і був названий "Багатоцільові розширення пошти Internet".

Докладний опис MIME можна знайти в стандарті RFC-1341.

MMF

Multimode Fider Optic — мультимодовий оптоволоконний кабель.

MNP

Microcom Networking Protocol — набір апаратно реалізованих протоколів захисту від помилок (MNP Levels 1-4) та техніки стискання даних (MNP Levels 5), що розроблені компанією Microcon для модемів.

MTA

Message Transfer Agent — агент передачі повідомлень — прикладний процес в еталонній моделі осіб, що забезпечує зберігання та перенаправлення повідомлень в системі обробки повідомлень стандарту X.400. Еквівалентний поштовому агенту в Internet.

MTU

Maximum Transmission Unit — максимальний блок (даних), що передається — максимальна довжина пакету даних, який може бути переданий через фізичне середовище (наприклад, MTU для Enternet складає півтори тисячі байт).

N

NE2000

Популярний тип адаптеру мережі Ethernet (компанія Novell), який став одним із фактичних стандартів.

NetBIOS

Network Basic Input/Output System — мережеве програмне забезпечення нижнього рівня, розроблене з самого початку компанією IBM і яке стало пізніше фактичним стандартом.

NIC

Network Interface Card — мережевий адаптер (контролер), мережева карта.

NLP

Normal Link Pulse — сигнали, які передаються в сегментах 10BASE-T між пакетами для контролю цілосності лінії зв'язку.

NRZ

Non-Return to Zero — найпростіший несамосинхронізуючий код передачі, який застосовується, наприклад, в інтерфейсі RS-232C.

NVP

Nominal Velocity of Propagation — швидкість поширення сигналу в кабелі — описується в долях від швидкості світла, наприклад, NVP = 0,7C.

O

Open-pipe

Скрізьний канал — метафора, за допомогою якої описується спосіб проходження даних по каналу зв'язку, що комутується або орендується. При такому з'єднанні дані йдуть неперервним потоком: вони не формуються в пакети та пересилаються по одному й тому ж маршруту.

OSI

Open Systems Interconnection — взаємозв'язок відкритих систем — оголошена Міжнародною організацією по стандартизації (International Standarts Organization - ISO) та ухвалена у 1983 році еталонна модель, яка встановлює стандарти термінології та мережевих протоколів для взаємозв'язку відкритих систем, тобто сполучення систем з відкритою архітектурою від різних виробників. Еталонна модель має сім шарів, або рівней, кожен з яких виконує чітко визначені функції, які забезпечують обмін даними між рівнями. При цьому більш низький рівень надає послуги суміжному з ним верхньому рівню, користуючись послугами суміжного нижнього рівня.

Цими рівнями є:

- Фізичний — забезпечує безпосередній взаємозв'язок із середовищем передачі, реалізуючи механічні, електричні, функціональні та процедурні стандарти взаємозв'язку із фізичними засобами передачі даних. Виконує передачу бітів по комунікаційному каналу, забезпечуючи різницю значень 1 та 0 як таких. Приймає та передає потік бітів безвідносно його структури та змісту.
- Канальний — реалізує відповідне оформлення блоків даних, розділяючи вхідних потік на кадри, котрі передаються послідовно, знаходить помилки в каналі зв'язку та відновлює дані (після помилок) при передачі до сусіднього вузла. Містить два підрівня: верхній — управління логічним каналом (Logical Link Control — LLC), основними функціями якого є перевірка правильності передачі інформації по з'єднанню, та нижній — управління доступом до середовища передачі (Medium Access Control — MAC).
- Мережевий — основні його функції полягають в управлінні адресацією та маршрутизацією в мережі, організації передачі по одному фізичному з'єднанню пакетів, адресованих різним мережам та управління потоком інформації.
- Транспортний — слугує для забезпечення пересилання повідомлень між двома системами, які взаємодіють, із використанням нижніх рівнів. Приймає деякий блок даних від верхнього рівня, розділяє його на фрагменти, передає по нижньому, мережевому, рівню та слідкує за тим, щоб всі фрагменти над-

ходили до одержувача у вірному порядку. Реалізує істинно скрізьний (end-to-end) процес, забезпечуючи транспортування даних від відправника до одержувача: програма на вузлі-відправнику обмінюється інформацією з аналогічною програмою, яка знаходиться на вузлі-одержувачі. Робота на рівні відправник-отримувач є однією з основних відмінних ознак протоколів верхніх рівнів.

- Сеансний (або рівень сесій) — організовує способи взаємодії прикладних процесів на віддалених вузлах. Встановлення зв'язку, яке виконується сеансним рівнем, є багатокроковою операцією, що включає адресацію хосту, аутентифікацію користувача та забезпечення використовуваних комунікаційних опцій. Встановивши зв'язок, сеансовий рівень управляє "діалогом" між процесами, постачаючи прикладним функціям користувача порції даних за допомогою транспортного рівня.
- Представлення — визначає синтаксис інформації, яка передається, тобто набір знаків та способи їх представлення. Протоколи рівня представлення форматують дані для задовільнення потреб різних комп'ютерів, терміналів або середовищ представлення. На цьому рівні може також виконуватись шифрування даних з метою забезпечення безпеки при передачі інформації через мережі та/або стискання даних.
- Прикладний — визначає протоколи взаємодії користувача (процесу-відправника) з програмами на віддаленому комп'ютері (процесом-отримувачем), такими як база даних, система електронного документообігу, фінансові транзакції тощо. Забезпечує інтерфейс між прикладними програмами ПЗ та системою зв'язку, надаючи прикладним програмам доступ до різних мережевих служб (наприклад, до електронної пошти).

OSPF

Open Shortest Path First — відкритий протокол переваги найкоротшого каналу — стандарт протоколу маршрутизації в мережах Internet, заснований на алгоритмі, що враховує стан каналів.

Out-of-band signalling

Управління зовні основної полоси (по зовнішньому каналу) — спосіб управління комунікаційною апаратурою, при якому сигнали посилаються по окремому каналу, не призначеному для передачі даних.

Overrun

Перевищення швидкості — в галузі інформаційних комунікацій помилка, яка виникає в ситуації, коли пристрій, що отримує дані, не встигає їх обробляти.

P

PACE

Priority Access Control Enabled — дозволене управління доступом по пріоритету — розроблена фірмою 3Com технологія побудови комутаторів для мереж з методом доступу CSMA/CD (Carrier Sence Multiple Access with Collision Detection —

множинний доступ з контролем несучої та виявленням конфліктів), що забезпечує більш керований доступ до середовища передачі даних. Технологія дозволяє: надати кожній станції гарантовану та максимальну полоси пропускання; забезпечити передбачуваний час затримки передачі пакетів та звести його розкидання практично до нуля; визначити для кожної станції два віртуальних канала з пропускною спроможністю, що регулюється, та різними приорітетами; оптимізувати алгоритми комутації та досягнути для кожного порту 98%-го використання повної ширини полоси пропускання мережі.

Packet

Пакет — впорядкована група даних та управляючої інформації, що передається по мережі як частина довшого повідомлення.

PAD

Packed Assembler/Disassembler — пристрий "збору/розбору" пакетів — перетворює послідовний цілий потік даних, який іде від передаючого міжмережевого вузла (моста чи маршрутизатора), в пакети, які передаються по мережі з комутацією пакетів, та збирає пакети в неперервний потік даних при прийомі.

Padding

Заповнення — вставлення бітів в потік даних для виконання вимог протоколів, які визначають межі структурних одиниць інформації.

PAP

Password Authentication Protocol — протокол парольної аутентифікації — метод аутентифікації, який реалізується шляхом обміну парами ідентифікаторів/паролів між двома пристроями.

Parity bit

Біт парності — службовий біт, який додається до групи інформаційних бітів, щоб забезпечити парну (або непарну) сумарну кількість одиниць в групі. В типовому випадку комунікацій за допомогою моделі OSI парність є одним з параметрів, які повинні бути узгоджені на передаючому та приймальному кінцях каналу перед початком обміну інформацією.

Path

Шлях — канал між двома мережевими вузлами.

PBX

Private Branch Exchange — приватна (відомча) АТС — АТС, що обслуговує, як правило, державні або приватні організації, і розміщена в приміщеннях, в яких працює користувач.

PCI

Protocol Control Information — управляюча інформація протоколу — службова інформація, яка додається кожним рівнем стеку протоколів до даних при передаванні їх наступному нижньому рівню, утворюючи протокольний блок даних (Protocol Data Unit).

PDH

Plesiochronous Digital Hierarchy — плезіохронна (майже синхронна) цифрова ієрархія — спосіб передачі даних в цифрових мережах, заснований на механізмі мультиплексування з чередуванням бітів, при якому вирівнювання вхідних швидкостей потоків від різних каналів (24 при інтерфейсі PRI) реалізується завдяки додаванню потрібної кількості надлишкових бітів в канали з меншою швидкістю передачі. На приймальному кінці для відновлення вхідної послідовності надлишкові біти видаляються.

PDN

Public Data Network — мережа передачі даних загального користування — мережа комутації пакетів, послуги якої доступні широкому колу користувачів засобів обробки даних.

PDU

Protocol Data Unit — протокольний блок даних — пакет в еталонній моделі взаємодії відкритих систем. PDU є елементом обміну між протоколами одного рівня в стеку протоколів і містить поряд з даними управляючу інформацію протоколу (Protocol Control Information).

Peer to peer communication

Зв'язок між рівноправними вузлами — прямий зв'язок між пристроями, що функціонують на одному комунікаційному рівні в мережі, яка реалізується без використання проміжного пристрою, наприклад хост-комп'ютера або сервера. Протокол одноранговой комунікації (інша назва цього терміну) визначає тільки основні механізми для передачі даних: він не має потреби у вказівках відносно того, коли і з якою метою прикладна програма або вузол починають обмін даними або яким чином організований зв'язок між ними в розподіленому середовищі. Це розв'язується на рівні одноранговой операційної системи домену.

Peer to peer directory propagation

Однорівневе розповсюдження каталогів — в електронній пошті спосіб поновлення адрес користувачів, при якому зміни, що відбулись в будь-якому з поштових відділень, розсилаються всім іншим відділенням.

Peer to peer network

Одноранговая мережа — мережа, яка складається із вузлів, кожний із яких може виконувати функції як клієнта, так і сервера, і в якій зв'язок та розділення ресурсів реалізується безпосередньо на рівні вузлів, а не за допомогою допоміжного арбітру. В одноранговій мережі всі вузли працюють під управлінням однієї (одноранговой) операційної системи, котра і наділяє їх вказаними властивостями. Прикладами одноранговых ОС можуть слугувати LANtestic фірми Artisoft, Windows for Workgroup, Windows 95 і Windows NT Workstation компанії Microsoft, Personal NetWare компанії Nowell.

PEM

Privacy Enhanced Mail — конфіденційна пошта — електронна пошта в Internet, яка забезпечує секретність, аутентифікацію та цілісність повідомлень за допомогою різних методів шифрування.

Phase-locked

З фазовою синхронізацією — означення, що характеризує взаємозв'язок двох сигналів, при якому взаємовідношення фаз утримується постійним за допомогою якогось управляючого (електронного) пристрою.

PING

Packed InterNet Groper — програма, що використовується для перевірки досягненості вузла-отримувача. Програма посилає вузлу пакет в форматі Internet Control Message Protocol та чекає відповіді. Абревіатура ввійшла в професійний лексикон як слово "пінг".

PIR

Peak Information Rate — пікова пропускна спроможність — максимальна швидкість передачі інформації, яку може надати канал.

Plenum

Тип кабелю у тефлоновій оболонці, більш стійкий до впливу навколишнього середовища, ніж звичайний кабель; при горінні не виділяє токсичних газів.

PMD

Physical Medium Dependent — нижній підрівень першого (фізичного) рівня моделі OSI, залежний від типу середовища передачі.

PMI

Physical Medium Independent — верхній підрівень першого (фізичного) рівня моделі OSI, незалежний від типу середовища передачі.

Polling cycle

Циклічне опитування — в технології 100VG-AnyLAN послідовне опитування портів концентратором, при якому цикл починається з опитування ввімкнених портів з фізичним номером меншого порядку та закінчується на тому, який має найвищий номер.

PoP

Point of Presence — закінчення міжміської АТС — точка, в якій місцева телефонна компанія підключається до основної мережі.

POP

Post Office Protocol — поштовий протокол — TCP/IP-базований протокол, за допомогою якого станція-клієнт може читати та завантажувати пошту з поштового серверу. Розроблені три версії цього протоколу: POP, POP2 та POP3, причому більш пізні версії не сумісні з попередніми.

POP server

Програма-сервер, запущена на комп'ютері, підключеному до Internet у режимі "on-line", що дозволяє користувачам одержати доступ до своїх поштових скриньок по поштовому протоколу POP (як правило, це протокол POP3).

Portmapper

Служба відображення потрів — мережева служба, на якій базуються всі служби, що використовують механізм визову віддалених процедур. Portmapper є своєрідним

регістратором, який відслідковує відповідність між портами (логічними комунікаційними каналами в даному контексті) та службами, що надаються даним вузлом, і забезпечує клієнту стандартний спосіб звертання до програми виклику віддаленої процедури, яка підтримується сервером.

PPP

Point-to-Point Protocol — протокол "точка-точка" — міжвузловий протокол, що забезпечує зв'язок типу "маршрутизатор-маршрутизатор" або "хост-мережа" по звичайним телефонним лініям, що комутуються. Крім протоколу TCP/IP, підтримує більшу частину протоколів звичайних локальних мереж. Являє собою варіанти високорівневого управління каналом передачі даних (High-level Data Link Control — HDLC). Реалізовує динамічне узгодження IP-адреси перед початком обміну даними, стискання даних, автоматичну аутентифікацію, має вбудований алгоритм коригування помилок.

Presentation address

Адреса рівня представлення даних — в еталонній моделі взаємодії відкритих систем (OSI) адреса, що використовується для локалізації прикладного компонента. Містить мережеву адресу та до трьох селекторів — для транспортного та сеансового рівнів, а також для рівня представлення даних.

PRI

Primary Rate Interface — інтерфейс первинного рівня — інтерфейс мережі ICDN, що визначає дисципліну підключення станцій ICDN до широкополосних магістралей, які зв'язують місцеві та центральні АТС або мережеві комутатори. Об'єднує 23 В-каналу і D-канал для стандарту T1 або 30 В-каналів і D-канал для стандарту E1, що забезпечує сумарну пропускну спроможність 1,544 Мбіт/с і 2,048 Мбіт/с відповідно. В загальному випадку інтерфейс PRI описується формулою nB+D.

Private mode

Індивідуальний режим — в технології 100VG-AnyLAN режим роботи концентратора, при якому кожний порт отримує пакети, адресовані лише йому.

Promiscuous mode

Загальний режим — в технології 100VG-AnyLAN режим роботи концентратора, при якому кожен порт отримує всі пакети без виключень.

Prospero

Розподілена файлова система, що дає можливість користувачеві переглядати списки заданої множини файлів, розподілених по вузлах Internet, в різних представленнях. Забезпечує лише механізм найменування файлів, а доступ до них реалізується іншими методами (наприклад, FTP). Протокол Prospero використовується також для комунікації між клієнтом та сервером в системі Archie.

Proxy

"Посередник" — механізм, за допомогою якого одна система стає фронтальною для іншої по відношенню до вимог протоколу обміну. Використовується в системах управління мережами для того, щоб уникнути необхідності реалізації повного стеку протоколів в простих пристроях, таких як модем.

Proxy ARP

Технологія, при якій одна машина (як правило маршрутизатор), відповідає на запитання протоколу дозволу адреси (Adress Resolution Protocol), призначені для іншої машини. Видозмінюючи процес ідентифікації, маршрутизатор бере на себе відповідальність за доставку пакету реальному отримувачу. Схожа технологія дозволяє сайту використовувати одну IP-адресу для двох фізично різних мереж, але в таких випадках доцільніше застосовувати механізм підмереж.

Proxy server

Proxy-сервери кешують запити до серверів інших видів сервісу Internet. Звичайно вони встановлюються в локальній мережі організації або у постачальника послуг Internet таким чином, щоб канал між користувачами і комп'ютером, на якому встановлений proxy-сервер, мав досить велику пропускну спроможність. По великому рахунку, proxy-сервер виконує роль великого централізованого кешу.

Розглянемо приклад роботи proxy-сервера по протоколу HTTP, що використовується для передачі даних у системі WWW. Всі запити до віддалених Web-серверів у даному випадку проходять через proxy-сервер. Якщо документ, що потрібен, уже запитувався ким-небудь іншим, то proxy-сервер зв'яжеться з відповідним Web-сервером, перевірить актуальність збереженої в нього копії запитуваного документу і, якщо ця копія ще не застаріла, видає її у відповідь на запит. Таким чином, користувачу не доведеться чекати, поки запитуваний документ буде отриманий із віддаленого Web-сервера. Якщо ж копія документа, збережена на proxy-сервері, застаріла або її там взагалі нéма, то proxy-сервер перенаправить запит на відповідний Web-сервер і розмістить копію отриманого документу до себе в кеш. Зважаючи на те, що канали передачі даних часто бувають перевантажені, і користувачу довгий час доводиться чекати, коли ж, нарешті, буде отриманий запитаний документ або ілюстрації до нього, використання proxy-серверів може в декілька разів прискорити роботу із системою WWW.

В більшості випадків для кожного виду сервісу Internet використовуються окремі proxy-сервери.

PSN

Packed Switch Node — вузол комутації пакетів — виділений комп'ютер в мережах з комутацією пакетів, в функції якого входять приймання, вибір маршруту та передача пакету суміжному вузлу в напрямку адресата.

PVC

Permanent Virtual Circuit — постійний віртуальний канал — логічний еквівалент виділеного фізичного каналу, на відміну від якого він не існує реально, тобто його не можна "продзвонити". В маршрутизаторах, по яких "проходить" цей канал, замість фізичних з'єднань постійно прописані адреси суміжного по напрямку каналу маршрутизатора, і віртуальний канал описується деякою множиною адрес, по яким пакет "іде" до отримувача.

Q

QIR

Quiscent Information Rate — статистична швидкість передачі інформації — швидкість передачі та приймання даних, яка встановлюється в якості початкової після стану неактивності каналу. Значення QIR як правило лежить між значеннями максимальної (MIR) та пікової (PIR) швидкостей обміну.

Quark

Кварк — в X-протоколі це ціле, яке ідентифікує ім'я, клас або рядок для менеджера ресурсів. Подібно автоматам та ідентифікаторам ресурсів, кварки звільнюють від необхідності передавати рядки довільної довжини мо мережі.

R

RARP

Reverse Address Resolution Protocol — протокол зворотного дозволу адреси — призначений для забезпечення отримання бездисковими станціями своєї IP-адреси від RARP-серверу на базі апаратної адреси Ethernet. RARP функціонує на рівні драйверів пристроїв. RARP-клієнт посилає спеціальний пакет із своєю апаратною адресою, у відповідь отримує IP-адресу від RARP-серверу тієї ж підмережі. IP-адреси, присвоєні RARP-клієнтам, як правило зберігаються на RARP-сервері в файлі з базою даних у вигляді звичайного тексту, який ставить у відповідність шестибайтній апаратній адресі чотирибайтну IP-адресу.

Read-after-write verification

Верифікація "читання після запису" — метод верифікації даних ОС NetWare. Після запису блоку даних на диск виконуються читання та порівняння інформації з даними, що знаходяться в оперативній пам'яті. При співпаданні даних пам'ять очищується. При неспівпаданні збійна ділянка диску маркується як "погана" і дані перенаправляються на іншу ділянку.

Real time

Реальний час — режим роботи обчислювальної системи, при якому час відгуку на подію не перевищує наперед визначеної величини. В цьому режимі як правило додатковою є вимога, щоб час відгуку "слабо" залежав від параметрів ефективності системи, наприклад від ступеню завантаження процесора.

Receive-only link

З'єднання тільки на приймання — тип комунікаційного з'єднання, при якому шлюз відповідає на вхідні виклики, але не може ініціалізувати виклик. В цьому випадку шлюз може розглядатися як підкорений пристрій у з'єднанні головний-підкорений.

Remote

Віддалений — той, що не знаходиться безпосередньо поруч. Як прикметник використовується для позначення комп'ютеру або іншого пристрою, розташованого на деякому віддаленні, так що для доступу до нього необхідний комунікаційний канал.

Remote access

Віддалений доступ — доступ до ресурсів мейнфрейму або локальної мережі по телефонним каналам зв'язку. Розрізняють два методи віддаленого доступу. Перший, метод віддаленого клієнта (вузла), передбачає підключення комп'ютера користувача як рівноправної робочої станції. В цьому випадку модем розглядається як повільний мережевий адаптер, і вся інформація направляється через послідовний порт ПК. Єдиною перевагою цього поідходу є надання віддаленому комп'ютеру повного набору дисководів та можливості використовувати в прикладних задачах стандартні шляхи передачі файлів програм та даних. Другий метод відомий як метод віддаленого управління, при якому віддалений комп'ютер підключається до робочої станції мережі в режимі емуляції терміналу, перетворюючи дану станцію в сервер доступу. При цьому стандартне введення/виведення реалізується на віддалений ПК, а програми, що викликаються, завантажуються та виконуються на сервері доступу.

Remote dialback

Віддалений зворотний виклик — метод забезпечення безпеки, при якому у відповідь на спробу під'єднатись до віддаленого серверу виконується виклик віддаленого клієнта.

Remote shell

Віддалена оболонка — інтерпретатор команд, який ініціалізується на одній машині, але виконується на іншій, що вказується в командному рядку.

Rendezvous

Рандеву — в мережевому контексті набір процедур, які виконуються для того, щоб клієнт-одержувач міг ідентифікувати клієнта-відправника.

Reply message

Зворотня відповідь — в архітектурі клієнт-сервер передача клієнту результатів виклику віддаленої процедури.

RFC

Request For Comments — запит на коментарі — серія документів, що публікувались, починаючи з 1969 року, які описують набір протоколів Internet та узагальнюють різного роду досвід, що відноситься до функціонування цієї мережі. Не всі (фактично дуже мало) документи цієї серії безпосередньо описують стандарти Internet, але всі без винятку стандарти описані як документ RFC. Серія документів відзначається тим, що протоколи запропоновані суспільством дослідників та розробників, що керувались практичними інтересами, в протилежність формально розглянутим та стандартизованим протоколам, що рекомендовані такими організаціями як CCITT та ANSI. Документи RFC можуть біти отримані, наприклад, за допомогою анонімного FTP по адресах: `nic.ddn.mil` і `nis.nsf.net`.

RG-11

Розповсюджений тип "товстого" коаксіального кабелю мережі Ethernet з хвильовим опором 50 Ом.

RG-58 A/U

Поширений тип "тонкого" коаксіального кабелю мережі Ethernet з хвильовим опором 50 Ом.

RG-62 A/U

Поширений тип коаксіального кабелю для мережі Arcnet з хвильовим опором 93 Ом.

RIP

Routing Information Protocol — протокол маршрутної інформації — протокол динамічної розподіленої маршрутизації, заснований на алгоритмі обміну маршрутними таблицями "вектор-довжина" (Vector Distance Algorithm — VDA). Розроблений корпорацією Xerox. В маршрутизаторі, який застосовує RIP, вся інформація зберігається в маршрутній таблиці, яка містить наступні поля: пункт призначення (в ньому перераховані всі локальні мережі); наступний транзитний вузол (це поле визначає, в який порт повинен бути пересланий пакет для відправлення на наступний маршрутизатор); "відстань" (кількість транзитних маршрутизаторів між даними відповідною мережею). Динамічний вибір оптимального (в деякому розумінні) маршруту забезпечується розсиланням маршрутної інформації.

RJ-45

Тип роз'єму для приєднання кабелю на основі витих пар.

RMON

Remote Monitoring — віддалений моніторінг — протокол мережевого управління для мереж Ethernet і Token Ring. В його основі, як і в протоколі SNMP, розширенням якого є RMON, лежать збір та аналіз інформації про функціонування мережі. Ці функції виконуються апаратно-програмними агентами, дані від яких поступають на комп'ютер, що реалізує управління мережею. Основна відмінність RMON від його попередника полягає в тому, що якщо SNMP збирає дані про події, які відбуваються тільки на тому пристрої, де встановлений агент, то RMON надає інформацію про трафік між мережевими пристроями. Крім того, агенти RMON самостійно виконують аналіз даних та надсилають програмам вже частково опрацьовані дані, що знижує трафік в мережі. Інформаційна база управління (MIB) RMON складається з десяти груп даних:

- в групі статистики збирається загальна інформація про трафік;
- група попередньої історії відповідає за збирання інформації в групі статистики протягом певного часу для аналізу тенденцій;
- в групі аварійних сигналів визначаються порогові значення параметрів, при перевищенні яких генерується аварійний сигнал;
- в групі хостів реєструють всі мережеві пристрої в даному сегменті та реалізується збирання основної статистики для них;
- таблиця головних хостів (HostTopN) містить список N перших хостів з максимальним значенням заданого статистичного параметру;
- матриця трафіку містить в якості своїх елементів значення інтенсивності трафіку між станціями;
- група фільтрів використовується для фільтрації пакетів;
- в групу перехвату направляються пакети, що відповідають умовам, сформульованим в групі фільтрів;

- в групі подій визначається, яким чином реагувати на ті чи інші події, що відбуваються в мережі;
- в групі, яка відноситься до до мереж Token Ring, визначається, зокрема, порядок чередування станцій в кільці.

RPC

Remote Procedure Call — виклик віддаленої процедури — одна із реалізацій моделі розподілених обчислень в архітектурі клієнт-сервер. В загальному випадку виклик посилається на віддалену систему (сервер), яка виконує вказану процедуру, використовуючи отримані аргументи, та повертає результат клієнту.

RSVP

Resource Reservation Setup Protocol — протокол резервування ресурсів — протокол, що забезпечує підтримку в мережах TCP/IP розвинутих додатків мультимедіа (наприклад, відеоконференцій). Необхідна пропускна спроможність та часова затримка на певному рівні забезпечуються за допомогою полоси пропускання мережі.

Rubber bandwidth

"Гумова полоса пропускання" — термін, що використовується для описування комунікаційного каналу, чия полоса пропускання може збільшуватись або зменшуватись без припинення передачі та переналагодженя каналу. Як правило, для цього застосовується інверсне мультиплексування.

RZ

Return to Zero — самосинхронізуючий трирівневий код передачі даних.

S

S-interface

S-інтерфейс — інтерфейс між NT1 та будь-яким кінцевим обладнанням, здатним працювати з ISDN. Реалізується за допомогою чотиридротового кабелю. Максимальна довжина петлі — 1 км.

Socket

Програмний сокет — тип міжпрограмного інтерфейсу в мережевих комп'ютерних комунікаціях. Для того, щоб зв'язуватись через мережу, дві прикладні програми на різних комп'ютерах відкривають сокет, який є конкатенацією IP-адреси та порту протоколу (поля, що забезпечує адресацію пакету прикладній програмі). Цей інтерфейс реалізує механізм ніжнього рівня, який підтримується більшістю мережевих програм.

Spoofing

Спуфінг (імітація з'єднання) — здатність маршрутизатора реагувати на деякі мережеві запити діями місцевого характеру з метою уникнути необхідності встановлення з'єднання з віддаленим пунктом.

SCS

Structured Cabling System — структурована кабельна система — кабельна система, що підтримує різноманітні інформаційні системи (комп'ютерні, телефонні та телевізійні мережі, системи пожежної та охоронної сигналізації) та розділена на

кілька рівнів в залежності від функціонального призначення та розташування її компонентів. Сучасні SCS являють собою складний комплекс, в який, крім власне кабелю, входять конектори, з'єднувальні шнури, крос-панелі, інформаційні розетки та інше обладнання.

SLIP

Serial Line Internet Protocol — протокол, що дозволяє комп'ютеру використовувати протоколи Internet за допомогою стандартної телефонної лінії і високошвидкісного модему.

SMI

Structure of Management Information — структура управляючої інформації — вимоги, яким повинен відповідати об'єкт для того, щоб він міг бути доступним по протоколу управління мережою.

SMTP

Simple Mail Transfer Protocol — протокол, що застосовується в якості стандартного методу передачі пошти в Internet, що описаний у RFC-821. SMTP застосовується в багатьох локальних і глобальних мережах, хоча для локальних мереж існує множина інших поштових протоколів. Багато в чому SMTP схожий на протокол FTP.

SNA

System Network Architecture — архітектура мережевих систем, запропонована компанією IBM і орієнтована на об'єднання комп'ютерів самих різних типів.

SNMP

Simple Network Management Protocol — протокол обміну для віддаленої управляючої станції репитера в мережі Ethernet, яка служить для контролю навантаження мережі і інтенсивності помилок в мережі, а також для автоматичного відключення несправних сегментів.

STDM

Statistical Time Division Multiplexing — стастичне часове мультиплексування — метод мультиплексування, при якому потік даних кожного каналу передавання розділяється на окремі блоки і до них додається заголовок, що містить ідентифікатор каналу і "хвостовик", що слугує ознакою кінця блоку. При цьому методі полоса пропускання вихідного каналу надається вхідним каналам по мірі необхідності. На статистичному мультиплексуванні заснований, зокрема, метод пакетної комутації.

STR

Shielded Twisted-Pair Cable — кабель на основі екранованих витих пар.

Stub network

Мережа-заглушка — локальна мережа, по якій передаються дані лише між локальними хостами, навіть якщо є підключення до інших мереж.

Supervisor

Супервізор (адміністратор) — персона, відповідальна за адміністрування та супроводження мережі або бази даних. Володіє максимально можливими правами та повним доступом доя всієї інформації.

T

T-connector

Т-образний з'єднувач, який служить для підключення двох сегментів "тонкого" коаксіального кабелю до мережевого адаптера.

TCP/IP

Назва родини протоколів, що забезпечують передачу даних у Internet. Термін TCP/IP містить у собі назви двох протоколів — Transmission Control Protocol (TCP) і Internet Protocol (IP). TCP — це протокол, що підтримує передачу даних, засновану на встановленні логічного з'єднання між комп'ютером, що посилає дані, і комп'ютером, що ці дані отримує. IP є протоколом більш низького рівня і забезпечує фактичну передачу даних.

Telnet

Протокол, що регламентує процес передачі даних при роботі в термінальному режимі на віддаленому комп'ютері. При цьому на екрані клієнтського комп'ютера емулюється робота терміналу, а всі процеси запускаються на віддаленому комп'ютері. Також Telnet — це загальновживана назва прикладної програми, що використовується для роботи на віддалених комп'ютерах в режимі терміналу за допомогою протоколу TELNET.

TN3270

Програма, що використовується для встановлення Telnet-з'єднання з великими комп'ютерами IBM.

Token Ring

Кільцева локальна мережа компанії IBM з маркерним методом доступу і швидкістю передачі 4 Мбіт/с.

TP0

Transport Protocol class 0 — транспортний протокол класу 0 — транспотрний протокол найпростішого класу в еталонній моделі візаємодії відкритих систем (OSI). Може використовуватись лише на вершині стеку протоколів мереж X.25 або інших мереж із низькою вірогідністю втрати або перетворення інформації.

TPFDDI

Версія мережі FDDI на електричному кабелі (витої пари) зі швидкістю передачі даних 100 Мбіт/с.

Transputer (TRANSistor/comPUTER)

Трансп'ютер — надвелика інтегральна схема, яка містить процесор, комунікаційні канали для міжпроцесорного зв'язку, оперативну пам'ять та кеш невеликого об'єму. Термін "трансп'ютер" використовують для позначення серії чіпів, вперше розроблених та реалізованих компанією INMOS, хоча існують мікросхеми інших розробників з аналогічними характеристиками.

U

UA

User Agent — агент користувача — в еталонній моделі взаємодії відкритих систем (OSI) прикладний процес, який представляє користувача (персону або ор-

ганізацію) в системі передавання повідомлень X.400. Створює, представляє та розсилає повідомлення від імені користувача.

UART

Universal Asynchronous Receiver/Transmitter — універсальний асинхронний передатчик — пристрій, що виконаний, як правило, як одна інтегральна мікросхема. Містить модулі, що забезпечують асинхронні послідовні (побітні) прийом та передачу.

UDP

User Datagram Protocol — протокол дейтаграм користувача — протокол транспортного рівня в стеку протоколів TCP/IP, є спрощеним варіантом TCP. Виконує, по суті, ті ж самі функції, що і TCP, але на відміну від останнього не забезпечує перевірку на наявність помилок і не підтверджує доставлення пакета.

U-interface

U-інтерфейс — точка розмежування дводротового абоентського шлейфу ISDN та мережевого кінцевого пристрою NTI (роз'єми RG-11/RG-12).

UMA

Unified Memory Architecture — архітектура об'єднаної пам'яті — стандарт для побудови комп'ютерних систем, запропонований асоціацією VESA (Video Electronics Standards Association), що базується на загальній пам'яті для центрального процесору аудіо-, відео-, графіки, системи обробки зображення та інших підсистем вводу/виводу.

UNI

User-to-Network Interface — інтерфейс "користувач-мережа" — специфікація, запропонована Форумом ATM, яка визначає інтерфейс прямого зв'язку пристрою користувача та ATM-комутатора.

URL

Uniform Resource Locator — уніфікований вказівник ресурсу — формат зовнішніх посилань, що вказує повний шлях до конкретного документу або розділу на комп'ютері, під'єднаного до вузла Internet, а також метод доступу до нього (протокол роботи з програмами-серверами, які функціонують на віддаленому комп'ютері). На комп'ютері можуть одночасно працювати кілька програмних серверів по обслуговуванню різних методів доступу — FTP, Gopher, WWW та інші. URL складається з трьох частин: протоколу доступу, імені програми-серверу та шляху до документа (protocol://server.directory/filename). Для документів HTML з програмою-сервером www, наприклад, URL може мати вигляд `http://www.field1.field2`, де `http` — протокол доступу, `www` — ім'я сервера, а `field1`, `field2` — адреса ресурсу.

Up-link port

Порт східного зв'язку — в технології 100VG-AnyLAN порт на концентраторі низького рівня, через який він підключається до концентратора високого рівня.

USENET

Глобальна мережа UNIX-систем з децентралізованим адмініструванням, яка використовується для передавання електронної пошти та телеконференцій — прилю-

дного обговорення по цілому ряду рубрик-направлень. Початково була організована на базі програми UUCP та на телефонних комунікаціях. Не всі хости Internet дозволяють підписатись на телеконференції, і не всі хости USENET є вузлами Internet.

UTR

Unshielded Twisted-Pair cable — кабель на основі неекранованих витих пар.

UUENCODE/UUDECODE

Механізм кодування, призначений для передачі двійкових даних через ті види сервісу Internet, що споконвічно передбачали передачу лише семирозрядних ASCII-даних. Такими видами сервісу є UseNet і електронна пошта. З цілком очевидних причин, двійковий файл не може бути безпосередньо вставлений в текст повідомлення. Для передачі таких файлів було розроблено декілька механізмів, одним із яких є кодування даних по алгоритму UUENCODE/UUDECODE. Після такого кодування, всі дані з файлу перетворяться в семирозрядний ASCII-формат, і цей файл уже може бути вставлений в текст повідомлення електронної пошти або статтю UseNet. Приклад того, який вигляд мають закодовані в такий спосіб дані, наведений нижче:

```
begin 644 aaa.zip
M4$L#!'H''''''-
&'MB#K:90ZR@''',H''''"''''ODA6+E185$)(5B'()&@MK:ДО
B+8^EXJ7@H>/@HPT*#OJ$J."EJN*NX""HIZ2@XJ6K['BHJ'@D
:7@HZ6EMH@T*@J"DJ*P@@*NEJN&@K:3@KJ*HYPT*#OJ2I:LN(# 4T,2TX-
2TU,OT*#OJ!MKJ'@XZGAJJ#0(..ДО+BP@1"X@-"P@HN*NX*ZI
(.WBH*8L#OKGI>"MH.\@IJ6KMI:>MH.\@I**EX.PN#OH-"H^NY^*H(*V@(.
.CJ^,@BZ7AK:ZCKB"OX"X@J"#CMJRX@D:RHX
*VNHJ'N#OIO2P$"%''*''''''#1@+8@ZVF4.LH'''#*''''!P''L'''''''!'"
''''''''''' ODA6+E185%!+!O8''''''O'!'#4'''#O''''''''''
end
```

Після отримання повідомлення з такими даними користувачу необхідно їх декодувати для отримання вихідного двійкового файлу. Сучасні програми для роботи з електронною поштою і UseNet залишають механізм кодування даних "за кадром", користуючись механізмом MIME. Єдина дія, яку виконує користувач при відправленні файлу, — приєднання файлу до повідомлення, що відправляється. Проте, варто враховувати, що кодування даних у семирозрядний формат веде до збільшення їхнього об'єму. Звідси випливає наступна рекомендація: користуйтесь для файлів, що приєднуються до повідомлення, програмами архівації.

V

VAN

Value-added network — мережа з додатковими послугами — мережа, яка працює, наприклад, по протоколу комутації пакетів X.25. Забезпечує виправлення помилок, включає засоби резервування та інші засоби, що забезпечують високу надійність передачі даних.

VDA

Vector Distance Algorithm — алгоритм вектор-довжина — алгоритм маршрутизації, заснований на обміні таблицями вектор-довжина. Таблиця містить пари (V,D):

V (вектор) — адреса доступної мережі, а D (довжина вектору) — "відстань" до відповідної мережі, яка вимірюється кількістю транзитних маршрутизаторів.

VLAN

Virtual LAN — віртуальна ЛОМ — об'єднання користувачів (кінцевих станцій), під'єднаних до фізично різних сегментів ЛОМ в логічні робочі групи за допомогою програмних засобів таким чином, що комунікації між ними стають "прозорими", ніби вони працюють в одному мережевому сегменті. Для побудови VLAN можуть бути використані будь-які характеристики кінцевих станцій: MAC-адреса, IP-адреса, адреса підмережі, тип протоколу і т. ін. Однак найбільше розповсюдження отримали три моделі, або способи конфігурації віртуальних локальних мереж:

- **За допомогою портів**. Логічний сегмент в цьому випадку створюється шляхом закріплення необхідної кількості портів в маршрутизаторах та комутаторах за конкретною VLAN. Повідомлення широкого віщання розповсюджуються по всім фізичним сегментам визначеної таким чином віртуальної мережі, але блокуються при спробі передачі іншим сегментам.
- **По MAC-адресах**. Віртуальна ЛОМ визначається на основі створених таблиць MAC-адрес кожної кінцевої станції.
- **По мережевій адресі** (IP-адресі хостів). Цей спосіб відрізняється від попереднього тим, що замість таблиці MAC-адрес створюється таблиця IP-адрес, і відповідно маршрутизація пакетів виконується на мережевому рівні моделі OSI.

VSAT

Very Small Aperture Terminal — малогабаритний термінал для цифрового супутникового зв'язку з діаметром антени 1–3 м.

W

WAIS

Wide Area Information Server — потужна система пошуку інформації в розподілених по Internet базах даних. Як і WWW, WAIS дозволяє знаходити ресурси в мережі й одержувати до них доступ незалежно від того, де вони знаходяться. Принцип роботи системи WAIS заснований на використанні ключових рядків при здійсненні пошуку інформації в індексованному матеріалі. Від користувача потрібно лише грамотно скласти ключовий рядок, по якому буде виконано пошук інформації — всю іншу роботу з пошуку ресурсів WAIS виконує самостійно.

WAN

Wide Area Network — глобальна обчислювальна мережа.

WWW

World Wide Web — "всесвітня павутина" — один із напопулярніших та багатих по можливостям сервіс Internet, що використовує для пошуку ресурсів Internet і доступу до них систему гіпертекстових посилань. Система WWW, подібно величезному павутинню, охоплює практично всі напрямки знань, інформація про які була коли-небудь внесена в комп'ютери, підключені до Internet.

Додаток В

Теги та атрибути мови HTML

Атрибути використовуються всередині тегу і відділяються один від одного пробілом. Розглянемо приклад простого HTML-документа, в якому наведено використання атрибутів тегу <body>:

```
<html>
  <body bgcolor=blue text=white>
    Текст документа
  </body>
</html>.
```

В даному прикладі атрибут bgcolor визначає фоновий колір документа при його відображені у Web-браузері, а атрибут text — колір тексту.

Для визначення кольорів можна вказувати просто їх назви (як в наведеному прикладі), або використовувати запис виду #RRGGBB, що відповдає моделі RGB, де RR — частка червоного кольору, GG — частка зеленого кольору, а BB — частка синього. Всі значення вказуються в шістнадцятковій системі счислення. Наприклад, значенню #000000 відповідає чорний колір, значенню #FFFFFF — білий, значенню #00FF00 — чистий зелений, а значенню #FFFF00 — жовтий колір.

Перелік основних тегів та атрибутів мови HTML наведений в табл. В.1.

Таблиця В.1

Тег	Атрибути	Опис
<html>	Дескриптор початку Web-документа	
<head>	Шапка документа	
<body>	Основний зміст документа	
	alink=*колір*	Колір активованого гіперпосилання
	bgcolor=*колір*	Колір фону
	background=*URL*	Графічний фон з вказівкою адреси зображення
	bgproperties	Фіксування фонової ілюстрації
	leftmargin=*n*	Відступ від лівого краю
	link=*колір*	Колір гіперзв'язку
	text	Колір тексту
	topmargin=*k*	Відступ від верхнього краю
	body vlink=*колір*	Колір гіперпосилання, що вже було використане
<base>	Вказується URL-адреса	
	href=*URL*	Базовий URL
	target=*Рядок_символів*	Ім'я кадру
<basefont>	Вказується базовий шрифт документа	
	size=*n*	Розмір шрифту

Таблиця В.1. Продовження

Тег	Атрибути	Опис
<isindex>	Організація пошуку за ключовим словом	
	action=*URL*	URL місця пошуку
	prompt=*рядок символів*	Запит (підказка) по шуканому тексту
<link>	Зв'язок з іншими документами	
	href=*URL*	URL документа, на який дано посилання
	methods=*методи*	Метод посилань, специфічний для браузера
	rel=*зв'язок*	Зв'язок з документом
	rev=*зв'язок*	Зв'язок від документа
	title=*рядок символів*	Заголовок документу, на який дане посилання
<meta>	Передача службової інформації	
	name=*рядок символів*	Імена параметрів службової інформації
	content=*рядок символів*	Службова інформація
	Ділить текст на логічні фрагменти	
	align=right	Текст розміщується праворуч
	align=left	Текст розміщується ліворуч
	align=center	Текст розміщується посередині
<h1> ... <h6>	Рівень підзаголовків	
<p>	Текстові абзаци	
	align=*вирівнювання*	Вирівнювання по горизонталі. Можливі значення: top, bottom, center
<blockquote>	Цитати	
	Центрування виділених фрагментів документу	
<listing>	Текст програми, яка застаріла	
<xmp>	За допомогою цього тегу можна коригувати екран	
<plaintext>	Переформатовує текст	
<pre>	Заздалегідь відформатований текст	
	width=*n*	Ширина тексту
 	Перенесення рядка, обрив тексту	
	clear=*напрямок*	Вибір позиції фрагменту тексту, що переноситься на наступний рядок
	brcleav=center	
	brcleav=right	
	brcleav=left	

Таблиця В.1. Продовження

Тег	Атрибути	Опис
<cite>	Текстове посилання	
<kbd>	Текст, що вводиться з клавіатури	
<samp>	Текст прикладу	
	Виділення тексту	
	Сильне виділення тексту	
<dfn>	Визначення	
<ins>	Підкреслений текст	
<abbr>	Абревіатури	
<strike>	Перекреслений текст	
<a>	Якір або гіперпосилання	
	name=*рядок_символів*	Ім'я локального посилання
	href=*URL*	Адреса посилання
	rel=*зв'язок*	Зв'язок до документу
	rev=*зв'язок*	Зв'язок від документу
<address>	Інформація щодо авторства	
<b>	Жирний шрифт	
<caption>	Надписи в таблиці	
	align=*вирівнювання*	Вирівнювання по горизонталі. Можливі значення: left, right, center
	valign=*вирівнювання*	Вирівнювання по вертикалі. Можливі значення: top, bottom, center
<dir>	Каталог	
	compact	Змінений зовнішній вигляд
<font>	Задання параметрів шрифта	
	size=*n*|*+n*|*-n*	Розмір шрифта
	color=*колір*	Колір
	face=*назва_шрифта*	Вид шрифта
<form>	Форма формуляру	
	action=*URL*	Адреса обробки формуляра на сервері
	method=get|post	Спосіб передачі параметрів
	enctype=*кодування*	Кодування тексту
<i>	Курсив	
<img>	Вставні графічні зображення	
	src=*URL*	Адреса зображення
	alt=*рядок_символів*	Текстовий опис зображення
	align=*вирівнювання*	Зображення відносно тексту
	height=*n*	Висота зображення у пікселах
	width=*n*	Ширина у пікселах

Таблиця В.1. Продовження

Тег	Атрибути	Опис
<img>	border=*n*	Ширина рамки
	hspace=*n*	Ширина незаповненого простору ліворуч і праворуч від зображення
	vgpace=*n*	Висота незаповненого простору вище і нижче зображення
<input type=checkbox>	Прапорець у формулярі	
	checked	Прапорець встановлено
	name=*рядок символів*	Ім'я елементу
	value=*рядок символів*	Значення, що встановлене по замовчуванню
<input type=file>	Вибір файлу в формулярі	
	maxlength=*n*	Максимальна кількість символів
	name=*рядок символів*	Назва елементу
	size=*n*	Кількість видимих символів
<input type=hidden>	Схований елемент в формулярі	
	maxlength=*n*	Максимальна кількість символів
	name=*рядок символів*	Назва елементу
	size=*n*	Кількість видимих символів
	value=*рядок символів*	Значення, що встановлене по замовчуванню
<input type=image>	Командна кнопка користувача	
	align=*вирівнювання*	Вирівнювання графіки
	name=*рядок символів*	Назва елементу
	src=*URL*	Вихідний URL графіки
<input type=password>	Введення пароля в формулярі	
	maxlength=*n*	Максимальна кількість символів
	name=*рядок символів*	Назва елементу
	size=*n*	Кількість видимих символів
	value=*рядок символів*	Значення, що встановлене по замовчуванню
<input type=radio>	Селекторна кнопка в формулярі	
	checked	Вибрана
	name=*рядок символів*	Назва елементу
	value=*рядок символів*	Значення, що встановлене по замовчуванню
<input type=reset>	Кнопка очищення всіх полів формуляра	
	value=*рядок символів*	Надпис на командній кнопці

Таблиця В.1. Продовження

Тег	Атрибути	Опис
<input type=submit>	Кнопка відправки формуляра	
	name=*рядок символів*	Назва елементу
	value=*рядок символів*	Надпис на командній кнопці
<input type=text>	Поле введення тексту в формулярі	
	maxlength=*n*	Максимальна кількість символів
	name=*рядок символів*	Назва елементу
	size=*n*	Кількість видимих символів
	value=*рядок символів*	Зміст поля, що встановлений по замовчуванню
	Елемент списку	
	type=*вид символу*	Маркер чи символ нумерації
	value=*рядок_символів*	Явне задання номеру елементу
	Упорядковані (ненумеровані) списки	
	type=*нумерація*	Тип нумерації
	start=*n*	Стартовий номер
	compact	Змінений зовнішній вигляд
<select>	Меню обо поле списку у формулярі	
	size=*n*	Кількість елементів, що одночасно відображаються на екрані
	multiple	Поле списку з можливістю множинного вибору
	name=*рядок символів*	Ім'я елемента
<sub>	Підрядковий знак	
<sup>	Надрядковий знак	
<table>	Визначення таблиці	
	align=*вирівнювання*	Горизонтальне вирівнювання таблиці. Можливі значення: left, right, center
	bgcolor=*колір*	Колір фону
	bordecolor=*колір*	Колір рамки
	bordecolordark=*колір*	Колір рамки, темна частина
	bordercolorlight=*колір*	Колір рамки, світла частина
	width=*n*|*n*%	Ширина таблиці
	border=*n*	Товщина ліній рамки
	cellspacing=*n*	Проміжок між комірками
	cellpadding=*n*	Проміжок між вмістом комірки і рамкою
	hspace=*n*	Відступ до таблиці по горизонталі
	valign=*вирівнювання*	Вирівнювання по вертикалі
	vspace=*n*	Відступ до таблиці по вертикалі

Таблиця В.1. Продовження

Тег	Атрибути	Опис
<td>	Табличні данні (комірка)	
	nowrap	Відміна перенесення слів
	rowspan=*n*	Кількість рядків, перекритих коміркою
	colspan=*n*	Кількість стовбчиків, перекритих коміркою
	align=*вирівнювання*	Горизонтальне вирівнювання вмісту комірки. Можливі значення: left, right, center
	valign=*вирівнювання*	Вертикальне вирівнювання вмісту комірки. Можливі значення: top, bottom, center.
	width=*n*	Ширина комірки
	height=n	Висота комірки
	bgcolor=*колір*	Колір фону
	bordercolor=*колір*	Колір рамки
	bordercolordark=*колір*	Колір рамки, темна частина
	bordercolorlight=*колір*	Колір рамки, світла частина
<th>	Комірка табличного заголовку	
	nowrap	Відміна перенесення слів
	rowspan=n	Кількість рядків, перекритих коміркою
	colspan=n	Кількість стовбців, перекритих коміркою
	valign=*вирівнювання*	Вертикальне вирівнювання даних у комірці. Можливі значення: top, bottom, center
	width=*n*	Ширина комірки
	height=*n*	Висота комірки
	align=*вирівнювання*	Горизонтальне вирівнювання у комірці. Можливі значення: left, right, center
	bgcolor=*колір*	Колір фону
	bordercolor=колір	Колір рамки
	bordercolordark=*колір*	Колір рамки, темна частина
	bordercolorlight=*колір*	Колір рамки, світла частина
<title>	Зовнішній заголовок	
<tr>	Табличний рядок	
	align=*вирівнювання*	Горизонтальне вирівнювання в комірці. Можливі значення: left, right, center
	valign=*вирівнювання*	Вертикальне вирівнювання в комірці
	bgcolor=*колір*	Колір фону
	border=*n*	Рамка таблиці
	bordercolor=*колір*	Колір рамки
	bordercolordark=*колір*	Колір рамки, темна частина
	bordercolorlight=*колір*	Колір рамки, світла частина

Таблиця В.1. Продовження

Тег	Атрибути	Опис
<u>	Підкреслювання	
	Невпорядковані (ненумеровані) списки	
	type=*тип*	Стиль для елементів
	compact	Зменшення проміжку між елементами
<option>	Елемент списку в формулярі	
	selected	Задавання опції по замовчуванню
	value=*n*	Параметри елемента
<marquee>	Рухомий рядок	
	align=*вирівнювання*	Вирівнювання по вертикалі. Можливі значення: top, bottom, center
	bgcolor=*колір*	Колір рядка
	behavior=*поведінка*	Стиль руху рядка
	scrolldelay=*n*	Час очікування між пропусками
	height=*n*	Висота вікна
	directoon=*напрям*	Напрям руху рядка
	hspace=*n*	Відступ вікна по горизонталі
	loop=*n*|*infinite*	Повторення рядка, що біжить
	vspace=*n*	Відступ від вікна по горизонталі
	width=*n*	Ширина вікна
<textarea>	Багаторядкове поле у формулярі	
	rows=*n*	Кількість рядків видимого тексту
	cols=*n*	Видима ширина тексту
	name=*рядок символів*	Ім'я поля
<applet>	Вставка аплету Java	
	align=*вирівнювання*	Вирівнювання у вікні. Можливі значення: left, right, center
	alt=*рядок символів*	Альтернативний текст, який з'являється, коли аплет не функціонує
	code=*клас*	Клас Java-програми
	codebase=*URL*	Базова адреса завантаження аплета
	height=*n*	Висота вікна аплету
	hspace=*n*	Відступ по горизонталі до вікна аплету
	name=*рядок символів*	Ім'я аплету
	vspace=*n*	Відступ по вертикалі до вікна аплету
	width=*n*	Ширина вікна аплету
<area>	Область, яка розпізнає "клікання" мишею з метою виклику певного ресурсу	
	coords=*координати*	Список координат, які задають область

Таблиця В.1. Продовження

Тег	Атрибути	Опис
<area>	href=*URL*	URL-ресурс для даної області
	nohref	Відсутність URL для даної області
	shape=*форма_області*	Форма області
<bgsound>	Фоновий звук	
	loop=*n*|*infinite*	Повторення звуку
	src=*URL*	URL звукового файлу
<big>	Збільшення шрифту	
<code>	Вихідний текст програми	
<comment>	Коментарій	
<dd>	Текст визначення в списку визначень	
<dt>	Визначений термін в списку термінів	
<frame>	Визначення кадру (незалежної області вікна браузера для відображення Web-сторінки)	
	marginheight=*n*	Відступ від верхнього/нижнього краю
	marginwidth=*n*	Відступ від лівого/правого краю
	name=*рядок символів*	Назва кадру
	noresize	Пропозиція по зміні розмірів
	scrolling=*полоси_прокрутки*	Приховування/показ смуг прокрутки
	src=*URL*	Вихідний URL для даного кадру
<frameset>	Визначення структури фреймів	
	cols=*n*|*n%*|**,.*	Кількість і ширина стовбців з кадрами
	rows=*n*|*n%*|**,.*	Кількість і ширина рядків з кадрами
<hr>	Горизонтальна лінія	
<map>	Визначення області для розпізнавання події "клікання" мишею (див. <area>)	
	name=*рядок символів*	Назва даної області
<nobr>	Абзац без розривів рядків	
<noframes>	Розділ для Web-браузерів, що не підтримують кадри	
<param>	Передача параметрів для аплетів	
	name=*рядок символів*	Ім'я параметра
	value=*рядок символів*	Значення параметра
<s>	Перекреслений текст	
<small>	Менший шрифт	
<tt>	Моноширний текст	
<var>	Текст змінної	
<q>	Текст в дужках	

Таблиця В.1. Закінчення

Тег	Атрибути	Опис
<lang>	Визначає мову відображення	
<au>	Автор	
<person>	Ім'я персони	
<acronym>	Акронім	
<abbrev>	Абревіатура	
<ins>	Вставлений текст	
<del>	Знищений текст	
<basefont>	Базовий розмір шрифта	
	size=*n*	Розмір
<multicol>	Багатоколоночний текст	
	cols=*n*	Кількість колонок
	gutter	Пропуск між колонками
	width=*n*	Ширина колонки
<spacer>	Порожній блок	
	type=*тип_блоку*	Тип порожнього блоку
	size=*n*	Розмір порожнього блоку
	width=*n*	Ширина порожнього блоку
	height=*n*	Висота порожнього блоку
	align=*вирівнювання*	Вирівнювання
<embed>	Вставка об'єкта	
	src=*URL*	Адреса об'єкта
	width=*n*	Ширина
	height=*n*	Висота
<hr>	Горизонтальний роздільник	
	align=*вирівнювання*	Вирівнювання. Можливі значення: left, right, center
	size=*n*	Товщина
	width=*n*|*n*%	Ширина
	height=*n*|*n*%	Висота

Додаток Г

Розробка Web-сайтів

Г.1. Етапи розробки Web-сайту

Розглянемо послідовність робіт (етапів) при розробці Web-сайту:

1. Визначитись з кінцевою метою створення сайту.
2. Ознайомитись з інформацією щодо створення сайтів:
   - `www.vitraj.dp.ua` — як створювати персональний сайт;
   - `www.itware.com.ua` — кращі сайти української мережі Internet.
3. Провести аналітично-маркентингове дослідження (аналіз) ринку Web-сайтів, вивчити "конкурентів".
4. Визначитись з програмно-технічним інструментарієм проектування сайту:
   - мови: HTML, Java Script, XHTML, DHTML, XML;
   - редактори HTML: HomePage, HomeSite, Word, EditPlus;
   - системи FrontPage;
   - технології mmFlash, Parser, Fordis.
5. Знайти фахівців, що допоможуть розробити сайт (або хоча б проконсультувати):
   - Web-дизайнер;
   - програміст;
   - контент-менеджер;
   - системний адміністратор;
   - Web-психолог;
   - Web-майстер.
6. Визначитись з коштами і часом, які можливо витратити на створення сайту (від 3000 грн., від 2-х місяців).
7. Продумати ідею "оригінальності" сайту, тобто створити ефективну бізнес-ідею.
8. Розробити бізнес-план сайту (прогноз-план).
9. Визначитись зі змістом сайту та його обсягом (середній сайт займає декілька десятків Мбайт).
10. Визначитись з цільовою аудиорією.
11. Розробити структуру сайту:
    - рубрики;
    - течії;
    - розділи;
    - модулі;
    - зв'язки;
    - лінійна (один аркуш за іншим);
    - ієрархічна (один до багатьох);
    - мережева (усі з усіма).

12. Визначитись яким буде сайт:
    - статистичним;
    - динамічним.
13. Визначити:
    - систему навігації по сайту;
    - можливості по модифікації.
14. Розробити дизайн сайту, тобто дизайн-концепцію (стиль, макет, шаблон, оформлення і т. ін):
    - кольори: основні; фонові; гіперпосилання;
    - шрифти: розмір; тип; вигляд;
    - фото, анімація, звук.
15. Продумати наявність інтерактивних (діалогових) елементів:
    - чати (розмови);
    - гостьові книги;
    - форуми;
    - лічильник;
    - анкетування;
    - безкоштовна інформація.
16. Підготовка текстового матеріалу:
    - спочатку можна у Microsoft Word з перевіркою орфографії або блокноті;
    - потім — у FrontPage або DreamWeare, або Flash.
17. Підготовка графічних матеріалів (сканер; Photoshop; Flash 5/Mx) та елементів оформлення анімації.
18. Верстка — створення з різних файлів, модулів, шаблонів, сторінок, програм, БД інтегровного інтерактивного Web-сайту:
    - Web-майстри — Internet Explorer, Netscape Navigator;
    - Front Page;
    - Flash.
19. Тестування (перевірка) сайту.
20. Розміщення у мережі — ***хостінг***: необхідно зробити на сайті гіперпосилання на ті системні сайти, які часто відвідуються.
21. Реклама сайту:
    - на сайтах партнерів та друзів;
    - на візитках;
    - створити посилання на персональний сайт на основних пошукових серверах Internet: закордонні — `google.com`, `infoscek.com`, `nothernlight.com`, `geocity.com`, `yahoo.com/help/`, `dmos.com`; вітчизняні — `meta.ua`, `topping.com.ua`; російські — `yandex.ru`, `rambler.ru`, `narod.ru`, `lycos.ru`.
22. Здійснювати постійне вдосконалення сайту, його підтримку, супроводження, виправлення помилок, т.т. робити:
    - редизайн;
    - реструктуризацію;
    - інтеграцію з новими IT.

Г.2. Загальні вимоги до Web-сайту

Хоча у Web-дизайні немає твердих правил, існує ряд загальних вимог, яких дотримуються при розробці Web-сайтів:

- сайт повинен мати:
  - головну сторінку;
  - декілька додаткових сторінок з інформацією про господаря сайту (резюме; коло наукових інтересів; комерційні проекти; пропозиції до співпраці; хоббі; фотоальбом і т.ін);
- навігація по сайту повинна бути інтуїтивно зрозумілою навіть дитині, тобто бути простою для використання;
- сайт не повинен бути переобтяжений анімацією, що ускладнює його сприйняття, або музикою;
- гіпертекстові посилання повинні бути кольоровими, наприклад, текст — чорний; фон — білий; посилання — сині;
- не використовувати на сайті більше, ніж двох-трьох типів шрифту;
- не використовувати:
  - кадри;
  - смуги прокрутки;
- текст сайту повинен бути:
  - лаконічним;
  - чітким;
  - виразним;
  - не набраним великими літерами;
- сайт повинен мати "девіз" господаря та логотип;
- сайт повинен часто оновлюватись та мати мінімальний час для завантаження (тобто файли повинні мати розмір менше 10–12 Кбайт);
- сайт повинен мати якусь "родзинку", чимось приваблювати;
- якщо є можливість, треба залучати до розробки або реконфігурації сайту фахівців-професіоналів :
  - Web-дизайнерів;
  - Web-майстрів;
  - програмістів;
  - психологів;
  - ергономістів;
  - системного адміністратора;
  - менеджерів (контент-менеджера, управління інформацією);
  - маркетолога;
  - верстальника тощо;
- графічні файли рекомендовано зменшити за рахунок скорочення кількості кольорів з 256 до 16;
- запропонуйте відвідувачу сайту щось безкоштовне, але корисне для нього:
  - інформацію по комп'ютерному авторському праву;
  - безпеку чи небезпеку;
  - співпрацю, за яку самі заплатите;

- безкоштовну консультацію з питань створення Web-сайту;
- подарунок, який Вам не дуже потрібний.

- Web-сайт може включати наступні засоби інтерактивного зворотного зв'язку:
  - гостьові книги — це демократичний і відкритий зворотний зв'язок спілкування з відвідувачами сайту; реєстрацію та підтримку гостьової книги пропонують сервери `guestbook.ru` та `guest.onecenter.com`;
  - форуми — це дискусія на сайті по кожному питанню чи темі; відкрити форуми на сайті допоможуть сервери `forum.onecenter.com`, `guest.onecenter.com`, `forum.eist.ru`;
  - чати – це спілкування на сайті усіх з усіма у реальному часі; список популярних чатів можна подивитися на сайті `Chats.top_100.ru`; організувати чат можна за допомогою сайтів `chats.ru`, `chatpod.com`, `chatterpop.com`.
  - лічильник — дозволяє визначити число відвідувань сайту; установити лічильник на сайті можна за допомогою сайтів `rumbler.ru`, `yndex.ru`, `bigmir.net` (веде статитику відвідувань українських сайтів), `TopPing`.
  - голосування — це спосіб довідатись про думки відвідувічів з якогось конкретного питання; організувати опитування відвідувачів допоможуть сайти `voting.ru`, `freetools.com`, `opros.alkar.net/`.
- більшість дизайнерів рекомендують розробляти сторінки у форматі:
  - 640x480 пікселів (щоб не застосувати горизонтальну прокрутку);
  - 800x600 пікселів (щоб якомога більша аудиторія користувалась сайтом).

Додаток Д

Історія Internet

1957 рік

Наприкінці 50-х років над Землею "забібікала" маленька металева кулька: радянський штучний супутник Землі — РШС3. Це дуже розхвилювало військових у США і як наслідок у 1957 році терміново створюється Агентство Перспективних Розробок (ARPA — Advanced Research Project Agency) у військовій предметній області. Через 10 років перед цим агентством постала нова проблема: об'єднати роботу дослідників, що розкидані по всій країні по принципам:

- кожен має зв'язок з кожним;
- вихід з роботи будь-якого вузла мережі не повинен впливати на працездатність мережі в цілому.

1960 рік

Розроблено перший модем. Розроблено методику "пакетної" передачі даних (Пол Берен, компанія AT&T).

1962 рік

Розроблена концепція "Галактичної мережі".

Вийшла у світ книга Дж. Лайклідера та У. Кларка "On-line Man Computer Communication" ("Інтерактивна комп'ютерна взаємодія людини з комп'ютером").

1965 рік

ARPA починає розробку моделі сумісної роботи компьютерів.

1967 рік

Друкується перший проект мережі Arpanet (Ларрі Робертс).

1968 рік

Перша демонстрація примітивного гіпертексту (Стенфордський інститут, Дуглас Енгельбарт).

1969 рік

Створена ОС UNIX (К. Томпсон, Д. Рітчі).

Здійснено зв'язок між двома компьютерами у Каліфорнійському Університеті (Лос-Анжелес). Вдалось передати слово на відстань 5 м.

Здійснено зв'язок між чотирма компьютерами Університетів:

- Каліфорнійський (Лос-Анжелес);
- Стенфордський (Стенфорд);
- Каліфорнійський (Санта-Барбара);
- Університет Юти (Солт-Лейк-Сіті).

Мережу назвали Arpanet (фінансування здійснювало Міноборони США).

1970 рік

Прокладається перший в історії канал комп'ютерного зв'язку між двома країнами (компанія AT&T).

1971 рік

Народження електронної пошти (Рей Томлінсон з програмою SENDMSG). В адресі e-mail з'являється літера "@").

До Arpanet підключені 15 дослідницьких установ США.

1972 рік

Винаходять систему зв'язку компьютерів, яку називають Ethernet (Боб Меткелф).

1973 рік

До мережі Arpanet підключаються:

- University College (Лондон)
- Royal Radar Establishment (Норвегія).

Розроблена специфікація передачі файлів по мережі (прототип FTP).

Кількість користувачів Arpanet — 2000.

З'являється термін Internet (Intercommercial Nerworks — об'єднані мережі).

Народжується концепція мережевого протоколу TCP/IP (впроваджений у 1983 році).

1974 рік

Народжується сучасна концепція гіпертексту (Тед Нельсон), мультимедія, гіпермедія.

1975 рік

Створена перша діюча поштова розсилка, яка об'єднує тих, хто полюбляє наукову фантастику (Стів Уолкер).

1976 рік

Створюється протокол UUCP для з'єднання компьютерів через звичайну телефонну лінію (компанія AT&T).

Розроблено коаксіальний кабель для передачі даних у LAN.

Вперше здійснена передача даних по каналу супутникового зв'язку

До мережі Arpanet підключено більше 100 компьютерів.

1979 рік

Створено USEnet — групу новин.

Створено MUD — мережева гра.

1981 рік

У Швейцарії створено центр CERN (European Center for Nuclear Research), аналог ARPA.

1982 рік

Розроблено протокол TCP/IP (впроваджено у 1983 році).

Керівництво ARPA приймає рішення виділити з агентства закриту мережу MILnet — лише для військових, а Arpanet стає відкритою та загальнодоступною.

Кількість підключених до Arpanet компьютерів — більше 500.

1984 рік

Кількість підключених до Arpanet компьютерів – 1000.

Створення D&S.

Народження Фідонет — безкоштовної компьютерної мережі.

1986 рік

Кількість підключених до Arpanet компьютерів — понад 2000.

Створена NFSnet — наукова мережа (яка у подальшому і стала сучасною Internet).

1988 рік

Кількість серверів Arpanet сягає 50000.

З'являється перший мережевий вірус — "хробак" Роберта Морріса.

1990 рік

Кількість серверів Arpanet — понад 300000.

Створення компанія Relcom — Reliable Communication

Народження ru.net (Росія)

1991 рік

Створена нова мова для Internet — Java (впроваджена з 1995 року).

Народження WWW — всесвітнього "павутиння" (система гіпертекстових сторінок).

Розроблено текстовий браузер для перегляду HTML (Швейцарія, Тім Бернерс-Лі).

Розроблено Gopher для пошуку інформації (Університет Мінесоти, США).

Розроблено PGP (Pretty Good Privacy) — система шифрування повідомлень у мережі (Філ Ціммерман).

1992 рік

З'являється термін "Internet-серфінг" т.т. розроблено програмне забезпечення для створення системи WWW.

Кількість серверів (Інтернет-хостів) — понад 1000000.

Розроблено Mosaic X — перший браузер (Марк Андресіен, Ерік Бін).

Народження української мережі Internet UANet.

Створена національна мережа УкрПАК.

1993 рік

Перші радіотрансляції по Internet.

Створена організація InterNIC для координації послуг по реєстрації у Internet.

1994 рік

З'являється браузер Netscape Navigator (Марк Андріссен).
Перші Internet-магазини, віртуальні банки, банери.
Розроблено ядро ОС Linux (Лінус Торвальдс)
Створена організація по розвитку Internet — WWW Consortium .
Перший спам.

1995 рік

З'являється:

- Internet Explorer — браузер;
- Real Audio — технологія трансляції аудіоінформації;
- Alta Vista — пошукова система.

Кількість користувачів — понад 10 млн.
Випуск ОС Windows 95.

1997 рік

До Internet підключено 100 млн. користувачів.

1998 рік

20 березня вперше проводиться Всесвітній День Internet.
З'явився перший mp3-плейєр.
150 млн. користувачів.
Перші Internet-аукціони та портали.
Запуск нової мережі — Internet-2.
Початок ери порталів.

1999 рік

В мережі з'являються перші музичні файли у форматі MP3.
Кількість серверів — 50 млн.
Кількість користувачів — 300 млн.
З'явився відомий вірус Melissa.

2000 рік

Доля жінок у Internet — понад 50,4 %.
Найбільш масова атака вірусу "I Love You" (30 млн. компьютерів у світі).
Кількість серверів — 100 млн.
Кількість користувачів — 500 млн.

2002 рік

Кількість користувачів — 580 млн. (США, Канада — 180 млн., Європа — 185 млн., Росія — 9 млн., Азія, Африка — 175 млн.).

Додаток Е

Internet в Україні

Організації з питань розвитку Internet на Україні:

- **ІнАУ** — Internet-асоціація України (www.inau.org.ua);
- **АУРІУ** — ІнАУ для регіонів, Асоціація участников ринку Internet України (www.auriu.org);
- **УІС** — Українська Internet-спільнота (www.uic.org.ua);
- **АІТУ** — Асоціація інформаційних технологій України (www.itukraine.org.ua);
- **УНІАН** — Українська національна Internet-асоціація (www.unian.net);
- **ДКЗУ** — Держкомзв'язок України (www.stc.gov.ua);
- **УЗТІ** — Управління зв'язку, телекомунікацій та інформації Київської міської Держадміністрації (www.cti.kmda.gov.ua);
- **ІСУ** — Інформаційна спільнота України (www.isu.org.ua);
- **УМІС** — Українська мережа Інформаційної спільноти (www.e-ukraine.org.ua);
- **ООО "Хостмастер"** — Українська спільнота "Хостмастер", діючий адміністратор домену .UA (www.hostmaster.net.ua);
- **УСІЦ** – Український мережевий інформаційний центр (www.ua-nic.net);
- **МФВ** — Міжнародний фонд відродження (www.irf.kiev.ua);
- **АОЗ "Телас"** — Асоціація операторів зв'язку "Телас" (www.telas.kiev.ua);
- **ДЦІР** — ДержЦентр інформаційних ресурсів України (www.dzir.gov.ua);
- **ООРДІ** — Організація офіційного реєстрування доменних імен другого рівня у домені .UA (www.imena.com.ua).

Найбільшою компанією України з питань впровадження і розвитку Internet вважається Internet-асоціація України (ІнАУ). Вона об'єднує 72 членів. ІнАУ створена з метою консолідації зусиль усіх зацікавлених сторін з питань розвитку Internet на Україні. До складу ІнАУ також входить третейский суд (www.arbitre.org.ua) і підприємство "UA-IX"", що опікується Internet–трафіком (www.ua-ix.net.ua).

Відомі українські пошукові системи:

- http://meta.ua/;
- http://www.uaplus.com/;
- http://www.favortes.com.ua/;
- http://www.atlasua.net/;
- http://www.uaportal.com/;
- http://bigmir.net;
- http://www.google.com.ua/;
- http://www.silver.kiev.ua;
- http://www.search.kiev.ua;
- http://www.meta.kharkiv.net.

Додаток Ж

Формати файлів

Ж.1. Загальні відомості про стандарти і формати

Розмаїтість інформації, що обробляється за допомогою комп'ютерів, обумовило появу певних стандартів зберігання цієї інформації та, відповідно до них, форматів файлів, в яких вона зберігається. З одного боку, стандартизація необхідна для того, щоб програмам обробки інформації було "зрозуміло", яку інформацію вони обробляють; з іншого боку, полегшується життя і користувачам — вони миттєво можуть зрозуміти, з чим мають справу. Проте, разом з очевидними перевагами стандартизації існують і певні труднощі, пов'язані, насамперед, з орієнтацією користувача в сучасному "океані" форматів файлів. Для того, щоб наблизитись до вирішення цієї проблеми, розглянемо основні стандарти і формати файлів — як ті, що відомі широкому колу користувачів, так і специфічні для певних галузей комп'ютерного світу.

Інше питання, пов'язане з представленням інформації, — це сумісність різних форматів та засоби її досягнення; воно сьогодні набуває особливої актуальності у зв'язку з розповсюдженням комп'ютерних засобів обробки інформації разом з розширенням спектру її видів — мультімедійні технології, розвиток комп'ютерних мереж, зв'язок СУБД з прикладними програмами (наприклад, бухоблік, АРМ, підготовка документації тощо). Про це піде мова в кінці додатку, де будуть розглянуті сучасні програмні засоби конвертації файлів. Отже, сучасність вимагає від програмістів чіткого вирізнення всіх видів інформації на рівні програмного та аппаратного забезпечення, а від користувачів — певного рівня ознайомленості з ними.

Ж.2. Стандарти на формати файлів документів

Системи управління документами (СУД) забезпечують впорядковану передачу документів у мережі, що задовольняє вимогам конфіденційності, швидкодії та надійності. СУД треба відрізняти від ПЗ документообігу, яке тільки визначає шлях руху документів. В системах типу "клієнт-сервер" в задачах керування документами з'являється ряд додаткових факторів. Користувачам надаються розподілені обробка та доступ, а також засоби керування документообігом і графічної взаємодії, тобто коло обов'язків менеджерів значно розширюється.

Стандартизація в деякій мірі допомагає уніфікувати різноманітні компоненти СУД. Основними стандартам є ODMA (Open Document Management API) та DMA (Document Management Alliance). Перший з них забезпечує сумісність прикладних програм, другий створено на базі стандартів DEN (Document Enabled Networking) і Shmarock. Стандарт DEN визначає вимоги до ПЗ проміжного рівня, яке слугує інтерфейсом між прикладним клієнтським ПЗ та сервером архіву електронних документів. Shmarock уніфікує структуру архивів документів маштабу підприємства.

Популярність мови HTML, що використовується для документів WWW, обумовила відновлення зацікавленості до стандарту SGML. Серед інших пропозицій та

стандартів де-факто треба відзначити OpenDoc і Microsoft OLE (Object Linking and Embedding), що визначають формати комплексних документів. У програмних продуктах Acrobat і Acrobat Capture (компанії Adobe Systems), які обробляють документи із складною графікою, використовується формат PDF (Portable Document Format).

Перелік стандартів на формати файлів документів наведено в табл. Ж.1.

Таблиця Ж.1

Назва стандарту	Ким запропоновано	Призначення
DMA	Assotiation for Information and Image Management (AIMM)	Об'єднує стандарти DEN та Shamrock, що регламентує створення електронних архівів підприємств
OpenDoc OLE	Component Integration Laboratories — Microsoft Corporation	Стандарти на складні документи, що дозволяють створювати розподілене, мобільне, модульне ПЗ об'єктно-орієнтованої ахітектури
ODMA	Консорціумом виробників за підтримки відділення GroupWare компанії Novell	Визначає простий інтерфейс прикладного ПЗ для сполучення ПЗ настільних комп'ютерів з СУД
PDF PDF	Adobe Systems	Відкритий формат файлів, що не залежить від платформи
SGML	Консорціум виробників SGML Open	Містить вимоги обміну документами та структурованими інформаційними обє'ктами у відкритих системах

Ж.3. Стандартні формати файлів для обміну

В табл. Ж.2 наведені стандартні формати файлів для обміну між різноманітними типами прикладних пакетів, таких, як, наприклад, текстові редактори, електронні таблиці, СУБД тощо.

Таблиця Ж.2

Формат	Де використовується
SDF	Standart Data Format — текстові файли з фіксованим форматом запису без роздільника між рядками
ASCII	American Standart Code for Information Interchange — американський стандартний код для обміну інформацією
DIF	Data Interchange Format — формат файлів пакету VisualCalc
WKS, WK1	Формати документів пакету Lotus 1-2-3
WR1	Формати документів пакету Symphony
DOC	Document — формати документів пакету Word (MS Office)
MDB	Microsoft Data Base — формати документів пакету Access
DBF	Data Base FoxPro — формати документів пакету FoxPro та Visual FoxPro
XLS	Excel — формати документів пакету Excel (MS Office)
PPT	PowerPoint — формати документів пакету PowerPoint (MS Office)

На сьогодні компанії, що займаються розробкою сучасних офісних пакетів намагаються обладнати свої програми вбудованними конверторами файлів для підвищення продуктивності при роботі з декількома офісними пакетами. Це стосується як конвертації файлів одного типу (наприклад, `.txt` та `.doc` або `.dbf` та `.mdb`) так і створення комплексних документів (наприклад, підготовка у Word наказу і додавання до нього таблиці з Excel і потрібного звіту з Visual FoxPro).

Ж.4. Формати графічних файлів

В сучасному програмуванні використовується велика кількість форматів для зберігання графічної інформації, і майже кожна програма, що працює з графікою, вводить свій формат. І тут природньо з'являється проблема сумісності файлів з графікою, створених в різних програмах. Проте, за наявності деякої інформації про формати та основні принципи їх побудови, подібних проблем можна досить легко позбутись.

Відповідно до двох принципів побудови зображення на дисплеї — растрового та векторного — графічні формати поділяють на дві великі групи: растрові та векторні формати. Растрові формати розрізняються за глибиною представлення кольору. Так, наприклад, 8-бітне представлення забезпечує опис 256 кольорів, а 24-бітне — яке ще називають True Color — 16,5 млн. кольорів. Векторні формати набули широкого розповсюдження в САПР-програмах та в графічних векторних програмах. В окремих форматах може зберігатись різноманітна інформація, включаючи растрову і текстову, в такому випадку вони називаються метафайловими.

Перелік основних растрових форматів наведений в табл. Ж.3.

Таблиця Ж.3

Формат	Характеристики
BMP Windows / OS/2 Bitmap	Растрові формати BMP (тип файлу — BMP, розширення `.bmp`, `.rle`, `.dib`) є власними для BMP-Windows та OS/2. Підтримують 24-бітне представлення кольору
CPT Corel PHOTO-PAINT	Тип файлу та розширення — `.cpt`, підтримує 1-, 2-, 4-, 8-, 16-, 24-, 32-бітне представлення кольору та 8-бітні зображення в градаціях сірого
DBX DataBeam	Тип файлу — DBX, розширення — `.dbx` та `.imb`. Використовується, головним чином, для зберігання і пересилки документів у системах DataBeam
DIB Windows Bitmap	Апаратно незалежний формат використовується в багатьох програмах Windows
GIF CompuServe	CompuServe Grasphics Interchange Format (тип файлу і розширення — `.gif`) належить компанії CompuServe і використовується нею. Підтримує до 256 кольорів (8 біт). Часто застосовується в Internet, може містити прозорі області, та підтримувати пошарове (з поступовим збільшенням дозволяючої спроможності) відображення при завантаженні

Таблиця Ж.3. Закінчення

Формат	Характеристики
ICO Winows Icon	Тип файлу та розширення — `.ico`, використовується Microsoft Windows для відображення елементів інтерфейсу та забезпечує 1-, 4-, 8-, 24-бітне кодування кольору та може містити декілька зображень.
JPG JPEG	Тип файлу — JPG, розширення — `.jpg`, `.jff`, `.jfif`, `.jif`, і `.jtf` — JPEG TIF. Використовує алгоритм стискання з дефектами якості
PCD Kodak Photo CD	Тип файлу та розширення — `.pcd`. Формат є патентованим форматом фірми Eastmen Kodak для запису зацифрованих зображень на компакт-диски. Містить зображення розміром 2048x3072 пікселів. За бажанням користувача можна завантажувати зображення наступних розмірів: 128x192, 256x384, 512x768, 1024x1536
PCX PC Paintbrush	Тип файлу — PCX, розширення — `.pcx`, `.pcc`. Формат є стандартним для багатьох програм на IBM PC. Може зберігати зображення розміром 64000x64000 пікселів при 24-бітному представленні кольору
PIC Pixar PIC	Тип файлу та розширення - `.pic`. Найчастіше використовується для представлення результатів роботи програм візуалізації. Файли можуть підтримувати 8-, або 12-бітне кодування кольору
RLE Windows Bitmap	Аналогія BMP-Windows / OS/2 Bitmap
TGA Truevision	Тип файлу та розширення — `.tga`. Стандартний кольоровий формат, який набув розповсюдження в обробці відео та мультімедіа. Забезпечує 8-, 15-, 16-, 24-, 32-бітне кодування кольору та альфа-канал
TIF TIFF	Tagged Image Format (тип файлу TIFF, а розширення `.tif`) підтримується майже всіма графічними програмами. Забезпечує 24-, 32-бітне кодування кольору. Максимальна кількість кольорів і палітра залежать від класу TIF
XMB X Windows Bitmap	Тип файлу та розширення — `.xmb`, створений для представлення чорно-білих зображень в X-Windows фірми MIT X, в основному для курсорів і піктограм. В своєму тілі формат містить код на мові C, що описує зображення. Апаратно незалежний

Основні векторні графічні формати наведені в табл. Ж.4.

Таблиця Ж.4

Формат	Характеристики
AI Encapsulated Post-Script	PostScript Language — це незалежна від пристроїв мова опису сторінок. Encapsulated PostScript — EPS є метафайловим форматом і дозволяє зберігати різні дані — векторні, растрові, текстові. AI-файли є власними для Adobe Illustrator і записуються як EPS з розширенням `.ai` або `.eps` у формі текта ASCII. AI підтримує чисельні примітиви малювання, розширену обробку шрифтів, напівтони, кольорові ефекти та розділенння кольорів
CDR CorelDRAW	Тип файлу та розширення — `.cdr`, є одним із стандартів фірми Corel
CGM Computer Graphics Metafile	Тип файлу та розширення — `.cgm`, вважається стандартом для обміну векторною графікою між векторними графічними пакетами. Обробляє RGB-модель кольору
CLP Windows Clipboard	Тип файлу та розширення — `.clp`, растровий, векторний та метафайловий формат, власний для буферу обміну у Windows а тому може містити всілякі данні. Має проблему системної залежності
MET PM-Metafile	Тип файлу та розширення — `.met`; метафайловий формат Presentation Manager Metafile є власним для ОС OS/2 і використовується як OS/2 Clipboard.
PCT1, PCT1 Macintosh PICT	Типи файлів — PCT1 (PCT2), розширення — `.pct`; є власними форматами для Macintosh Clipboard
PIC Lotus	Тип файлу та розширення — `.pic`, є власним форматом Lotus 1-2-3 PrintGraph і використовується, головним чином, для створення простої графіки на базі електронних таблиць
WMF Windows Metafile	Тип файлу та розширення — `.wmf`, є власним внутрішнім форматом Windows. Підтримує як растрове, так і векторне представлення і 24-бітну кольорову модель RGB. Більшість Windows-програм та Windows Clipboard підтримують цей формат
WPG WordPerfect Graphic	Тип файлу та розширення — `.wpg`, створений для роботи з графікою в родині текстових процесорів WordPerfect. Підтримує максимум 256 кольорів

Ж.5. Формати звукових файлів

Комп'ютер працює з графікою та звуком у цифровому вигляді. Звукова інформація (як і будь-яка інша) зберігається у файлах. Звукові файли поділяються на три типи: з оцифрованим звуком, з нотним записом, з нотним записом та оцифрованим звуком. Отже, розглянемо які формати використовує кожний з них.

Файли з оцифрованим звуком бувають двох видів: із заголовком (*header*) та без нього (*headerless*, *raw*). У заголовку вказуються параметри, що характеризують оцифрований звук.

Перелік файлів із заголовком наведено в табл.. Ж.5.

Таблиця Ж.5

Розширення	Опис
.aiff, .aif	Тип AIFF (Audo IFF), комп'ютери Apple, SGI
.au	Комп'ютери Sun, NeXT, DEC
.avr	Запропоновано фірмою Audio Visual Research
.hcom	Застосовано кодування Хафмана
.sbk	SoundFond Bank — оцифровки для WT-синтезатора звукової плати SoundBlaster AWE32 фірми Creative Labs
.pat	Patches — оцифровки для WT-синтезатора звукової плати UltraSound компанії Advanced Gravis
.iff	Тип IFF/8S VX, комп'ютери Amiga
.nti	Комп'ютери Amiga
.smp	Програма SampleVision
.snd	Комп'ютери Sun, NeXT
.sou	Sound File
.voc	Voice File (фірма Creative Labs)
.wav	Waveform Audio File (Microsoft Corp.)
.snd	Комп'ютери IBM, Macintosh, Amiga. (байт, моно)
.sb, .sw, .ub, .ul, .uw	За угодою 8000 Hz, моно

Файли з нотним записом містять послідовність команд, що повідомляють, яку ноту, яким інструментом і як довго потрібно "програвати" в той чи інший момент часу. Формат може передбачати звучання декількох інструментів одночасно — тоді мова йде про відповідну кількість голосів. Найбільш відомі стандарти, що дозволяють це робити, наведені в табл.. Ж.6.

Таблиця Ж.6

Назва стандарту	Автори
MIDI, Bsic MIDI, Extended MIDI	Roland
LA	Roland
GS	General Sound
XG	Extended General

MIDI-файли бувають трьох форматів:

- **формат 0** — одна багатоканальна доріжка, яка може нести інформацію про 16 каналів, тобто для 16 інструментів;
- **формат 1** — одна чи декілька паралельних доріжок, що звучать одночасно;
- **формат 2** — одна чи декілька незалежних послідовних доріжок.

Найбільш часто зустрічаються файли з розширеннями, наведеними в табл. Ж.7.

Таблиця Ж.7

Розширення	Автори
.cmf	Creative Music File (фірма Creative Labs)
.mid	MIDI (Musical Instrument Digital Interface)
.mff	MIDI File Format
.orc	Програма Digital Orchestrator Plus (фірма Voyetra)
.org	Програма FM Intilligent Organ (фірма Creative Labs)
.rmi	MIDI
.pol	Програма Visual Composer (фірма Adlib)
.sng	Програма Sequencer Plus (фірма Voyetra)

При синтезуванні музики її оцифровку можна зберігати не лише в пам'яті звукової плати, але і в самому файлі з нотним записом. Такий файл містить і нотний запис і оцифрований звук. Цей спосіб почали застосовувати в файлах `.mod` ще для комп'ютера Amiga, а надалі, завдяки гнучкості цього формату, він набув широкого розповсюдження на різних типах комп'ютерів.

Перелік форматів з нотним записом наведено в табл. Ж.8.

Таблиця Ж.8

Розширення	Автори
.amf	Внутрішній модуль DSMI (Digial Sound and Music Interface)
.dmf	Delusion Digital Music File (32 канали)
.far	Farandole tracker format (16 каналів)
.mod	Підтримується багатьма програмами для IBM-сумісних комп'ютерів. 4 канали, є варіанти до 32 каналів
.mtm	Multi Tracker Module (32 канали)
.nst	Noice Tracker Module (4 канали)
.okt	Oktalizer Module (8 каналів)
.sbi	Sound Blaster Instrument File
.stm	Scream Tracker Module (4 канали)
.s3m	Scream Tracker Module (16 каналів)
.ult	Ultra Trtacker Module (32 канали)
.wow	Grave Composer Format (8 каналів)
.xm	Extended Module (2,4,6,8,10,..,32 канали)
.669	Composer 669 Module (8 каналів)

Ж.6. Формати мультимедіа-файлів

Мультімедійне програмне забезпечення є, у певному розумінні, поєднанням відео- та аудіо-інформації. Але вони не просто поєднані — між ними існує динамічний зв'язок і взаємозалежність. В залежності від ступені складності цього зв'язку ускладнюється спосіб зберігання такої інформації; відповідно змінюються формати мультімедійних файлів.

До мультімедійних файлів відносять формально і ті, які були розглянути у двох попередніх пунктах, хоча на перший погляд, мультімедійними є лише файли, в яких звук та зображення поєднані безпосередньо. Але не треба забувати, що мультімедіа-програми можуть використовувати аудіо- та відео-файли паралельно, наприклад, завантажуючи послідовно зображення `*.jpg` і відтворюючи яку-небудь мелодію `*.mid`.

Формати мультимедійних файлів наведені в табл. Ж.9.

Таблиця Ж.9

Розширення	Призначення
`.avi`	Audio Video Interleaved — відео зі звуковою доріжкою. Доступ через Windows-утиліту Media Player
`.bmp`	Bitmap — формат Windows для збереження графіки. Доступ — через Paintbrush. Використовується в OS/2
`.cgm`	Computer Graphics Metafile — 16-кольорова графіка. Міжнародний стандарт для переміщення простих графічних зображень в родинах ЕОМ IBM, Macintosh. Доступ — в CorelDRAW та багатьох інших програмах
`.flc`, `.fli`	Flick — анімаційний формат, розроблений фірмою Autodesk; може працювати з багатьма програмами
`.jpg` (`.jpeg`)	Joint Photographic Experts Group — високоякісні фотографії, що зберігаються у спеціальному форматі стискання
`.mff`	MIDI File Format — музика; сумісний з багатьма програмами і секвенсорами
`.mid`	MIDI — музика; дещо скорочений MIDI, розроблений спеціально для MS Windows. Доступ через Windows-утиліту Media Player
`.mov`	QuickTime — відеоформат, з'явився вперше на Appleax, а зараз успішно функціонує в ОС Windows
`.mpg` (`.mpeg`)	Motion Picture Expert Group — формат з високим ступенем стискання, потребує спеціального ПЗ і ТЗ для выдтворення та запису
`.pcd`	Photo CD — формат для збереження фотографій; розроблений фірмою Kodak
`.pcx`	Популярний графічний формат, доступний для багатьох графічних програм, включаючи Paintbrush
`.tif` (`.tiff`)	Tagged Image File Format — формат Aldus PageMaker для переміщення графічних об'єктів і будь-яких ПЗ та ПК та при скануванні. Дозволяє зберігати чорно-білі та кольорові зображення у вигляі бітових карт
`.voc`	Creative Voice — звук. В цьому форматі зберігається звук плати Sound-Blaster.
`.wav`	Waveform — основний формат для збереження записаних звуків у Windows

Додаток 3

Дистанційна освіта

3.1. Загальні положення

У час Internet-технологій багато аспектів нашого життя переноситься в мережу, прискорюючи тим самим темпи розвитку інформаційного суспільства і долаючи географічні бар'єри. Не стає виключенням і освіта. Зараз вже не обов'язково знаходитись поруч з викладачем. Достатньо великий час існує заочна форма навчання студентів. Але її можливості дуже обмежені. Internet дає змогу розширити їх, зробити заочне навчання справді повноцінним та всеохоплюючим.

Дистанційне навчання (ДН) — це сукупність наступних засобів:

- засоби надання учбового матеріала студенту;
- засоби контролю успішності студента;
- засоби консультації студента програмою-викладачем;
- засоби інтерактивної співпраці викладача і студента;
- можливість швидкого доповнення курсу новою інформацією, коригування помилок.

Підручник завжди був ключовим елементом в системі освіти, але з переходом освіти на електронний рівень він має відповідати новій якості. Поширеною помилкою є сприйняття підручника як тексту. Це пов'язано з його паперовою формою, а звідси — неспроможністю заміни живого спілкування. Але сучасний комп'ютер — це потужний комплекс по вирішенню різноманітних задач, і його можливості значно ширші. Тому логічним є намагання привнести нові елементи до класичного поняття підручника. Більш того, враховуючи заочність, електронний підручник повинен взяти на себе частину функцій викладача, збільшуючи загальну ефективність ДН.

Наступна концепція систематизує підхід до розробки електронних підручників за допомогою визначення комплексу вимог до його структури.

Основні вимоги:

- можливість аутентифікації користувачів;
- можливість інтерактивного коригування учбового матеріалу підручника;
- можливість диференціювання за рівнем підготовки;
- можливість персоніфікації навчання;
- мжливість швидкого поточного тестування;
- збезпечення опосередкованого взаємозв'язку "студент-викладач";
- можливість функціонування підручника як автономно, так і у складі системи ДН.

Основні компоненти:

- головна сторінка — під головною сторінкою розуміють комплекс Web-сторінок, для відвідування яких не потрібна авторизація, і які містять наступні елементи:

- інформацію про предметну область викладеного матеріалу;
- короткий опис підручника;
- інформація про необхідний рівень знань для ефективного засвоєння матеріалу;
- зміст всіх питань підручника;
- інформація про умови реєстрації тощо;

- учбова частина — цей розділ являє собою комплекс, який повинен забезпечувати навчання студента; він складається з наступних компонентів:
  - навчальні модулі та система контролю — окремий модуль має містити: коротку інформацію щодо ефективного вивчення матеріалу модулю; теоретичний матеріал модулю; практичні завдання по матеріалу модулю та приклади розв'язування задач; вправи для самостійної роботи; список додаткової літератури; список рефератів з даної теми; засоби контролю, такі як тести, експрес-опитування, контрольні роботи, лабораторні роботи та система оцінювання;
  - система навігації;
  - система пошуку;
  - система підказок та рекомендацій;
  - словник термінів;
- засоби взаємодії з іншими компонентами системи ДН;
- викладацька частина — являє собою сукупність різноманітних засобів перевірки робіт студентів викладачем та можливість мануального коригування учбового процесу окремого студента.
- бази даних — якщо підручник є частиною курсу ДН, то деякі БД будуть спільними для кількох компонентів курсу (наприклад, база викладачів), якщо ж підручник є автономним, то всі бази даних є локальними.

Загальна схема взаємодії між структурними компонентами підручника зображено на рис. 3.1.

3.1.1. Вимоги щодо інформаційного забезпечення підручника

Однією з особливостей є можливість швидкого внесення змін до контекстного наповнення учбової частини. Тому під час конкретної реалізації потрібно таким чином реалізувати структуру даних, щоб вона не вимагала великих зусиль по її коригуванню. Необхідно максимізувати кількість інформації, що зберігається у базі і визначає вигляд та структуру учбової частини. Схему роботи такої системи можна представити у вигляді, наведеному на рис. 3.2.

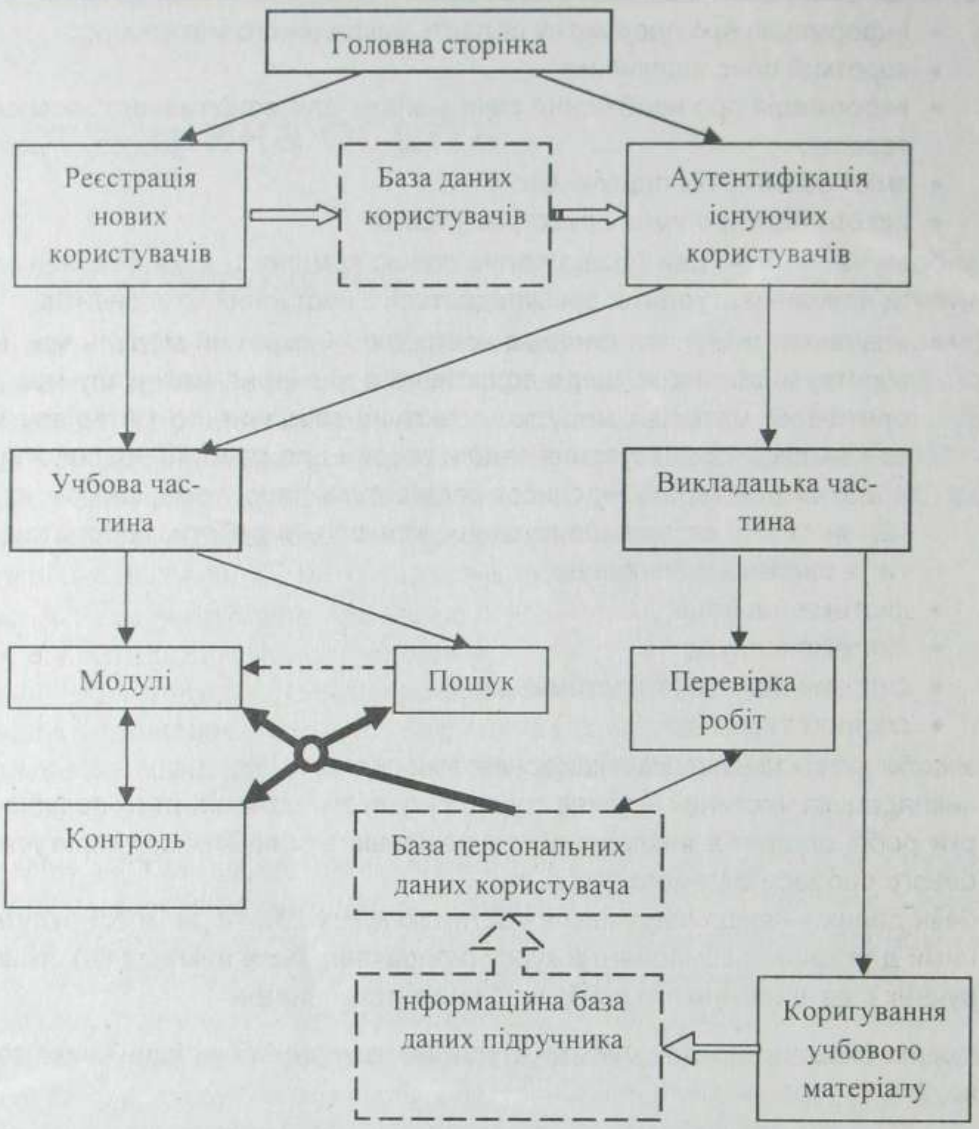


Рис. 3.1. Загальна схема взаємодії між структурними компонентами підручника

Розглянемо більш детально окремі етапи цього процесу:

- **статичні документи** — це сторінки з теоретичним матеріалом, що викладається у даному підручнику; оскільки цей матеріал займає великий обсяг, відносно рідко зазнає змін, і зміни, як правило є комплексними, то зберігання їх у динамічній базі даних слід вважати недоцільним;
- **данні про структуру модулю** — це інформація, яка стосується наповнення таких компонентів, як перелік літератури, список тем рефератів, тестів тощо;
- **система компонування навчальної частини** — це програмні засоби, які об'єднують статичні та динамічні елементи учбової частини;
- **данні про структуру тестів** — це інформація, на основі якої генерується елемент контролю знань; під час генерації, питання на контроль повинні вибиратись випадковим чином з переліку існуючих запитань по темі;
- **динамічні бази даних** — тут накопичуються, класифікуються та структуруються дані, знання, метазнання стосовно всіх інформаційно-технологічних змін, що стосуються системи дистанційного навчання.

В кінці процесу, система остаточного компонування вибудовує представлення модулю, з яким буде працювати користувач.

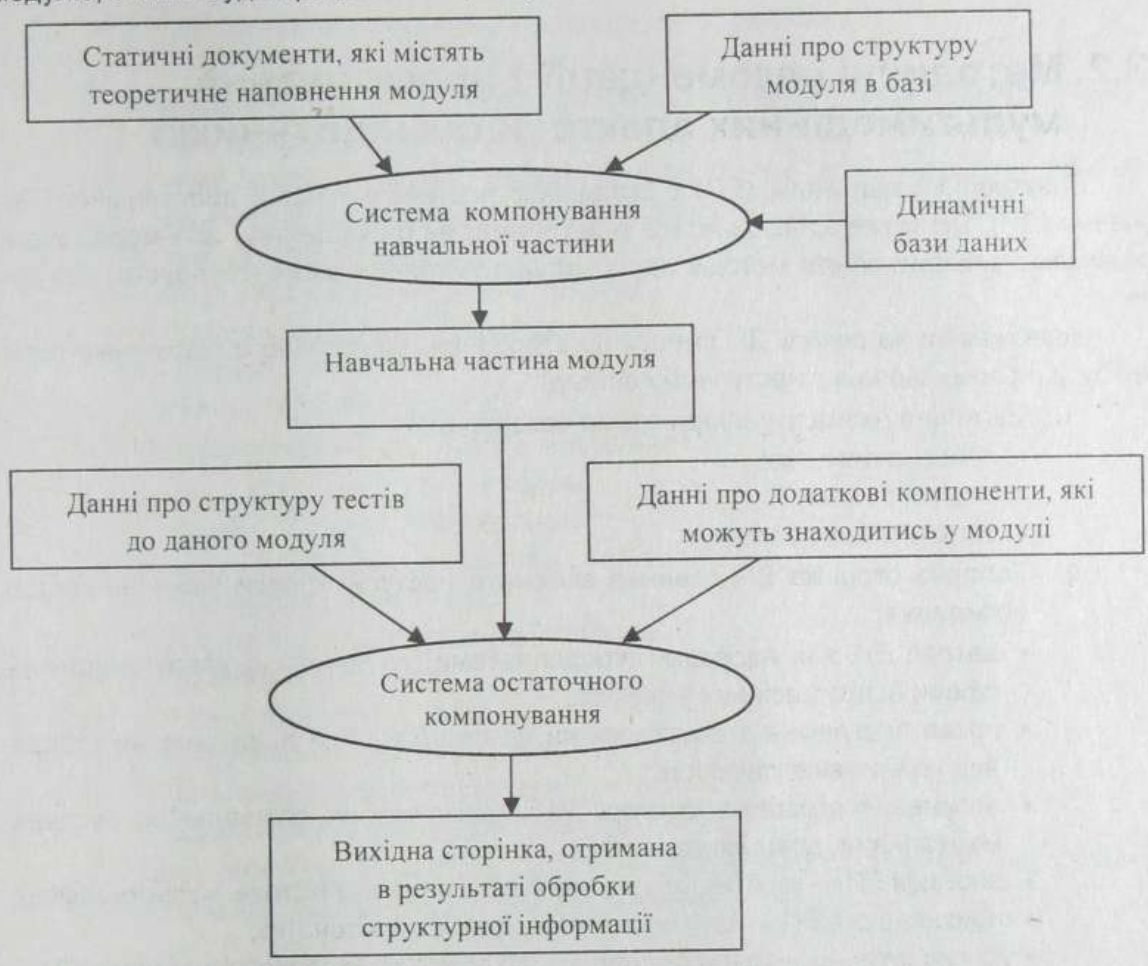


Рис. 3.2. Схема роботи системи дистанційного навчання

Опишемо механізм внесення змін до контекстного наповнення:

- внесення змін до теоретичної частини проводиться шляхом редагування статичних документів; час, що відводиться на їх оновлення — умовно-необмежений, тому до закінчення роботи користувачі працюють із старою версією;
- внесення змін до динамічних компонентів навчальної частини проводиться за допомогою редагування відповідних таблиць у базах даних.

Деякі зміни у підручнику, що пов'язані із структурою навчальної частини (наприклад, додавання чи видалення модулю), пов'язані із значними перетвореннями інших пов'язаних з цим структур (системи тестів, оцінок тощо). Якщо ці зміни проводитимуться вручну, то дуже багато часу піде на виправлення помилок, що в свою чергу призведе до зниження якості обслуговування користувачів. Тому система, що розв'язує цю проблему, є не просто електронним підручником, а являє собою засіб створення підручників одного типу.

Якщо ж піти далі та забезпечити можливість зміни інших частин електронного підручника (головної сторінки, викладацької частини), то така система буде вже ге-

нератором широкого переліку підручників, які знайдуть своє втілення у багатьох учбово-наукових сферах.

3.2. Методичні рекомендації з проектування мультимедійних електронних підручників

Електронний підручник (ЕП) є складовою частиною курсу з дистанційного навчання (ДН). ДН може здійснюватись в автономному (локальному) або мережевому режимах з використанням методів персонального і колективного (групового) навчання.

Незважаючи на режим ДН вимоги до створення електронного підручника повинні бути сформульовані у наступному вигляді:

1. Загальна (концептуальна) схема розділів ЕП:
   - головна сторінка;
   - модулі (порції);
   - загальна інформація.
2. Головна сторінка ЕП повинна включати наступні кнопки (ключові слова), посилання:
   - автори ЕП з їх адресами (координатами) та короткою характеристикою сфери їх професійних інтересів;
   - назва підручника з вказівкою, чи це оригінальний підручник, чи створений на базі вже існуючого;
   - пояснення стосовно категорії ЕП: нормативний, спеціальний, експериментальний, довідниковий тощо;
   - анотація ЕП — коротка, розширена (текстова, гіпертекстова, мультимедійна);
   - призначення ЕП — для початківців, для професіоналів;
   - якими знаннями треба володіти, щоб ефективно засвоїти матеріал ЕП — по цій темі, по суміжним темам;
   - зміст всіх питань ЕП;
   - назва модулів ЕП — меню;
   - література — основна, додаткова, URL;
   - тезаурус
   - нормативно-правова база;
   - реєстрація користувачів ЕП та створення відповідних баз даних — БД нових користувачів, БД архівних користувачів, БД постійних користувачів (користувачів-співпрацівників).
3. Модулі ЕП:
   - перелік модулів ЕП та коротка інформація з вимогами стосовно до ефективності вивчення матеріалу модуля;
   - теоретичний матеріал модуля ЕП у вигляді гіпертексту, мултимедіа, гіпермедіа;
   - задачі, завдання до (по) матеріалу модуля;
   - приклади розв'язування задач;
   - вправи для самостійної роботи;
   - система ранжованих підказок;

- система навігації відповідей, нечітких ("розмитих") підказок, наприклад 10, 20, 30,..,100%;
- система тестів з вибором відповіді з 5–10 варіантів;
- контрольні підсумкові роботи по модулю;
- експрес-опитування, тобто контрольні питання з обмеженням часу на відповідь, або блок відповідей;
- теми запропонованих рефератів (оглядів), які треба написати після засвоєння кожного модуля (при необхідності);
- теми для самостійних робіт;
- блок рейтингової оцінки за модуль;
- список допоміжної літератури;
- словник.

4. Загальна інформація по ЕП:
   - задачі, завдання до всього матеріалу ЕП;
   - приклади розв'язування задач;
   - вправи для самостійної роботи;
   - система ранжованих підказок, навігації;
   - тестові завдання по ЕП;
   - контрольна підсумкова робота по ЕП;
   - теми рефератів по ЕП;
   - теми самостійних робіт;
   - модульна рейтингова оцінка;
   - загальний список літератури;
   - загальний гіпермедійний словник.
5. Блок сертифікації слухачів та видачі відповідних документів, що засвідчують якість навчання з дистанційної освіти.

3.3. Українські центри дистанційної освіти

Для з'ясування майбутнього дистанційної освіти в Україні надзвичайно важливим є не лише вивчення попиту, а й дослідження множини спеціальностей та дисциплін з можливостями дистанційного навчання. Наведемо спочатку перелік доступних з Internet курсів, які пропонуються відомими українськими центрами дистанційного навчання. Маються на увазі таки центри, на які є посилання на офіційних сайтах, або які зареєстровані у каталогах ресурсів Internet. Оцінити якість навчальних матеріалів в більшості випадків не вдалось через обмеженість доступу до цих ресурсів. Основна ж маса відкритих матеріалів являє собою електронні варіанти методичок на кшталт тих, що масово випускаються для студентів-заочників.

Міжнародний дослідно-навчальний центр інформаційних технологій та систем.

`http://tel.dlab.kiev.ua`

Підрозділ Кібернетичного центру ім. В. М. Глушкова, дослідно-навчальна організація, підпорядкована Міністерству освіти та Національній академії наук. Це перша

освітня організація в Україні, що має досвід у розробці та впровадженні курсів дистанційного навчання.

Міжнародний центр дистанційної освіти на базі Української Академії державного управління при Президентові України

http://www.uapa-dlc.org.ua/

Основна аудиторія для навчання у Центрі :

- урядовці, що приймають рішення на найвищих рівнях державного управління;
- державні службовці (центрального та місцевого рівня);
- керівники державних, неурядових і приватних підприємств, установ та організацій.

Сервер дистанційного навчання Київського Інституту Інвестиційного Менеджмента (КІІМ)

http://srv.kimi.kiev.ua/DISTANT/uindex.html

Для дистанційного навчання КІІМ пропонує програми другої вищої освіти "Інвестиційний менеджмент" і "Фінансовий менеджмент":

- Ринок валют FOREX;
- Управління фінансовими ризиками.

Українська Система Дистанційного Навчання — UDL System

http://www.udl.org.ua/

UDL System (м. Львів) — партнерська організація , яка об'єднує вищі навчальні заклади, науково-дослідні інституції, банки, корпорації та неприбуткові організації для створення нової якості за допомогою інноваційного застосування новітніх інформаційних технологій в освіті.

Провайдер: Львівський банківський інститут.

Існують Web-курси Українсько-Канадського "МБЕРІФ - БІЗНЕС ЦЕНТРУ".

Лабораторія Віртуального Дистанційного навчання Харківського технічного університету радіоелектроники (ХТУРЕ)

http://vdll.kture.kharkov.ua/

Напрямки:

- вступ до комп'ютерних наук;
- радіотехніка;
- математика;
- філософія.

Проблемна Лабораторія Дистанційного Навчання Національного технічного університету "Харківський політехнічний інститут"

http://users.kpi.kharkov.ua/lre/

Напрямки:

- українська та російська мови;
- віртуальне навчальне середовище "Веб-клас ХПІ";
- загальна педагогіка;
- вища математика для студентів економічних спеціальностей;
- дистанційні проекти для школярів;
- Web-дизайн дистанційних курсів;
- "Інтернет в освіті";
- "Ризик прийняття управлінських рішень" тощо.

Міжнародний університет фінансів (на базі КПІ)

iuf.ntu-kpi.kiev.ua

Спеціалізується на підготовці фахівців для фінансово-банківської системи України і входить до складу Навчально-методичного комплексу "Національний технічний університет України "Київський політехнічний інститут" — Міжнародний університет фінансів".

Український інститут науково-технічної та економічної інформації. Центр комп'ютерного навчання SEMicom

http://www.semi.com.ua/dist_obuch.html

Напрямок — мови програмування C/C++, Java, Visual C++, SQL, HTML та ін.

Проект "Distance Learning"

http://www.distance-learning.com.ua/

Проект виконує "Центр сприяння об'єднанню світових інформаційних мереж університетів" (UniNet), який займається дослідженням технологій та розробкою програмних продуктів, що призначені для автоматизації процесу дистанційного навчання та підвищення ефективності самостійної роботи студентів. Засновником Центру є Київський національний університет імені Тараса Шевченка.

Український центр дистанційної освіти (УЦДО)

http://udec.ntu-kpi.kiev.ua/

УЦДО — це структурний підрозділ Національного технічного університету України "Київський політехнічний інститут" (НТУУ "КПІ"). Напрямки: інформаційні технології, економічні дисципліни, іноземні мови, технічні дисципліни.

Інформаційно-видавничий центр ІНТЕЛЕКТ+ (м. Львів)

http://www.ipk.polynet.lviv.ua/ipklp/InfVinf.htm

З метою забезпечення навчальними матеріалами слухачів факультету "Менеджмент та підприємництва" розпочато роботу з підготовки до видання експериментальних навчальних посібників з серії "Дистанційне навчання" у відповідності до "Збірника навчальних програм", зокрема:

- "Макроекономіка";
- "Структурно-логічні схеми до мікроекономіки";
- "Мікроекономіка";
- "Ділова українська мова";
- "Історія України";
- "Маркетинг";
- "Виробничий менеджмент";
- "Філософія";
- "Фінансовий менеджмент";
- "Економіка підприємств";
- "Історія економічних вчень";
- "Статистика";
- "Інвестиційний менеджмент";
- "Основи бухгалтерського облік".

Сумський державний університет, кафедра прикладної математики, лабораторія дистанційного навчання

http://dl.sumdu.edu.ua/

Напрямки: методи оптимізації, основи педагогіки, англійська мова, економічне прогнозування, елементи обчислювальної математики, мови програмування, Web-дизайн тощо.