Міністерство освіти і науки України
Запорізький національний університет

В. І. Горбенко, А. О. Лісняк, Є. В. Панасенко

ОРГАНІЗАЦІЯ КОМП'ЮТЕРНИХ МЕРЕЖ

Методичні рекомендації до лабораторних занять
для здобувачів ступеня вищої освіти бакалавра
спеціальності 126 «Інформаційні системи та технології»
освітньо-професійної програми «Інформаційні системи та технології»

Затверджено
вченою радою ЗНУ
Протокол № 6
від 21.12.2021

Запоріжжя
2021

УДК: 004.7(076.5)
Г673

Горбенко В. І., Лісняк А. О., Панасенко Є. В. Організація комп'ютерних мереж: методичні рекомендації до лабораторних занять для здобувачів ступеня вищої освіти бакалавра спеціальності 126 «Інформаційні системи та технології» освітньо-професійної програми «Інформаційні системи та технології». Запоріжжя : ЗНУ, 2021. 71 с.

Методичні рекомендації містять теоретичний матеріал з кожної теми дисципліни «Організація комп'ютерних мереж», хід виконання та приклади виконання лабораторних робіт, які пропонуються при вивченні даного курсу. Кожна лабораторна робота супроводжується необхідними прикладами та поясненнями до них, певними практичними завданнями, виконання яких поглиблює сприйняття матеріалу та контрольними запитаннями.

Методичні рекомендації з дисципліни «Організація комп'ютерних мереж» призначені для студентів спеціальності 126 «Інформаційні системи та технології» освітньо-професійної програми «Інформаційні системи та технології».

Рецензент *С. М. Гребенюк, д. т. н., доцент*
завідувач кафедри фундаментальної та прикладної математики

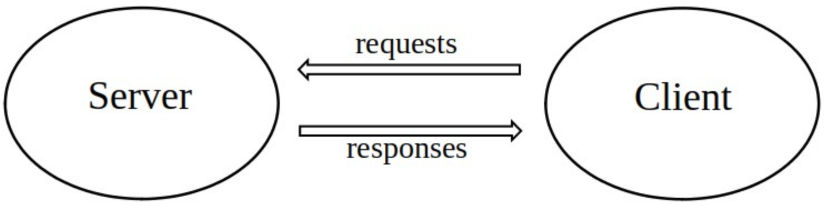
Відповідальний за випуск *А.О. Лісняк, к. ф-м. н., доцент*
завідувач кафедри програмної інженерії

ЗМІСТ

ВСТУП

Станом на початок 2021 року чисельність населення світу становила 7,83 мільярда осіб. У січні 2021 року за самим найскромнішим підрахунком інтернетом (англ. Internet) користувалося 4,66 мільярда людей у всьому світі. Internet є основою мережі (the Web), технічною інфраструктурою, завдяки якій існує Всесвітня Павутина (World Wide Web). За своєю суттю Internet – дуже велика мережа комп’ютерів, які можуть взаємодіяти один з одним. Але слід зазначити, що ні Internet, ні World Wide Web, строго кажучи, не є мережами. Internet – це мережа мереж, а Всесвітня Павутина – це розподілена система на базі Internet. Отже, вивчення інтернет-технологій студентами спеціальності «121 – Інженерія програмного забезпечення» природно розпочати з розгляду основ комп’ютерних мереж.

Комп’ютерні мережі (англ. Network) – це сукупність персональних комп’ютерів, розподілених на деякій території і об’єднаних для спільного використання ресурсів. По суті метою такого об’єднання є надання користувачам доступу до різного роду інформації, баз даних, чатів, документів та програм. Комп’ютери, підключені до мережі, називаються клієнтами та серверами (Client) (Server). Спрощена схема того, як вони взаємодіють, може виглядати так:



При побудові мережі визначається широта охоплення віддалення комп’ютерів, що складають мережу. На сьогодні мережі поділяють за територіальною ознакою:

1. Локальні мережі (LAN – Local Area Network). До локальних мереж зазвичай відносять мережі, комп’ютери яких зосереджені на відносно невеликих територіях. Прикладом локальної мережі є мережа Запорізького національного університету, яка охоплює всі корпуси університету, яка забезпечує високу швидкість обміну інформацією між комп’ютерами.
2. Глобальні мережі (WAN – Wide Area Network). До глобальних відносять мережі, які побудовані на основі використання супутникових та наземних ліній зв’язку розташованих па значній відстані один від одного. Велика протяжність таких мереж впливає на швидкість передачі даних, яка істотно нижче, ніж в локальних мережах.
3. Мережі мегаполісів (MAN – Metropolitan Area Network). Подібні мережі призначені для обслуговування території великого міста-мегаполіса.

Кожна така мережа є частиною деякої глобальної мережі. Для побудови таких мереж використовуються досить якісні цифрові лінії зв'язку, що дозволяють здійснювати взаємодію на відносно високих у порівнянні з глобальними мережами швидкостях.

Метою вивчення навчальної дисципліни «Організація комп'ютерних мереж» є засвоєння систематичних знань із базових технологій сучасних комп'ютерних мереж, систем передачі інформації, методів комутації, стандартів інформаційних та обчислювальних мереж, зокрема аналізу та моделюванню процесів та явищ в галузі інформаційних технологій.

Основними завданнями вивчення дисципліни «Організація комп'ютерних мереж» є:

– виробити навички із принципів побудови та стандартів комп'ютерних мереж;
– ознайомитися із топологією комп'ютерних мереж;
– набути навички із застосування протоколів інформаційного обміну;
– набути навички із використання сучасних технологій комп'ютерних мереж та їх використання у локальних та глобальних мережах;
– набути навички із методів використання комп'ютерних мереж та їх технологій;
– навчитися розробці структури комп'ютерних мереж;
– навчитися використовувати програмні засоби для діагностики та адміністрування комп'ютерних мереж.

Згідно з вимогами освітньо-професійної програми студенти повинні досягти таких результатів навчання (компетентностей): здатність застосовувати знання у практичних ситуаціях; здатність учитися та отримувати сучасні знання; здатність до пошуку, оброблення та аналізу інформації з різних джерел; здатність аналізувати, вибирати і застосовувати методи і засоби для забезпечення інформаційної безпеки (в тому числі кібербезпеки); здатність до алгоритмічного та логічного мислення; знати та вміти застосовувати інформаційні технології обробки, зберігання та передачі даних.

Основою для вивчення дисципліни «Організація комп'ютерних мереж» є опанування студентами дисципліни «Методи та засоби комп'ютерних інформаційних технологій». Набуті при вивченні даного курсу знання необхідні для подальшого вивчення курсу «Технології та протоколи Інтернет» та подальшої дослідницької діяльності в науках про інформаційні технології та інших галузях науки та техніки.

Крім того, у рамках курсу буде приділено увагу інтернет-підключенню, протоколу TCP/IP, протоколам UDP та TCP, системі доменних імен DNS, протоколу передачі гіпертексту HTTP, вивчення структури та вмісту IP-пакетів та маршрутизації в IP-мережах. Методичні вказівки містять рекомендації щодо виконання лабораторних робіт та їх хід виконання. Кожна лабораторна робота супроводжується необхідними прикладами та поясненнями до них, певними практичними завданнями, виконання яких поглиблює сприйняття матеріалу та контрольними запитаннями.

Загальні рекомендації до виконання лабораторних робіт

Лабораторні заняття з дисципліни «Організація комп’ютерних мереж» призначені для студентів освітньо-кваліфікаційного рівня «бакалавр» спеціальності 126 «Інформаційні системи та технології» освітньо-професійної програми «Інформаційні системи та технології».

При виконанні кожної роботи необхідно ознайомитися з теоретичним матеріалом та завданнями до лабораторної роботи. Захист лабораторної роботи відбувається за наступним сценарієм: студент відповідає на поставленні викладачем питання, демонструє оригінальність роботи, пояснює хід виконання лабораторної роботи. Для пояснення деяких результатів та демонстрації набутих навичок студент може скористатись комп’ютером.

Для отримання оцінки з виконання лабораторної роботи студент повинен підготувати звіт та завантажити його до СЕЗН Moodle у встановлені терміни. Звіт з виконання лабораторної роботи повинен мати:

- титульний аркуш;
- тему роботи;
- хід роботи, який позначається у виконанні її завдань;
- результати розрахунків;
- аналіз отриманих результатів виконання завдань;
- скорочені відповіді на запитання для самоперевірки.

Для завантаження до СЕЗН Moodle файл повинен бути у таких форматах, як docx, odt або pdf. Формат pdf є переважним. Якщо розмір файлу перевищує встановлені обмеження для завантаження, скористайтесь будь-яким графічним редактором для зменшення розміру наявних е ньому рисунків. Не використовуйте програмне забезпечення, що використовується для об’єднання файлів у один архівний файл. Звіт, який завантажено у архівному файлі буде вважатись непідготовленим.

ТЕМА 1. ЗАГАЛЬНІ КЛАСИФІКАЦІЇ ТА ПРИНЦИПИ КОМП'ЮТЕРНИХ МЕРЕЖ

Лабораторна робота №1. Мережеві утиліти операційної системи

Тема: Вивчення роботи мережевих утиліт операційної системи.

Мета: Вивчити синтаксис та принципи використання основних мережевих утиліт, які застосовуються в операційних системах Windows та Linux для управління, контролю та зміни мережевих ресурсів та оточення.

Теоретичні відомості

В операційних системах для визначення мережевої конфігурації вузла (персонального комп'ютера, сервера, комутаційного обладнання тощо), його інтерфейсів, а також для налаштування та зміну параметрів мережі використовують програмне забезпечення – утиліти. З розвитком операційних систем змінювалась і функціональність утиліт, стало можливим отримати деякі параметри або виконати одні і ті самі налаштування за допомогою різних утиліт, з'явились утиліти з гіпернабором можливостей, наприклад, ip-пакет в Unix системах. Спектр можливостей утиліт та їх різноманіття у різних версіях та клонах операційних систем потребує інтенсивного самостійного вивчення як для тих, хто буде адмініструвати комп'ютерні мережі, так і для тих, хто буде створювати програмне забезпечення, що їх використовує. Тому в лабораторній роботі розглядається певний базовий набір утиліт та деякі приклади їх використання, що дозволяють отримати інформацію про наявну мережеву конфігурацію вузла та його оточення у мережі.

Утиліта ifconfig (операційні системи Linux та UNIX) використовується для отримання інформації по мережевим інтерфейсам комп'ютера та конфігурування їх параметрів. Також вона може використовуватись для перевизначення адреси мережевого інтерфейсу. В операційних системах Windows подібну функцію має утиліта `ipconfig`.

Якщо не вказуються опції, то утиліта `ifconfig` виводить поточну конфігурацію мережевих інтерфейсів. Якщо необхідно вивести мережеву конфігурацію певного інтерфейсу, то в команді вказується його ім'я. Наприклад, для wi-fi адаптеру за командою `ifconfig wlo1` було отримано:

```
wlo1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.165  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 fe80::667f:65e2:8ebb:5fcb  prefixlen 64  scopeid 0x20<link>
        ether 24:77:03:51:bd:e8  txqueuelen 1000  (Ethernet)
        RX packets 12832  bytes 9166196 (9.1 MB)
        RX errors 0  dropped 1  overruns 0  frame 0
        TX packets 6060  bytes 807819 (807.8 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

де wlo1 вказує на інтерфейс; flags вказує на прапорці, що встановлені для інтерфейсу – UP (увімкнено) BROADCAST (може приймати пакети широкомовлення) RUNNING (має підключення до мережі) MULTICAST (може приймати групові пакети) PROMISC (працює у нерозбірливому режимі); mtu – розмір максимального блоку; inet – IP-адреса інтерфейсу; netmask – мережева маска або маска підмережі; broadcast – IP-адреса для пакетів широкомовлення; inet6 – IP-адреса версії 6; prefixlen – довжина префіксу для IP версії 6; scopeid – вказує на зв’язок із пристроєм; ether – MAC-адреса пристрою; txqueuelen – максимальна довжина черги; RX – вказує на кількість прийнятих пакетів та байт; TX – вказує на кількість переданих пакетів та байт.

Утиліту `ping` призначено для тестування зв’язку у комп’ютерній мережі. Утиліта генерує, так званні, пакети відлуння та відправляє їх до вузла призначення. Якщо вузол в мережі існує, він у робочому стані та його налаштування працездатні, то у відповідь прийде відповідь на відлуння. Утиліта `ping` має певну кількість різних аргументів та опцій, що також дозволяє її використовувати до діагностики зв’язку. Корисно застосовувати опцію `-c` для задавання кількості пакетів відлуння, що генерується (за замовчуванням, пакети генеруються безперервно, поки не буде натиснуто одночасно клавіші Ctrl-C). Опція `-i` дозволяє задавати інтервал часу між пакетами (за замовчуванням 1 секунда). Опція `-I` дозволяє задати конкретний інтерфейс, із якого будуть відправлятись пакети (має сенс, якщо в системі декілька мережевих інтерфейсів). `-S` дозволяє задати деяку конкретну IP адресу джерела пакетів відлуння. Опція `-f` використовується системними адміністраторами для одночасного відправлення множини пакетів для перевірки стабільності роботи інтерфейсу (як правило використовується разом з опцією `-c`). Нижче наведено простий приклад використання утиліти `ping`:

```
vgorbenko@vgorbenko-HP-EliteBook-8460p:~$ ping 10.1.100.31
PING 10.1.100.31 (10.1.100.31) 56(84) bytes of data.
64 bytes from 10.1.100.31: icmp_seq=1 ttl=62 time=12.8 ms
64 bytes from 10.1.100.31: icmp_seq=2 ttl=62 time=9.30 ms
64 bytes from 10.1.100.31: icmp_seq=3 ttl=62 time=3.06 ms
^C
--- 10.1.100.31 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 3.060/8.411/12.871/4.055 ms
```

У команді використано певну IP-адресу 10.1.100.31 локальної мережі. Кожний рядок відлуння вказує на довжину пакету відлуння – 64 байти, номер пакету у послідовності, значення TTL для прийнятого пакету та час, що пройшов між відправленням та прийняттям пакету.

Утиліта `route` використовується для управління таблицею маршрутизації. Таблиця маршрутизації дозволяє встановлювати шляхи відправлення пакетів як до сегментів локальної мережі, так і до певних мереж глобальної мережі Інтернет, а також встановлювати пріоритетність використання інтерфейсів та шляхів при пересиланні пакетів. У найпростішому випадку утиліта `route` використовується без опцій і дозволяє отримати наявну таблицю маршрутизації:

```
vgorbenko@vgorbenko-HP-EliteBook-8460p:~$ route
Таблица маршутизации ядра протокола IP
Destination Gateway Genmask Flags Metric Ref Use Iface
default         router.asus.com 0.0.0.0         UG    600    0        0 wlo1
link-local      0.0.0.0         255.255.0.0     U     1000   0        0 wlo1
192.168.1.0     0.0.0.0         255.255.255.0   U     600    0        0 wlo1
192.168.122.0   0.0.0.0         255.255.255.0   U     0      0        0 virbr0
```

Усю інформацію розбито на декілька стовпчиків, що залежить від використаних опцій. Стовпчик Destination вміщує певні елементи мережі – вузли, мережеві сегменти, інтерфейси. Стовпчик Gateway вміщує адресу шлюзу для пересилання пакетів (0.0.0.0 – за замовчуванням). Стовпчик Genmask – маски підмереж, що вказують на об'єм елементу мережі. У стовпчику Flags прапорець U вказує, що шлях активний, а прапорець G, що цей елемент мережі є основним шлюзом. Metric вказує на пріоритетність маршруту – найменше значення вказує на найвищий пріоритет. Ref вказує на кількість посилань на певний маршрут, а Use – на інтенсивність його використання. Iface вказує на зв'язок з певним інтерфейсом.

Утиліту `traceroute` часто використовують для визначення маршруту від одного вузла комп'ютерної мережі до іншого. Для цього створюються ICMP-пакети з адресою вузла призначення. Суттєвим є те, що створювані пакети мають послідовно збільшуваний за значенням параметр TTL (time to live). Цей параметр зазвичай вказує на максимальну кількість вузлів-маршрутизаторів, що дозволяється пройти пакету. Будь який вузол мережі, коли отримує пакет, що має бути переправлений на інший вузол, обов'язково зменшує значення TTL на 1. Якщо TTL стає рівним 0, то такий пакет далі не пересилається, а на вузол, що його відправив, повертається ICMP-повідомлення «time exceeded in transit». Утиліта traceroute фіксує адрес цього маршрутизатора, а також проміжок часу між відправленням пакету та отриманням відповіді. Саме ця інформація і виводиться на термінал. Після цього traceroute повторює створення і відправлення пакетів, але значення TTL збільшується на 1, що дозволяє "проскочити" вузли-маршрутизатори, які вже відповіли, і досягти наступний за маршрутом. Якщо послідовно відправляти на вузол-призначення пакети, що відрізняються значенням TTL на 1, то отримаємо пакети з ICMP-повідомленням «time exceeded in transit» від усіх проміжних вузлів-маршрутизаторів.

```
vgorbenko@vgorbenko-HP-EliteBook-8460p:~$ traceroute www.yahoo.com
traceroute to www.yahoo.com (87.248.98.8), 30 hops max, 60 byte packets
 1  router.asus.com (192.168.1.1)  2.954 ms  2.890 ms  2.841 ms
 2  herr.znu.edu.ua (10.1.10.1)  2.797 ms  2.767 ms  2.992 ms
 3  212.111.202.5 (212.111.202.5)  5.937 ms  5.922 ms  9.064 ms
 4  fe0-1-701.zpr0.uran.ua (212.111.192.233)  15.632 ms  15.639 ms  15.594 ms
 5  ae2-236.RT.NTL.KIV.UA.retn.net (87.245.237.16)  15.546 ms  15.508 ms  15.466 ms
 6  ae2-7.RT.IRX.VIE.AT.retn.net (87.245.233.137)  47.133 ms  40.517 ms  34.958 ms
 7  193.203.0.242 (193.203.0.242)  34.980 ms 193.203.0.42 (193.203.0.42)  34.958 ms  34.935 ms
 8  UNKNOWN-188-125-89-X.yahoo.com (188.125.89.53)  53.516 ms  53.498 ms  53.480 ms
 9  UNKNOWN-188-125-89-X.yahoo.com (188.125.89.53)  53.370 ms  53.754 ms xe-4-2-0.pat1.tc2.yahoo.com (66.196.65.210)  53.643 ms
10  ge-4-2-0.pat1.the.yahoo.com (66.196.65.208)  53.683 ms  53.677 ms UNKNOWN-66-196-65-X.yahoo.com (66.196.65.217)  69.698 ms
11  UNKNOWN-66-196-65-X.yahoo.com (66.196.65.217)  69.751 ms  78.933 ms  78.896 ms
12  eth-2-5.bas1-1-prd.ir2.yahoo.com (217.146.186.79)  78.825 ms et-1-1-0.msr1.ir2.yahoo.com (66.196.65.19)  82.150 ms  70.400 ms
13  eth-1-5.bas1-1-prd.ir2.yahoo.com (217.146.185.176)  79.827 ms media-router-fp2.prod1.media.vip.ir2.yahoo.com (87.248.98.8)  66.214 ms
```

На кінцевому вузлі пакет з TTL=1 не відкидається та ICMP-повідомленням «time exceeded in transit» не створюється. Для визначення того, що вузол-призначення був досягнутий, усі пакети відправляються на порт, який не використовується. Його номер дорівнює 33434 + (максимальна кількість транзитних вузлів) – 1. Після отримання такого пакету вузол-призначення повертає ICMP-повідомлення про помилку «порт недоступний». Саме це дозволяє визначити, що вузол-призначення досягнуто.

Команда `netstat` входить до стандартного набору мережевих утиліт операційних систем Windows, Linux, UNIX. За її допомогою можна отримати інформацію про підключення до комп’ютерної мережі, статистику відповідно до кожного інтерфейсу, таблиці маршрутизації, masquerade, multicast та інше.

Визначення активних підключень до комп’ютерної мережі виконується наступним чином:

команда `netstat -a` дозволяє визначити всі підключення

```
# netstat -a | more
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 localhost:domain        *:*                     LISTEN
udp6       0      0 fe80::20c:29ff:fe68:ntp [::]:*

Active UNIX domain sockets (servers and established)
Proto RefCnt Flags       Type       State         I-Node   Path
unix  2      [ ACC ]     STREAM     LISTENING     20492    /var/run/mysqld/mysqld.sock
unix  2      [ ACC ]     STREAM     LISTENING     23323    /var/run/php5-fpm.sock
```

команда `netstat -at` дозволяє визначити TCP підключення

```
# netstat -at
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 localhost:domain        *:*                     LISTEN
tcp        0      0 *:ssh                   *:*                     LISTEN
tcp        0      0 localhost:ipp           *:*                     LISTEN
tcp        0      0 *:http                  *:*                     LISTEN
```

команда `netstat -au` дозволяє визначити UDP підключення

```
# netstat -au
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
udp        0      0 localhost:domain        *:*
udp        0      0 *:bootpc                *:*
udp6       0      0 fe80::20c:29ff:fe68:ntp [::]:*
```

Для визначення портів, що знаходяться у стані прослуховування, використовується ключ -l. Наприклад: команда `netstat -lt` використовується для визначення TCP-портів, що знаходяться у стані прослуховування.

Для отримання статистики по кожному протоколу використовується команда `netstat -s`, наприклад:

```
# netstat -s
Ip:
    11150 total packets received
    1 with invalid addresses
    0 forwarded
    0 incoming packets discarded
    11149 incoming packets delivered
    11635 requests sent out
Icmp:
    13791 ICMP messages received
    12 input ICMP message failed.
Tcp:
    15020 active connections openings
    97955 passive connection openings
    135 failed connection attempts
```

```
Udp:
    2841 packets received
    180 packets to unknown port received
......
```

Для отримання PID та імені процесу за форматом «PID/Program Name» `netstat` використовується з опцією -p, яку можна поєднувати з іншими опціями. Як правило вона використовується при налагоджені для визначення того, яка програма використовує який порт. Наприклад:

```
# netstat -pt
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 org-ru-putty.vm.udf:www 52-106.plus.kerch:55723 ESTABLISHED 9486/nginx: worker
tcp        0      0 org-ru-putty.vm.udf:www 52-106.plus.kerch:55757 ESTABLISHED 9486/nginx: worker
```

Для безперервного виводу інформації `netstat` використовується опція -c. Наприклад:

```
# netstat -c
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 org-ru-putty.vm.udf:www 182.131.74.202:59933    FIN_WAIT2
tcp        0      0 org-ru-putty.vm.udf:www 182.131.74.202:63761    FIN_WAIT2
tcp        0      0 org-ru-putty.vm.udf:www 92-181-66-102-irk.:4585 ESTABLISHED
^C
```

Перервати вивід можна одночасним натисканням CTRL-C.

Список мережевих інтерфейсів отримується за допомогою команди `netstat -i`, а більш розширена інформація про інтерфейси отримується за допомогою: `netstat -ie`. Наприклад:

```
# netstat -ie
Kernel Interface table
eth0      Link encap:Ethernet  HWaddr 00:0c:29:68:4c:a4
          inet addr:192.168.128.134  Bcast:192.168.128.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe68:4ca4/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:24278 errors:0 dropped:0 overruns:0 frame:0
          TX packets:11275 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:33203025 (33.2 MB)  TX bytes:665822 (665.8 KB)
          Interrupt:19 Base address:0x2000
```

Взагалі можна використати набір із декількох опцій для отримання певної визначеної інформації:

```
# netstat -lnptux
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN      9614/nginx
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      601/sshd
udp        0      0 8.8.4.4:123             0.0.0.0:*                           574/ntpd
udp        0      0 127.0.0.1:123           0.0.0.0:*                           574/ntpd
udp        0      0 0.0.0.0:123             0.0.0.0:*                           574/ntpd
Active UNIX domain sockets (only servers)
Proto RefCnt Flags     Type       State     I-Node   PID/Program name  Path
unix  2      [ ACC ]  STREAM   LISTENING  4233     826/python        /var/run/fail2ban/fail2ban.sock
unix  2      [ ACC ]  STREAM   LISTENING  8122     2561/mysqld       /var/run/mysqld/mysqld.sock
unix  2      [ ACC ]  STREAM   LISTENING  160413   7301/php-fpm.conf /var/run/php5-fpm.sock
```

Універсальною консольною утилітою є `ip`, реалізація якої доступна в операційних системах клону Linux та UNIX. Утиліта `ip` об'єднує можливості `ifconfig`, `arp`, `route` та деяких інших утиліт та команд для управління мережею Linux.

Утиліта `ip` має значну функціональність та дозволяє визначати властивості підключення до комп'ютерної мережі, IP-адреси інтерфейсів та `arp` таблицю, налаштовувати політики маршрутизації, змінювати записи arp таблиці та деякі параметри стеку TCP/IP. Значна кількість функцій потребує дещо специфічного її використання. Синтаксис командного рядку цієї утиліти виглядає наступним чином:

`ip` [опції] об'єкт команда [параметри]

Опції є глобальними налаштуваннями, що впливають на роботу всієї утиліти незалежно від інших аргументів та можуть не задаватись. Об'єктом у командному рядку виступає об'єкт або пристрій, для якого виконується команда. Командою є певна дія з об'єктом, якій можуть надаватись деякі параметри.

Основними опціями є: -V (отримати інформацію про утиліту та її версію); -h (виводити інформацію у зручному для людини форматі); -b (читати команди із наданого файлу); -s (виводити статистичну інформацію); -f (специфікує сімейство протоколів, що буде використовуватись); -o (вказує на вивід кожного запису з нового рядку); -r (вказує на необхідність використання резолверу системних імен).

Об'єктами у командному рядку утиліти `ip` можуть бути: address (мережева адреса інтерфейсу); link (фізичний мережевий пристрій); monitor (моніторинг стану пристроїв); neighbour (управління ARP); route (управління маршрутизацією); rule (правила маршрутизації); tunnel (тунелювання IP), а також інші.

Під час введення ім'я об'єкту може бути скорочено до однієї або до кількох літер, у залежності від однозначності. Наприклад, команду `ip address show` можна записати як `ip a show`.

Серед типових команд, що можуть виконуватись над об'єктами є: add, change, del або delete, flush, get, list або show, monitor, replace, restore, save, set, а також update. Для отримання переліку команд, що виконуються для певного об'єкту, слід задати: `ip` об'єкт `help`.

Наприклад, `ip link help` виводить довідку за командами для об'єкту link.

Якщо команду не задано, то за замовчуванням виконується команда `show`. Також підтримується скорочення і у більшості випадків для виконання потрібної дії достатньо декількох символів. Слід пам'ятати, що алфавітний порядок не завжди підтримується, наприклад, `ip a s` — означає `ip address show`, а не `ip address set`.

Параметри залежать від об’єкта та вказаної команди. Найчастіше використовуються наступні: dev ім’я_пристрою; up (увімкнути); down (вимкнути); llarr MAC-адреса; initcwnd розмір вікна перевантаження TCP при ініціалізації; window розмір вікна TCP; cwnd розмір вікна перевантаження TCP; type тип; via (підключитись до роутеру); default маршрут за замовчуванням; blackhole — маршрут "чорна дірка" (відкидає пакети і не посилає ICMP-повідомлення про недоступність); prohibit - маршрут "заборони" (відкидати пакети та повертати ICMP повідомлення про заборону доступу); unreachable - маршрут "недосяжний" (відкидати пакети та відправляти ICMP пакети про недосяжність вузла).

Нижче наведемо декілька прикладів використання утиліти `ip`.

ip link show – показує стан усіх мережевих інтерфейсів.

ip link show eth0 – показує стан інтерфейсу з ім'ям eth0.

ip link list up – показує стан усіх увімкнених мережевих інтерфейсів.

ip link set eth1 up – включає мережевий інтерфейс з ім'ям eth1.

ip link set eth1 down – виключає eth1.

ip neigh show – показує усі записи ARP

ip nei add 1.1.1.13 lladdr AA:BB:CC:DD:EE:FF dev eth0 – додає до ARP-таблиці запис для певної IP адреси, де 1.1.1.13 – IP-адреса, AA:BB:CC:DD:EE:FF – MAC-адреса, eth0 — мережевий пристрій.

ip address show – показати усі IP-адреси та їх інтерфейси

ip a l permanent – показати тільки статичні IP-адреси

ip a l dynamic – показати тільки динамічні IP-адреси

ip addr add 1.1.1.13/24 dev eth0 – задати IP-адресу для інтерфейсу eth0

ip addr del 1.1.1.13/24 dev eth0 – видалити IP-адресу інтерфейсу eth0

ip add flush dev eth0 – видалити усі IP-адреси інтерфейсу eth0

Із точки зору адміністрування комп'ютерних мереж важливою функцією утиліти `ip` є можливість налаштовувати мережеві маршрути. Таблиці маршрутизації для утілити ip ідентифікуються за номером (від 1 до 255). За замовчуванням використовується таблиця маршрутизації 254. Для відображення усіх маршрутів таблиці маршрутизації використовується командний рядок: `ip r sh`:

```
(base) vitaliy@vitaliy-HP-EliteBook-8460p:~$ ip route show
default dev ppp0 proto static scope link metric 50
default via 192.168.0.1 dev wlo1 proto dhcp metric 600
10.1.100.153 dev ppp0 proto kernel scope link src 10.1.100.228 metric 50
169.254.0.0/16 dev wlo1 scope link metric 1000
172.17.0.0/16 dev docker0 proto kernel scope link src 172.17.0.1 linkdown
192.168.0.0/24 dev wlo1 proto kernel scope link src 192.168.0.111 metric 600
192.168.0.1 dev wlo1 proto static scope link metric 600
212.111.202.6 via 192.168.0.1 dev wlo1 src 192.168.0.111
212.111.202.6 via 192.168.0.1 dev wlo1 proto static metric 600
```

ip route show table 255 — показує усі маршрути з таблиці 255 ip route get 10.10.20.0/24 – показує маршрут до цієї підмережі

ip route get 10.10.20.0/24 from 192.168.12.9 – показує маршрут до заданої підмережі від вказаного інтерфейсу

ip route add 10.10.20.0/24 via 192.168.50.100 – створює маршрут до заданої підмережі через інтерфейс за вказаною IP-адресою

ip route delete 10.10.20.0/24 – видаляє маршрут до підмережі

ip route del 10.10.20.0/24 via 192.168.50.100 – видаляє маршрут до підмережі через заданий інтерфейс

ip route add default via 192.168.50.100 – створює маршрут за замовчуванням

ip route add 10.10.20.0/24 dev eth0 – створює маршрут до вказаної підмережі через вказаний інтерфейс

ip route add table nnn 10.10.20.0/24 dev eth0 – додає до певної таблиці маршрутизації вказаний маршрут

ip route add blackhole 10.10.20.0/24 dev eth0 – створює маршрут типу blackhole

ip route add unreachable 10.10.20.0/24 dev eth0 – створює маршрут типу unreachable

Завдання до лабораторної роботи №1

1. Для виконання роботи використовуйте програми “Командний рядок” (ОС Windows) та “Термінал” (ОС Linux).
2. Вивчить призначення, синтаксис та опції (ключі) наступних мережевих утиліт:
   – `hostname`;
   – `ipconfig` (для Windows) та `ifconfig` (для Linux);
   – `ping`;
   – `arp`;
   – `netstat`;
   – `route`;
   – tracert (для Windows) та traceroute (для Linux).
3. За допомогою `hostname` визначте ім’я комп’ютера.
4. Використовуйте `ipconfig` (для Windows) та `ifconfig` (для Linux) для визначення мережевих інтерфейсів комп’ютера, за яким виконується робота. З отриманої інформації визначте IP-адресу, мережеву маску, адресу широкомовлення, MAC-адресу та кількість прийнятої та переданої інформації.
5. Використайте команду `ping` для наступних адрес:
   – 10.1.100.31 (в мережі ЗНУ)
   – www.znu.edu.ua
   – www.ukr.net

6. Виконайте команду `arp` без задавання опцій та окремо з опціями -a та -n. Порівняйте отримані результати.
7. Для виконання команди netstat використовуйте наступні опції: -r, -i, -g, -s, -v, -p, -l.
8. Виконайте команду `route` без задавання опцій та з опцією -n.
9. Виконайте команди `tracert` (для Windows) та `traceroute` (для Linux) для наступних адрес:
   - 10.1.100.31 (в мережі ЗНУ)
   - www.znu.edu.ua
   - www.ukr.net
10. Використовуючи утиліту `ip` виконайте завдання пунктів 4, 6, 8 для комп'ютера, що працює під операційною системою Linux. Порівняйте з отриманими результатами за допомогою утиліт, вказаних у цих пунктах.
11. Підготуйте звіт, в який занесіть результати за пунктами 3-10.

Запитання для самоперевірки

1. Що таке утиліти?
2. Для чого використовуються мережеві утиліти?
3. За допомогою яких утиліт можна визначити IP-адресу комп'ютера?
4. Як можна визначити існування (наявність підключення до мережі) комп'ютера з певною IP-адресою?
5. Що таке TTL?
6. Для чого призначено таблицю маршрутизації?
7. Яку інформацію містить ARP-таблиця?
8. Як визначити маршрут (проміжні вузли-маршрутизатори) до певного вузлу-призначення?
9. Що дозволяє визначити утиліта netstat?
10. Який загальний синтаксис командного рядку використання утиліти ip?

ТЕМА 2. ТЕХНОЛОГІЇ ЛОКАЛЬНИХ МЕРЕЖ

Лабораторна робота №2. Аналізатор мережевих пакетів Wireshark

Тема: Аналізатор мережевих пакетів Wireshark. Захват та аналіз мережевих кадрів.

Мета: За допомогою аналізатора вивчити структуру та типи кадрів у комп’ютерній мережі

Теоретична частина

Аналізатори мережевих протоколів (англ. – sniffer) відносяться до категорії засобів, які є необхідними адміністратору мережі для визначення та аналізу подій, що відбуваються в мережі. Робота аналізаторів базується на використанні так званого "нерозбірливого" (promiscuous) режиму роботи адаптеру мережевого інтерфейсу. У цьому режимі кадри, які пересилаються мережею, буферизуються мережевим адаптером і, незалежно від MAC-адреси призначення, аналізуються. Крім цієї основної функції аналізатори збирають та визначають у реальному часі завантаження мережі та окремих робочих станцій, ведуть статистику використання мережевих протоколів, визначають розподіл кадрів та пакетів за розміром тощо.

Одним з найпопулярніших аналізаторів кадрів та пакетів у комп’ютерній мережі є Wireshark. Аналізатор Wireshark дозволяє виконувати захват кадрів з мережі, до якої комп’ютер підключено та проводити детальний аналіз його вмісту. Як правило аналізатор використовується:

- для рішення проблем із мережею;
- для оцінки рівня мережної безпеки;
- для вивчення мережевих протоколів та виявлення хибного їх використання;
- для діагностування роботи мережевого програмного забезпечення, що розробляється;
- для виявлення типів протоколів, що використовуються в локальній мережі;
- для виявлення прихованого мережевого трафіку.

Основними відмінностями Wireshark є:

- кросплатформна реалізація;
- захоплення кадрів та пакетів, що надходять до мережевого адаптеру;
- повна деталізація інформації про протоколи, що використано у пакетах;
- збереження даних пакету, який було захоплено, та можливість подальшого аналізу цих даних;
- імпортування та експортування пакетів у різні відомі формати для інших програм-аналізаторів;
- фільтрування пакетів за багатьма критеріями;
- пошук пакетів за багатьма критеріями;
- забарвлення рядків із захопленими пакетами відповідно до фільтрів;

- створення різноманітних статистик;
- та інше.

Захоплення кадрів можна виконати через меню Capture–Options або за допомогою команд Capture-Start та Capture-Stop. Через вкладку Options можна зробити додаткові налаштування, щодо процесу захоплення та показу захоплених даних під час його виконання.

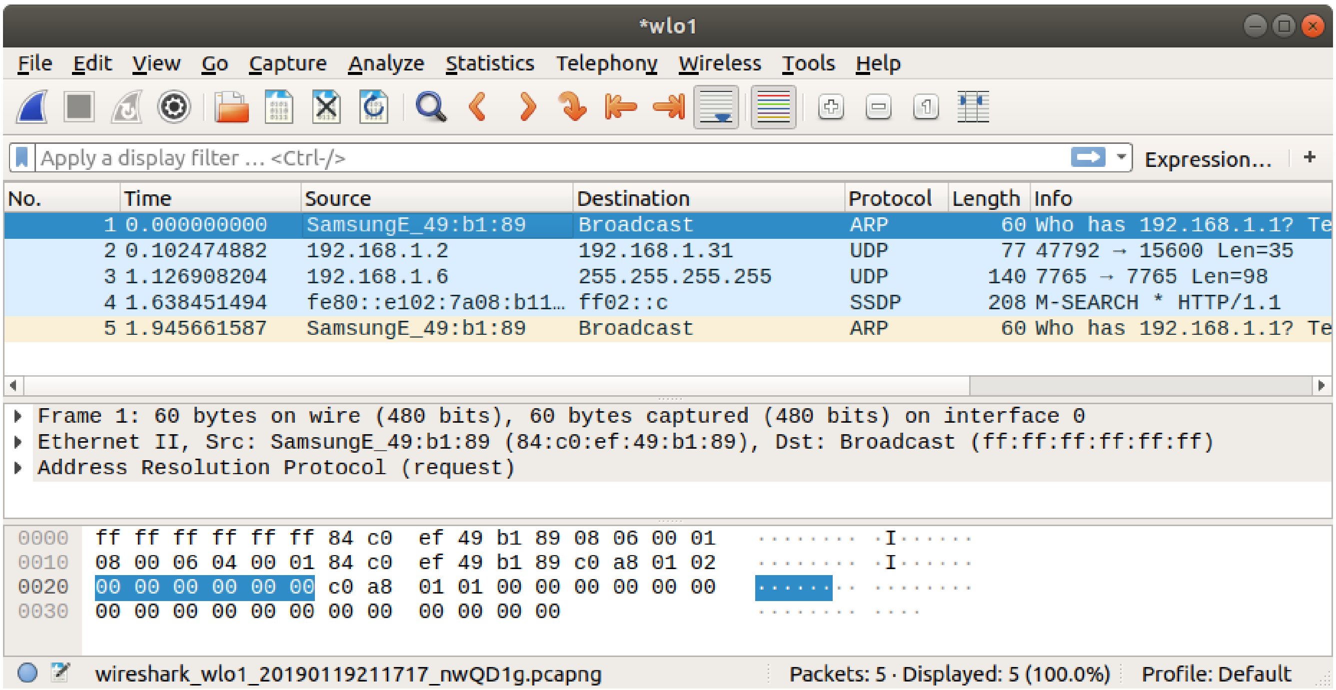


Рис. 1. Wireshark дозволяє захопити мережевий кадр та виконати аналіз його вмісту

У верхній частині вікна аналізатору Wireshark показуються захоплені кадри з мережі з вказівкою їх основних параметрів: мережевої адреси відправника та отримувача, типу кадру, довжини у байтах, часу отримання після початку процесу захоплення. У центральній частині показується ієрархічна структура мережевих протоколів, що використовуються на відповідних рівнях передавання даних. Якщо розгорнути певний рівень, то можна побачити опис полів його заголовку. У нижній частині вікна - шістнадцятирічний дамп кадру (байти кадру, які представлено у шістнадцятирічному форматі).

Адреса відправника та отримувача у кадрі Ethernet зазвичай мають довжину 6 байтів. У MAC-адресі молодший біт першого байту є ознакою індивідуальної або групової адреси: 0 - вказує на певну станцію, 1 – вказує на групову адресу декількох станцій мережі. При широкомовній адресації усі біти адреси встановлюються в 1. Адреса отримувача може бути як індивідуальною, так і груповою або широкомовною. Відправник може мати лише індивідуальну адресу, тобто молодший біт першого байта його адреси завжди має значення 0. Другий біт вказує на унікальність адреси у глобальному сенсі: 0 – унікальний глобально; 1 — унікальний у межах локальної мережі. Перші 3 байти (за порядком їх передавання у мережі та традиційному запису MAC-адреси) є унікальним ідентифікатором організації (OUI), які реєстраційна адміністрація IEEE надає виробникам мережевого обладнання. Останні 3 байти (на Рис.1 це 4, 5 та 6 байти) призначаються виробником для кожного виготовленого екземпляру пристрою і тому забезпечують повну унікальність усієї MAC-адреси.

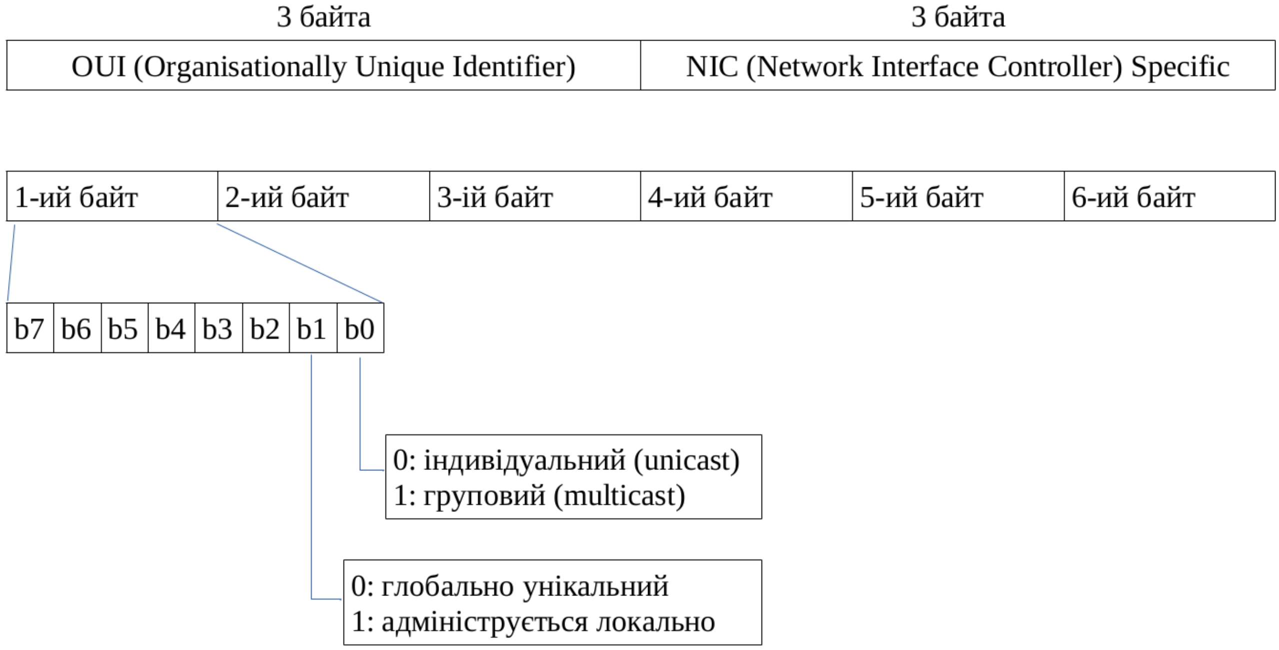


Рис.2. Структура MAC-адреси у канонічному форматі

На практиці у комп’ютерній мережі на канальному рівні зустрічаються 4 типи кадрів Ethernet (Рис.2), що обумовлено тривалою історією розвитку технологій до прийняття стандартів IEEE 802. Практично усе мережеве обладнання вміє може працювати з цими форматами кадрів.

Кадр **Ethernet DIX** (або кадр Ethernet II) визначає наступні поля заголовку:

− адреса отримувача - 6 байтів (MAC-адреса отримувача);

− адреса відправника - 6 байтів (MAC-адреса мережевого інтерфейсу, з якого відправлено кадр);

− тип протоколу (поле Type), що вказує на мережевий протокол верхнього рівня, який використовує поле даних кадру для передачі свого пакету. Для позначення типу протоколу використовуються значення, що перевищують значення максимальної довжини поля даних кадру (1500 байт) для забезпечення відмінності кадрів Ethernet DIX від інших типів;

− кадр може вміщувати не більше 1500 байт даних, але якщо даних передається менш 46 байт, то поле даних заповнюється байтами з 0 значенням до мінімально припустимої довжини. Загальна довжина мінімального за розміром кадру складає 7+1+6+6+2+46+4=72 байта = 576 біт. Такий мінімальний розмір був вимушений для забезпечення коректної роботи механізму виявлення колізій на початку розвитку технологій Ethernet;

− поле контрольної суми (4 байти) має значення, що обчислюється за алгоритмом CRC-32 та призначене для виявлення цільності кадру на стороні отримувача. Якщо значення цього поля переданого кадру не буде збігатись з обчисленим на стороні отримувача, то такий кадр буде проігноровано, якщо не задано іншого механізму.

Кадр 802.3/LLC (або кадр 802.3/802.2) має заголовок, який є результатом поєднання заголовків кадрів, визначених стандартами 802.3 та 802.2. Відповідно до стандарту 802.2 в поле даних кадру 802.3 (MAC-підревнь) вкладається кадр підрівня LLC з видаленими прапорцями початку та кінця кадру. Поля DSAP та SSAP дозволяють вказати на сервіс верхнього рівня, що пересилає данні за допомогою цього кадру. Зазвичай ці поля мають однакові значення. Поле управління використовується для позначення типу кадру даних – інформаційний, кадр управління або ненумерований (зазвичай в Ethernet використовуються ненумеровані кадри). На відмінність від формату Ethernet DIX двобайтове поле довжини вказує довжину поля даних, яке не може перевищувати 1500 байтів. Тому що кадр LLC додає до заголовку 3 байти, то максимальний розмір поля даних зменшується до 1497 байт.

Кадр 802.3

6	6	2	46-1500	4
DA	SA	L	Data	FCS

Кадр 802.3/LLC

6	6	2	1	1	1	43-1497	4
DA	SA	L	DSAP	SSAP	Control	Data	FCS
			Заголовок LLC				

Кадр Ethernet DIX (II)

6	6	2	46-1500	4
DA	SA	T	Data	FCS

Кадр Ethernet SNAP

6	6	2	1	1	1	3	2	38-1492	4
DA	SA	L	DSAP	SSAP	Control	OUI	T	Data	FCS
			AAh	AAh	03h	000000			
			Заголовок LLC			Заголовок SNAP			

Рис. 3. Формати кадрів Ethernet. DA – MAC-адреса отримувача, SA – MAC-адреса відправника, T – поле, що вказує тип протоколу верхнього рівня, L – довжина полю даних, FCS – поле контрольної суми (CRC-код).

Кадр **Raw 802.3** (або кадр Novell 802.3). У ранніх версіях операційної системи NetWare її розробник фірма Novell проігнорувала вимогу щодо розміщення LLC кадру у кадр 802.3. Безпосередньо за заголовком кадру 802.3 вони розташовували IPX-пакет, який починається з FFFFh ідентифікатору кадру цього типу. Novell тривалий час не використовувала службові поля кадру LLC у своїй операційній системі NetWare, тому що в цьому не було необхідності – у полі даних знаходився пакет протоколу IPX. Операційна система NetWare сьогодні практично не використовується, фірма Novell вже втратила свою самостійність, а підтримка протоколу IPX у світі майже не здійснюється.

Кадр **Ethernet SNAP** (SNAP – SubNetwork Access Protocol, протокол доступу до підмереж). Кадр Ethernet SNAP визначено стандартом 802.2H і він є розширенням кадру 802.3/LLC завдяки введенню додаткового поля ідентифікатора організації OUI, яке також може використовуватись для обмеження доступу до мережі комп’ютерів інших організацій. Додатковий заголовок SNAP використовується для надання більшої впорядкованості при позначені типу протоколу, який розміщує свою інформацію у полі даних кадру LLC. Стандарт 802.2 використовує для цього однобайтові поля DSAP та SSAP (максимально 256 протоколів). У версії протоколу Ethernet, яку запропонували разом компанії Digital, Intel та Xerox (версія Ethernet DIX), для позначення типу протоколу у полі даних кадру використовується двобайтове поле Type. Для позначення протоколів мережевого рівня використовуються значення вище 05DCh (шістнадцятковий формат), наприклад, 0800 використовується для позначення протоколу IP. Заголовок SNAP також вміщує двобайтове поле Type, призначення та формат якого такі самі як і у поля Type кадру Ethernet DIX. Трьохбайтовий код організації (OUI) використовується для позначення тієї організації по стандартизації, яка відповідає за числові значення поля Type. Так, числові значення поля Type для заголовку SNAP у випадку використання його у кадрі Ethernet визначає комітет 802.3 IEEE, код якого 00 00 00. Для інших протоколів канального рівня значення кодів поля Type визначають інші організації по стандартизації. Таким чином, при використанні додаткового заголовку SNAP досягається сумісність кадрів 802.3 з кадрами Ethernet DIX за методом кодування пакетів протоколів верхнього рівня, які передаються у полі даних. Поля DSAP та SSAP, при використанні заголовку SNAP, отримують значення $170_{(10)}$ = AAh, що вказує на те, що у полі даних кадру LLC знаходиться заголовок SNAP. Поле управління зазвичай має значення 03h, яке вказує на відсутність попереднього з’єднання на канальному рівні.

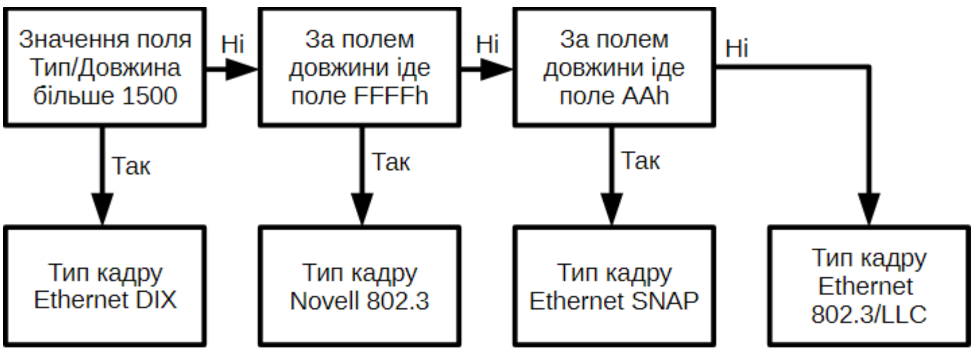
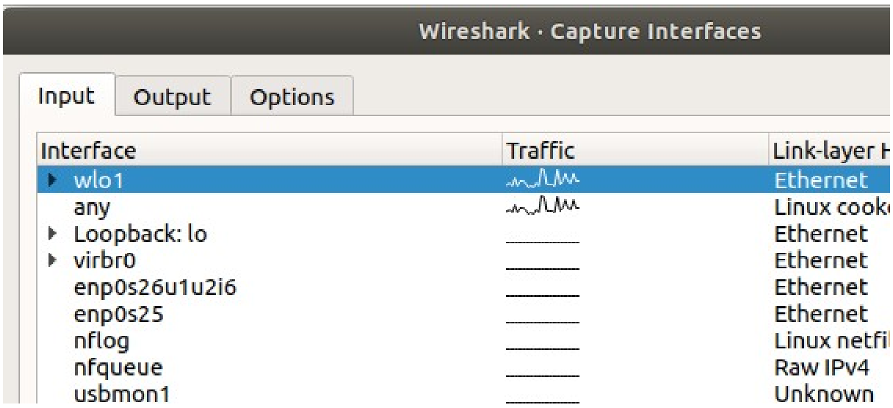


Рис. 4. Алгоритм визначення типу кадру Ethernet

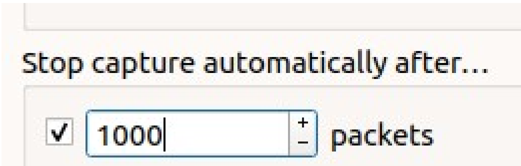
Сучасне мережеве обладнання підтримує усі 4 типи кадрів Ethernet.

Завдання до лабораторної роботи №2

1. Запустіть аналізатор Wireshark. Через панель меню оберіть (Capture → Options) або натисніть Ctrl+K.



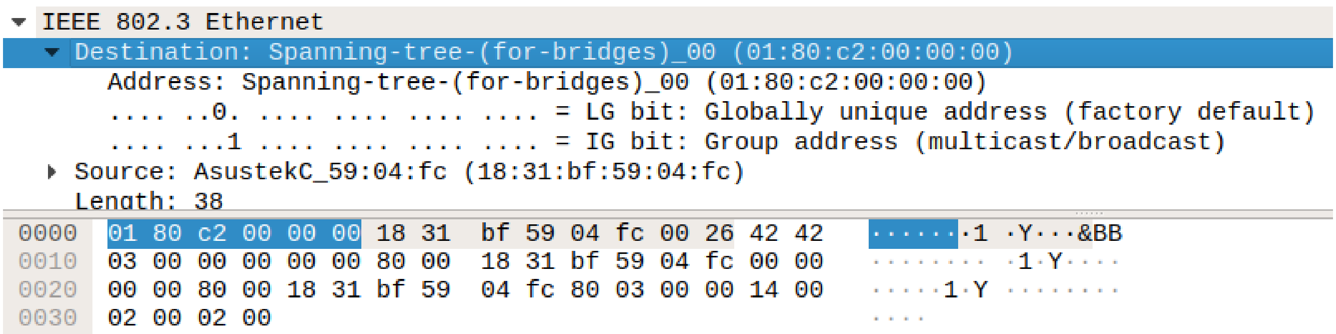
Оберіть вкладку “Options” та задайте автоматичну зупинку захоплення після 1000 кадрів. Натисніть “Start”.



Дочекайтесь захоплення 1000 кадрів і автоматичної зупинки цього процесу.

2. Виконайте статистичний аналіз захоплених кадрів за типом протоколів: Statistics → Protocol Hierarchy. Визначте кількість кадрів за типами: Ethernet DIX; 802.3/LLC; Novell 802.3; Ethernet SNAP.
3. Серед захоплених кадрів знайдіть один кадр Ethernet DIX та один 802.3/LLC.

Наведіть для кожного з них бінарний вміст та відповідний до нього аналіз. Для цього використовуйте переміщення у вікнах аналізатору



Аналізатор показує бінарний вміст кадру та відповідне призначення цього поля. Зробіть повний аналіз кожного із двох обраних кадрів, визначте призначення кожного поля та його вміст.

4. На прикладі кадрів з пункту 3 покажіть, що за алгоритмом на рис.4 дійсно можна визначити тип цих кадрів.
5. Визначте які типи MAC-адрес використано у цих кадрах: індивідуальні, групові, широкомовні. Надайте пояснення.
6. Серед захоплених кадрів знайдіть кадр з ARP-пакетом. Наведіть його бінарний вміст та за допомогою аналізатора визначте призначення та вміст кожного його поля. Наведіть, скільки серед захоплених 1000 кадрів вміщувало пакети ARP.
7. Наведіть розподіл кадрів за довжиною. Кадрів якої довжини найбільше? Скільки відсотків кадрів максимальної довжини?
8. Підготуйте звіт, в який внесіть результати 2-7 пунктів.

Запитання для самоперевірки

1. Що позначається терміном sniffer?
2. До якого типу програмних засобів відноситься Wireshark?
3. У чому особливість режиму роботи “promiscuous” мережевого адаптеру?
4. Які функції виконує аналізатор Wireshark?
5. Яка довжина та структура MAC-адреси?
6. Яку максимальну і яку мінімальну довжину можуть мати Ethernet кадри?
7. Що таке преамбула та який вона має вміст?
8. Скільки і які типи кадрів Ethernet? Чим вони відрізняються?
9. З яких полів складається кадр Ethernet DIX?
10. З яких полів складається кадр Ethernet 802.3/LLC?
11. Скільки максимально може бути передано кадрів Ethernet DIX мінімального розміру в мережі зі швидкістю 10 Мбіт/с?
12. Скільки максимально може бути передано кадрів Ethernet DIX максимального розміру в мережі зі швидкістю 10 Мбіт/с?
13. Яка максимальна продуктивність передачі даних у мережі, якщо в ній пересилають кадри Ethernet DIX мінімального розміру зі швидкістю 10 Мбіт/с?
14. Яка максимальна продуктивність передачі даних у мережі, якщо в ній пересилають кадри Ethernet DIX максимального розміру зі швидкістю 10 Мбіт/с?

ТЕМА 3. IP МЕРЕЖІ

Лабораторна робота №3. Вивчення принципів IP-адресації

Тема: Вивчення принципів IP-адресації.

Мета: Вивчити принципи IP-адресації та її використання у локальних комп’ютерних мережах.

Теоретична частина

IP-адресація

Для організації інформаційного обміну в локальних та глобальних комп’ютерних мережах широко використовують стек протоколів TCP/IP. Для здійснення цього кожному комп’ютеру призначають унікальну IP-адресу. Унікальність адреси підтримується або у глобальному масштабі, або у масштабі локальної мережі. Сьогодні найчастіше використовують IP-адресацію 4 версії. Адреса має 4-байтне значення, але для її запису зазвичай використовують десятково–крапкову (dotted-decimal) нотацію. За цією формою адресу записують за допомогою чотирьох десяткових чисел – по одному числу на байт, наприклад: 192.168.100.15.

IP-адреса складається з двох компонент – з ідентифікатору мережі (Net Id) та ідентифікатору вузлу мережі (Host Id).

мережа *вузол*

192.168.1. 15

Історично увесь простір IP-адресації 0.0.0.0 — 255.255.255.255 був поділений на п’ять класів мереж: A, B, C, D та E (Рис.5).

Клас	Діапазон адрес	мережа	вузол
A	0.0.0.0 - 127.255.255.255	8 біт	24 біта
B	128.0.0.0 - 191.255.255.255	16 біт	16 біт
C	192.0.0.0 - 223.255.255.255	24 біта	8 біт
D	224.0.0.0 - 239.255.255.255	-	-
E	240.0.0.0 - 247.255.255.255	-	-

Рис. 5. Діапазон IP-адрес у класовій моделі

У сучасних комп’ютерних мережах використовується безкласова адресація. У доповнення до IP-адреси вузлам необхідно мати інформацію про розподіл біт між ідентифікатором мережі та ідентифікатором вузла. Для цього використовується так звана маска підмережі. Маска також складається з 4 байт (32-біта). У масці біти, які встановлені як одиниця, вказують на ідентифікатор мережі відповідних біт IP-адреси, а біти, встановлені як 0 - на ідентифікатор вузла відповідних біт IP-адреси. Відповідність біт маски та IP-адреси відповідає їх однаковому розташуванню. На Рис.6 показана дія маски підмережі для двох адрес мережі. У верхньому прикладі наведено відповідність маски 255.255.255.0 до стандартного розподілу IP-адреси на 24 біта ідентифікатору підмережі та 8 біт ідентифікатору вузла, як у класі С. У нижньому прикладі показано поділ адреси на 26 біт ідентифікатору підмережі та 6 біт ідентифікатору вузла.

	24 біта			**8 біт**	
Клас С	ідентифікатор мережі			ідент. вузла	
Маска підмережі:	11111111	11111111	11111111	00000000	=255.255.255.0
	255	255	255	0	

	26 біта			**6 біт**	
	ідентифікатор мережі			ідент. вузла	
Маска підмережі:	11111111	11111111	11111111	11 000000	=255.255.255.192
	255	255	255	192	

Рис.6. Приклади роботи масок для двох підмереж.

Для вивчення принципів конфігурування в IP-мережах необхідно мати достатню кількість різноманітного мережевого обладнання та певну кількість комп’ютерів різного призначення. У цьому випадку можна створити реальну комп’ютерну мережу, але її можна попередньо змоделювати за допомогою сучасного програмного забезпечення Packet Tracer. Packet Tracer було розроблено однією з авторитетніших світових компаній мережевого обладнання Cisco Systems. Дане програмне забезпечення є безкоштовним та крос-платформеним і широко використовується при навчанні системних адміністраторів за сертифікаційними програмами Cisco Certified Network Associate (CCNA) та Cisco Certified Network Professional (CCNP). Перевагою Cisco Packet Tracer є те, що він дозволяє створювати різноманітні моделі мереж, налаштовувати різноманітне користувацьке обладнання: комутатори, роутери, безпровідне обладнання та перевіряти наявність мережевих з’єднань між ними.

Для створення мережі у Packet Tracer на Робочу область переносяться необхідні пристрої – комп’ютери, ноутбуки, сервери, принтери та комутаційне обладнання. Тип обладнання обирається на панелі типів пристроїв (див. Рис.7), а саме обладнання – в області показу “З’єднання та пристрої” (Рис.8).

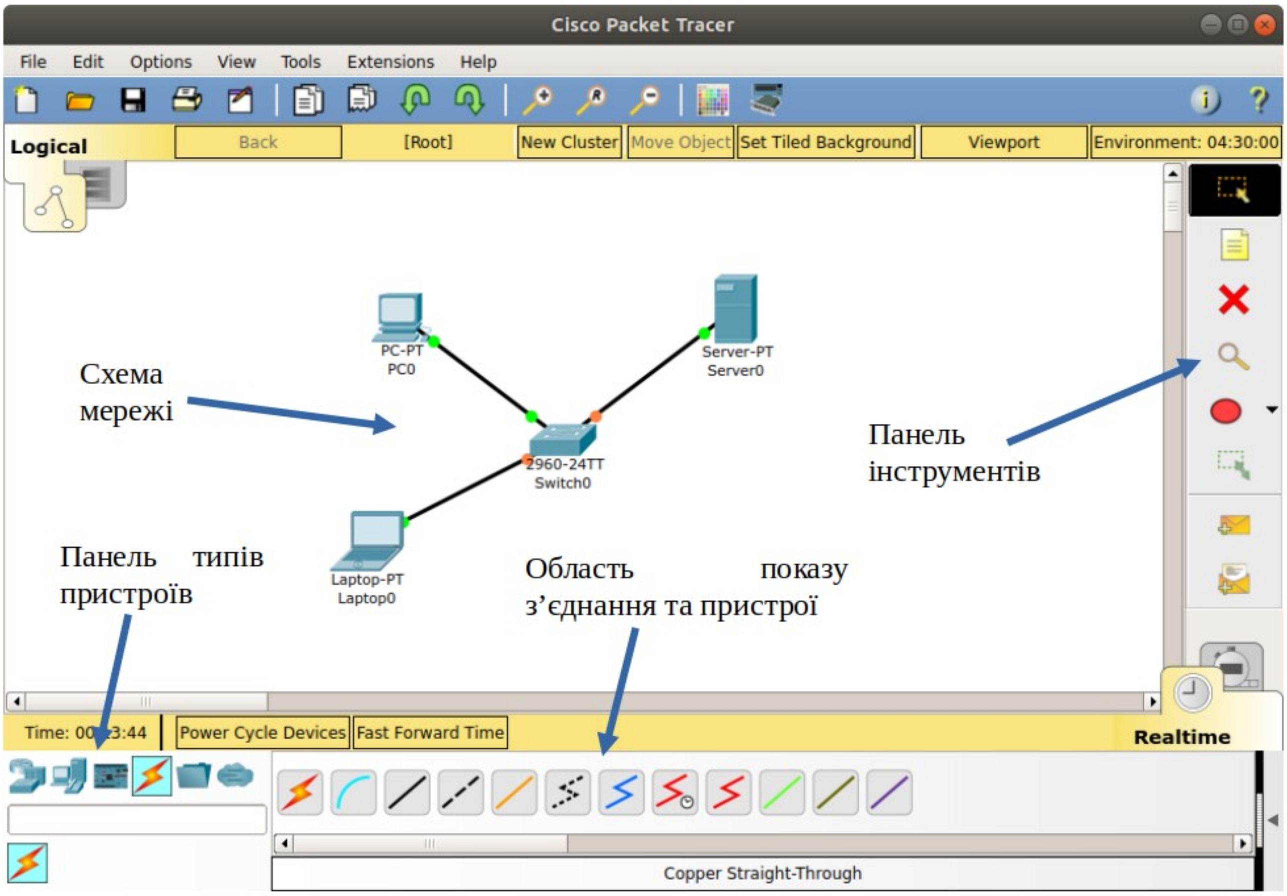


Рис.7. Вікно робочої області Packet Tracer

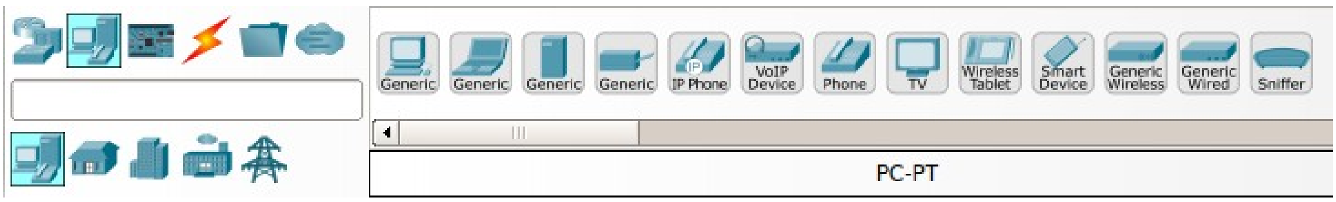


Рис.8. Панель типів пристроїв та кінцеве обладнання

Таким самим способом обирається та розміщується необхідне комутаційне мережеве обладнання: маршрутизатори (Routers), комутатори (Switches), концентратори (Hubs), бездротові пристрої (Wireless Devices) та інші (Рис.9).

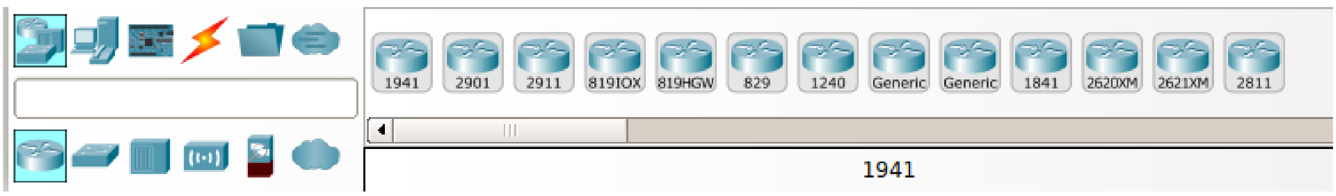


Рис.9. Комутаційне мережеве обладнання

Для з'єднання кінцевого пристроїв користувачів та мережевого обладнання на панелі вибору типів пристроїв обираються З'єднання

(Connections), а в області показу – необхідний тип з’єднання (Рис. 10). Серед типів з’єднання можна обрати певний, або скористатись автоматичним вибором типу з’єднання.

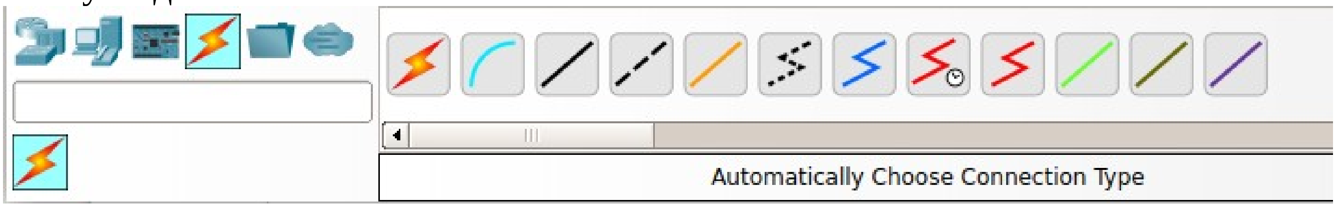


Рис.10. Типи мережевих з’єднань

Після завершення з’єднання пристроїв та мережевого обладнання Packet Tracer показує його наявність на фізичному та канальному рівнях. Якщо з’єднання з’явилось, то наявні на з’єднані кружечки будуть зеленого кольору, а якщо з’єднання немає, то червоного (Рис.11).

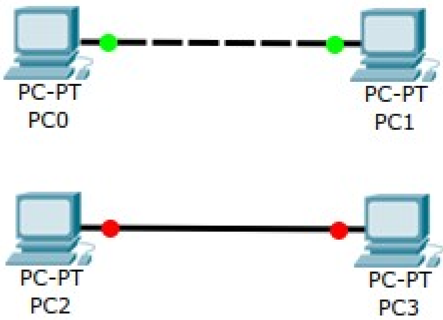


Рис.11. Індикація наявності та відсутності з’єднань

Більшість представленого в Packet Tracer обладнання та пристроїв може бути переобладнано та додатково сконфігуровано. Для цього використовується спеціальне віконце, яке відкривається після клацання лівою кнопкою миші на іконці відповідного пристрою, що вже знаходиться на Робочій області Packet Tracer (Рис.12 та 13).

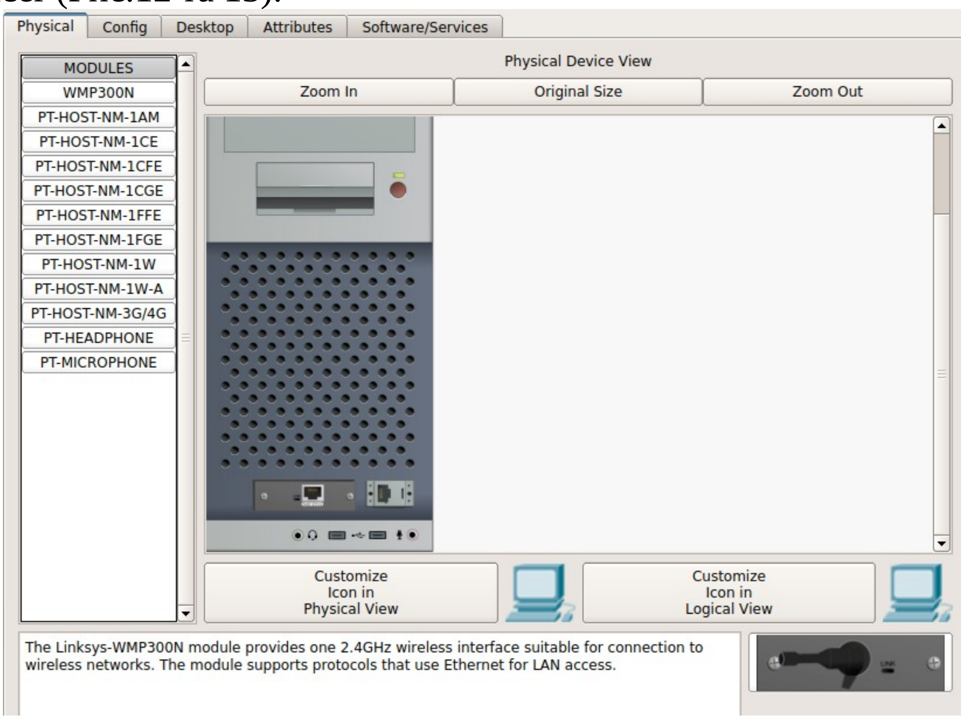


Рис.12. Вікно фізичного переобладнання та конфігурації комп’ютера.

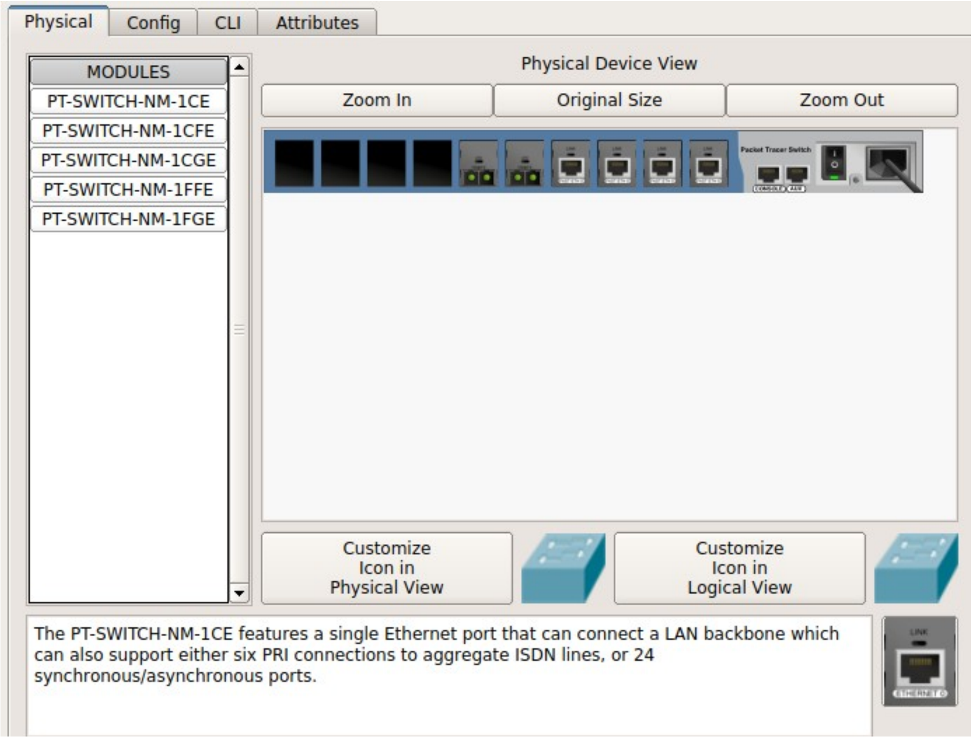


Рис.13. Вікно фізичного переобладнання та конфігурації комутатора (Switch).

Як і з реальними пристроями перед переобладнанням їх необхідно вимкнути, а після зміни або додавання компонент – увімкнути. Для програмного конфігурування використовують закладки Config або CLI (для комунікаційного обладнання) та Desktop (для персональних комп'ютерів).

Крім того, як і для реальних комп'ютерів їх мережева конфігурація може бути задана у спеціальному діалоговому вікні "Config".

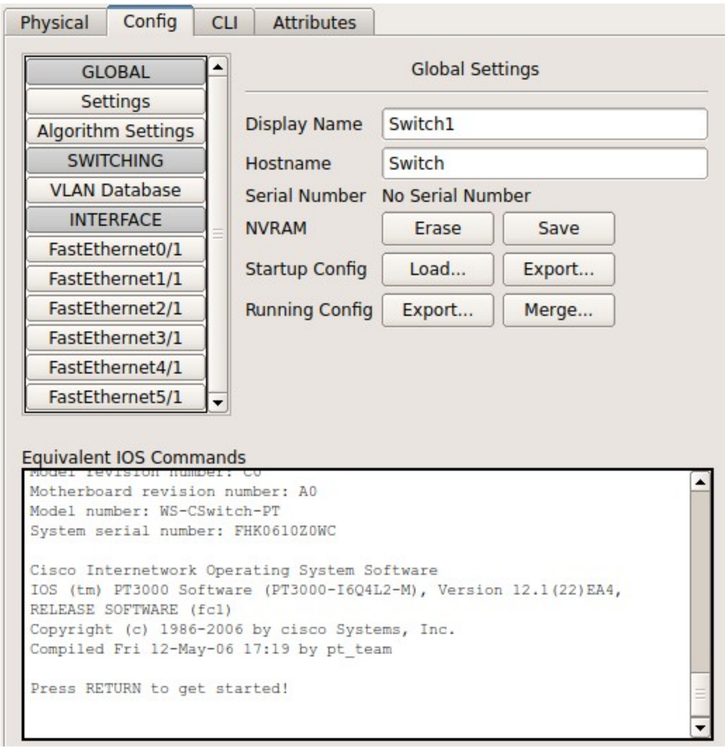


Рис.14. Поле для задавання команд комунікаційному обладнанню типу Switch.

В Packet Tracer мережеві параметри комп’ютера поділяються на глобальні та інтерфейсні. До глобальних відносяться такі параметри, як ім’я комп’ютера та IP-адреси мережевого шлюзу (Gateway) та DNS-серверу. IP-адресу та маску підмережі, які дозволять взаємодіяти комп’ютеру у мережі, задають у конфігурації інтерфейсу.

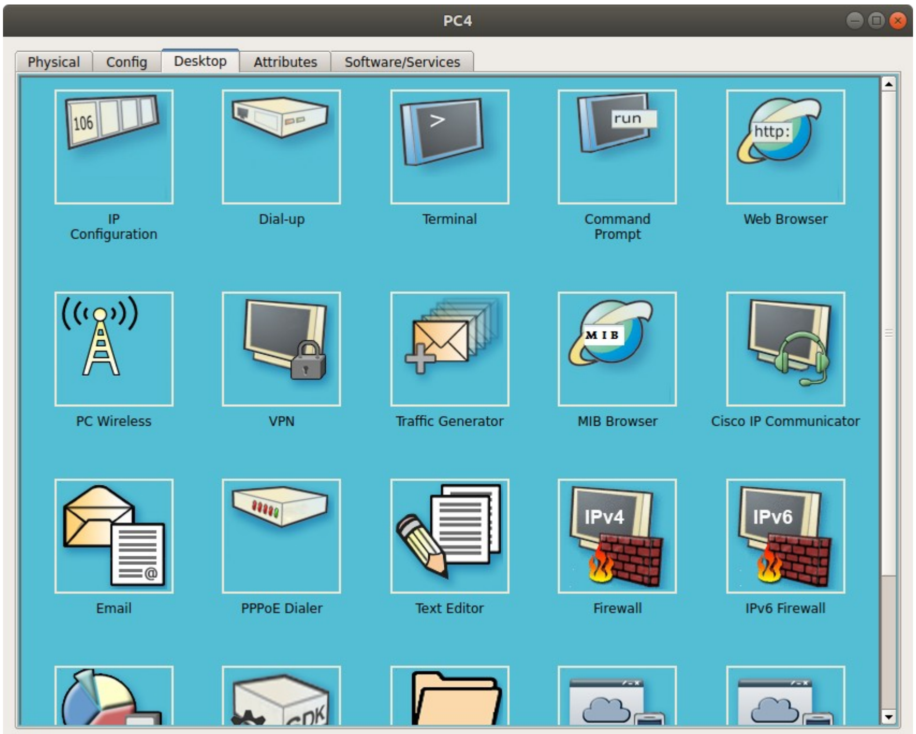


Рис.15. Різноманіття програмного забезпечення персонального комп’ютера.

Завдання до лабораторної роботи №3

1. За допомогою Packet Tracer побудуйте модель локальної комп’ютерної мережі, в яку обов’язково повинні входити концентратори (Hubs) та принаймі одна бездротова точка доступу. Кількість та тип кінцевого обладнання обирається відповідно варіанту та табл.1. Варіант відповідає Вашому номеру (N) у списку журналу академічної групи. Якщо номер N більше 20, то Ваш варіант: N-20. Ноутбуки облаштовуються бездротовим інтерфейсом, а сервери – інтерфейсами Gigabit Ethernet. Кількість концентраторів обирається за потребами, а їх з’єднання може біти або за типом топології “зірка”, або “ієрархія”.
2. Діапазон IP-адрес, що Ви можете використовувати у своїй мережі повинен відповідати 192.168.N.0 — 192.168.N.255, де N позначає Ваш номер у списку групи. Наприклад, якщо Ви у журналі академічної групи записані за номером 13, то для своєї моделі комп’ютерної мережі можете використовувати діапазон
192.168.13.0 — 192.168.13.255. Незалежно від варіанту використовуйте маску підмережі 255.255.255.0.

Таблиця 1. Варіанти завдань

варіант	ПК	сервери	принтери	ноутбуки	варіант	ПК	сервери	принтери	ноутбуки
1	5	3	3	4	11	15	1	1	1
2	6	3	3	4	12	13	2	1	1
3	7	3	2	4	13	11	2	2	1
4	8	3	2	3	14	12	2	1	2
5	9	2	2	3	15	10	2	2	1
6	10	2	1	2	16	8	3	2	3
7	11	2	1	2	17	9	3	2	2
8	12	2	1	2	18	7	3	2	3
9	13	1	1	1	19	12	3	1	1
10	14	1	1	1	20	11	3	1	2

3. Зробіть скриншот створеної моделі мережі. Занесіть його до звіту. Додайте у звіт таблицю, в якій наведіть ім’я мережевих пристроїв та комп’ютерів, а також їх мережеву конфігурацію.
4. Виконайте перевірку зв’язку між комп’ютерами та пристроями мережі, для чого використайте команду ping. Для задання цієї команди клацніть лівою кнопкою по іконці одного з комп’ютерів. Коли з'явиться вікно конфігурацій, перейдіть на вкладку Desktop та запустіть Command Prompt.

Командою ipconfig перевірте мережеву конфігурацію цього комп’ютера та зробіть її скріншот. Скріншот занесіть у звіт. Далі, з цього комп’ютеру, відправте ping до кожного комп’ютера та пристрою Вашої мережі. Результати занесіть у таблицю, в якій позначте IP-адресу та час проходження першого та четвертого пакету ping.

Запитання для самоперевірки

1. Яку комп’ютерну мережу можна назвати локальною, а яку — глобальною?
2. Яке тлумачення абревіатури IP?
3. На якому рівні моделі OSI використовується MAC-адресація?
4. На якому рівні моделі OSI використовується IP-адресація?
5. З яких частин складається IP-адреса? На що вони вказують?
6. Скільки комп’ютерів може бути у мережі класу A, B та C?
7. Для чого слугує маска підмережі?
8. Скільки IP-адрес у підмережі, якщо її маска 255.255.255.0?
9. Для чого використовується Packet Tracer?

Лабораторна робота №4. Вивчення структури та вмісту IP-пакетів

Тема: Вивчення структури та вмісту IP-пакетів.

Мета: За допомогою аналізатора Wireshark вивчити структуру та типи кадрів у комп’ютерній мережі.

Теоретична частина

Протокол IP описано у стандарті RFC 791. Основне призначення цього протоколу – передавання пакетів (дейтаграм) між мережами. У протоколі IP відсутні механізми попереднього встановлення з’єднання та підтвердження доставки. При виявлені помилки у пакеті протокол IP не забезпечує будь-яких дій щодо його повторного передавання. Важливою особливістю протоколу IP (на відміну від IPX) є динамічна фрагментація-дефрагментація пакетів при передавані їх між мережами, в яких різне максимально можливе значення довжини поля даних кадрів (Maximum Transfer Unit – MTU). При цьому, фрагментацію виконує маршрутизатор, що знаходиться на шляху руху пакету, якщо він передає пакет до мережі із меншим значенням MTU, а відновлення (збирання) пакету робить вузол-отримувач. Такий підхід дозволяє відправляти фрагменти за незалежними маршрутами.

<table>
<tr><td>Біти</td><td>0-3</td><td>4-7</td><td>8-15</td><td colspan="2">16-31</td></tr>
<tr><td>0-31</td><td>Версія IP</td><td>Довжина заголовку</td><td>Тип обслуговування</td><td colspan="2">Загальна довжина IP пакету</td></tr>
<tr><td>32-63</td><td colspan="3">Ідентифікація фрагмента</td><td>Прапорці (3 біти)</td><td>Зміщення фрагменту (13 бітів)</td></tr>
<tr><td>64-95</td><td colspan="2">TTL (час використання)</td><td>Протокол верхнього рівня</td><td colspan="2">Контрольна сума IP-пакету</td></tr>
<tr><td>96-127</td><td colspan="5">IP адреса відправника</td></tr>
<tr><td>128-159</td><td colspan="5">IP адреса отримувача</td></tr>
<tr><td>160-191</td><td colspan="5">Опції (необов’язкові) та заповнювач (за потреби)</td></tr>
<tr><td>192-...</td><td colspan="5">Корисні дані (максимальне значення - 65535 байтів мінус довжина заголовку)</td></tr>
</table>

Рис.16 Загальна структура IP пакету

IP-пакет складається із заголовку (у більшості випадків це 20 байт) та поля даних, максимальна довжина пакету - заголовок + дані складає 65535 байт, а мінімальна – визначається мінімальним розміром кадру канального рівня, який переносить IP-пакет кадру канального рівня (для Ethernet 64 байти).

Поле “Номер версії” (Version) вказує версію протоколу IP. Зараз широко використовується IP-протокол 4 версії (IPv4), а також паралельно, у більшості випадків, забезпечується підтримка IP-протоколу версії 6 (IPv6).

Поле “Довжина заголовку” (IHL) вказує значення довжини заголовку IP-пакету у 32-бітних (4-байтових) словах. Зазвичай довжина заголовку складає 5 таких слів, але може бути більшою за рахунок додаткових байт у полі “Опції” (максимальна довжина заголовку - 60 байт або 15 4-байтових слів).

Перші три біти поля “Тип сервісу” (Type of Service) (біти 0-2) задають пріоритет пакету від самого низького 000 (звичайний пакет) до самого високого 111 (пакет з інформацією керування). Біти 3-5 визначають критерій вибору маршруту, який використовується у протоколах маршрутизації OSPF (Open Shortest Path First) та BGP (Border Gateway Protocol). Реально вибір відбувається між трьома альтернативами: малою затримкою (біт 3 Delay = 1), високою пропускною здібністю (біт 4 Throughput = 1) та високою достовірністю (біт 5 Reliability = 1). Зазвичай покращення одного параметру викликає погіршення іншого, тому обирається один критерій вибору маршруту.

Поле “Загальна довжина” (Total Length) вміщує загальну довжину IP-пакету разом із заголовком. Виходячи з розрядності поля (2 байти) максимальною довжиною пакету є 65535 байт. Великі пакети використовуються рідко, навіть при пересиланні файлів та потокового відео. Зазвичай розмір пакету обирається з урахуванням максимального полю даних кадру канального рівня (MTU). Для Ethernet MTU складає 1500 байт, для FDDI MTU – 4096 байт.

При перенаправлені IP-пакета з однієї мережі в іншу маршрутизатор може зустрітись з проблемою різних значень MTU у суміжних мережах. У цьому випадку йому необхідно виконати фрагментацію пакету (тобто розбивання на декілька пакетів) при передаванні у мережу з меншим значенням MTU та дефрагментацію (об’єднання декількох пакетів у один) при передаванні пакету у мережу з великим значенням MTU. З метою розпізнавання пакетів, утворених у результаті фрагментації, використовується поле “Ідентифікатор пакета” (Identification).

Усі фрагменти фрагментованого пакету мають однакове значення цього поля. Поле “Прапорці” (Flags) має такі біти: 0 біт є резервним і його значення 0, 1 біт - DF - Do not Fragment у випадку встановлення у 1 забороняє фрагментацію пакету, 2 біт - MF - More Fragments при встановлені у 1 вказує, що пакет є проміжним (не останнім) фрагментом.

У полі “Зміщення фрагмента” (Fragment Offset) вказується зміщення у 8-байтних блоках поля даних цього пакету-фрагменту від початку загального полю даних вихідного пакету, який було фрагментовано (8 байт $\times 2^{13} = 2^{16}$ максимальний розмір пакету). Значення цього полю у першому фрагменті дорівнює 0.

Поле “Час життя” (Time To Live – TTL) вказує граничний термін часу, протягом якого пакет може переміщуватись мережею. Цей час задається джерелом пакету. При пересиланні пакету через маршрутизатор значення TTL зменшується на 1, навіть якщо час передавання через маршрутизатор менше 1 секунди. Тому кажуть, що цей час вимірюється у кількості переходів через маршрутизатори або хопи (hops). При досягненні цього значення 0 пакет далі не передається (тобто знищується).

Поле “Протокол” (Protocol) є вказівкою на протокол верхнього рівня, який розміщується у полі даних IP-пакету.

Поле “Контрольна сума заголовку” (Header Checksum) розраховується тільки по заголовку IP-пакету, тобто до поля “Дані”. При кожній зміні полів заголовку на проміжних маршрутизаторах контрольна сума заголовку перераховується. Алгоритм розрахунку – доповнення до суми усіх 16-бітних слів заголовку. При обчислені контрольної суми значення самого поля "Контрольна сума" вважається такою, що дорівнює 0.

Поля “IP-адреса відправника” (Source IP Address) та “IP-адреса отримувача” (Destination IP Address) мають значення відповідних адрес.

Поле “Опції” (IP Options) є необов’язковим та зазвичай використовується при налаштуванні мережі. У ньому може вказуватись точний маршрут проходження пакету, дані про безпеку, різноманітні часові відмітки тощо. Поле не має фіксованої довжини, тому для вирівнювання пакету відповідно до 32-бітної межі використовується поле “Вирівнювання” (Padding). Також це поле використовується для доповнення невеликого пакету до мінімального розміру даних кадру канального рівня (для Ethernet 64 байта).

<table>
<tr><td>Біти</td><td>0-3</td><td>4-7</td><td>8-15</td><td colspan="4">16-31</td></tr>
<tr><td>0-31</td><td>Версія IP</td><td>Довжина заголовку</td><td>Тип обслуговування</td><td colspan="4">Загальна довжина IP пакету</td></tr>
<tr><td>32-63</td><td colspan="3">Ідентифікація фрагмента</td><td>0</td><td>DF</td><td>MF</td><td>Зміщення фрагменту (13 бітів)</td></tr>
<tr><td>64-95</td><td colspan="2">TTL (час використання)</td><td>Протокол верхнього рівня</td><td colspan="4">Контрольна сума IP-пакету</td></tr>
<tr><td>96-127</td><td colspan="7">IP адреса відправника</td></tr>
<tr><td>128-159</td><td colspan="7">IP адреса отримувача</td></tr>
<tr><td>160-191</td><td colspan="7">Опції (необов’язкові) та заповнювач (за потреби)</td></tr>
<tr><td>192-223</td><td colspan="2">Тип ICMP</td><td>Код ICMP</td><td colspan="4">Контрольна сума ICMP повідомлення</td></tr>
<tr><td>224-...</td><td colspan="7">ICMP дані</td></tr>
</table>

Рис. 17. Структура IP пакету з ICMP-заголовком.

Для перевірки з’єднання між двома вузлами часто використовують програму ping. Ця програма формує повідомлення протоколу ICMP (Internet Control Message Protocol - протокол контрольних повідомлень Інтернету) типу "Запит відлуння" та "Відповідь на відлуння". Відповідно до вимог ICMP, які виконує кожний вузол з встановленим на ньому стеком TCP/IP, при отримані такого пакету вузол відправляє відправнику відповідь у зворотньому напрямку. За результатом цього відправник отримує відповідь і, таким чином, визначає працездатність цього з’єднання. Повідомлення ICMP пересилаються у полі даних IP пакету, при цьому, розмір заголовку ICMP зазвичай складає вісім байт.

Завдання до лабораторної роботи №4

1. Запустіть програму «Командний рядок» (Пуск —> Виконати —> cmd) або термінал.
2. Виконання команди ping з ключем -f веде до встановлення прапорця IP, що забороняє фрагментацію пакету з повідомленням ICMP. Зміна розміру поля даних ICMP дозволяє підібрати значення MTU для мережі, до якої направляється ping. Виконайте команду ping у наступному форматі:

   ping –f –l розмір (для ОС Windows)

   ping -M do -s розмір (для ОС Linux)

   де розмір — це величина блоку даних ICMP пакету, яка задається користувачем, що використовує утиліту ping.

   Поступово змінюйте параметр розмір і за результатом відмови виконання команди ping з ключем –f (-M do) визначте максимальну величину даних ICMP пакету, при якій фрагментація не виконується.
3. Запустіть аналізатор протоколів WireShark.
4. Підготуйте програмний аналізатор до захоплення пакетів ICMP за допомогою налаштування відповідних фільтрів.
5. За допомогою команди ping відправте на суміжний у мережі комп'ютер пакет відлуння з полем даних ICMP розміром 5000 байт:

ping – l 5000 *ip_адреса_суміжного_комп'ютера*

Захопіть та визначте усі пакети з фрагментами одного ICMP повідомлення.

6. Для першого фрагменту надайте вміст заголовку кадру Ethernet, IP заголовку та ICMP заголовку.
7. Надайте для кожного фрагменту значення:
   − загальної довжини пакету,
   − ідентифікатору пакету,
   − прапорців (у двійковому форматі),
   − зміщення фрагменту.
8. Виконайте додавання довжин фрагментів та порівняйте результат із заявленою довжиною блоку даних (5000 байт). Надайте пояснення результату.
9. Наступний експеримент проведіть з копіюванням файлу з локальної мережі або доступному ресурсу Інтернет на комп'ютер. Знайдіть об'єкт для копіювання (інструкцію або книжку у форматі pdf, архівний файл, тощо). Запустіть аналізатор протоколів WireShark у режимі захоплення пакетів та почніть завантаження файлу. Якщо послідовності IP пакетів, пов'язаних із завантаженням файлу, чітко визначаються у WireShark, захоплення пакетів можна зупинити. Для будь-якої такої послідовності із кількістю не менш 10 пакетів (10 рядків WireShark) визначити:

− загальну довжину пакету,
− ідентифікатор пакету,
− прапорці (у двійковому форматі),
− час життя,

− протокол верхнього рівня.
Ці значення занесіть у відповідну таблицю.
10.Підготуйте звіт.

Запитання для самоперевірки

1. Яке значення першого байту IP пакету є найбільш типовим? Дайте пояснення.
2. Яке значення буде мати поле “Загальна довжина IP пакету”, якщо заборонено фрагментування пакету?
3. Яке максимальне та мінімальне значення поля “Загальна довжина IP пакету” ?
4. Як змінюються значення ідентифікатору в IP пакетах при передаванні значних об’ємів даних у мережі?
5. Поясніть чому різняться значення MTU, максимальний розмір пакету в команді ping та значення Length у відповідному рядку WireShark для цього пакету?
6. Як у програмі WireShark задати захоплення тільки пакетів ICMP?
7. Які прапорці встановлено у IP пакеті, якщо значення поля прапорців та зміщення має наступні значення: $0 \times 0000, 0 \times 4000, 0 \times 2000, 0 \times 10FE$?
8. Яке максимальне та мінімальне значення поля TTL (час життя)?
9. Що повинен зробити маршрутизатор з отриманим IP пакетом та значенням його поля TTL, якщо:
   - IP адреса призначення, яку вказано в пакеті, не належить йому, а поле TTL має значення 60 ?
   - IP адреса призначення, яку вказано в пакеті, належить його інтерфейсу, а поле TTL має значення 1 ?
   - IP адреса призначення, яку вказано в пакеті, не належить його мережевому інтерфейсу, а поле TTL має значення 1 ?
10. Які значення буде мати поле “Протокол”, якщо IP пакет використовується для пересилання ICMP повідомлень, даних протоколів TCP та UDP ?

ТЕМА 4. ПРОТОКОЛИ ТРАНСПОРТНОГО РІВНЯ

Лабораторна робота №5. Протоколи UDP та TCP

Тема: Протоколи UDP та TCP. Структура пакетів UDP та TCP
Мета: Вивчити призначення та використання протоколів UDP та TCP

Теоретичні відомості

Протокол користувацьких датаграм UDP (User Datagram Protocol) описано у документі RFC 768. UDP є первинним сервісом, що пересилає прості повідомлення в IP пакетах без використання механізмів, що гарантують їх доставлення отримувачу. У полі заголовку IP пакета "Протокол верхнього рівня" вказує код протоколу, що переносить користувацьку датаграму на транспортному рівні (для стеку TCP/IP це або 06 - для TCP, або 17 для UDP). Аналогічно, у полі заголовку датаграми повинна бути адресна інформація, що вказує на сервіс прикладного рівня, який надає користувацькі дані для пересилання по мережі. Такою адресною інформацією є 16-розрядний номер порту. У табл.2 наведено значення та використання деяких портів протоколу UDP загальнодоступних сервісів.

Таблиця 2. Деякі загальновідомі порти протоколу UDP

Порт	Ім’я сервісу або протоколу	Документ RFC	Опис
7	echo	792	Призначено для тестування зв’язку засобом відправлення даних на вузол та отримання від нього їх у незмінному вигляді
9	DISCARD	863	Призначено для тестування зв’язку засобом відправлення даних на вузол, який відкидає прийняте та не відправляє ніякої відповіді
11	SYSTAT	866	Видає список активних користувачів в операційній системі
13	DAYTIME	867	Призначено для тестування зв’язку засобом отримання від серверу актуальної дати та часу у текстовому вигляді
19	CHARGEN	864	CHARGEN (Character Generator Protocol). Відповідає на датаграму датаграмой, що вміщує випадкове число байт (0-512)
53	DNS	1034	Domain Name System — система домених імен. Комп’ютерна розподілена система для отримання інформації про домени.
123	NTP	5905	NTP (Network Time Protocol) — використовується для синхронізації часу

Порти з номерами від 0 до 1023 закріплено у документі RFC Assigned Numbers за стандартними сервісами (well-known services). Інші номера портів (вище 1023) надаються клієнтському програмному забезпеченню по мірі необхідності способом виділення номеру порту з пулу доступних портів.

Комбінація IP-адреса та номер порту, що використовується для адресації, іноді зветься адресою socket. Адреса socket забезпечує для серверу та клієнту всю інформацію, яка необхідна для ідентифікації партнера з комунікації. За допомоги команди netstat -na можна вивести список наявних комунікацій вузлу з відображенням адрес socket.

UDP – мінімальний орієнтований на обробку повідомлень протокол транспортного рівня, який задокументований в RFC 768. Пакет UDP має визначену структуру (Рис.18). Перші два байти визначають порт відправника UDP-пакету, а наступні два — порт його отримувача. Якщо отримувач — сервер, то це буде «добре відомий» статичний порт. Поле довжини визначає загальну кількість байт у заголовку UDP та області даних. Фактична межа для довжини даних при використанні IPv4 — 65507 (8 байт на UDP-заголовок та необхідно ще 20 байт на IP-заголовок). Поле контрольної суми перевіряє коректність вмісту UDP повідомлення. Дані повинні бути кратними двом байтам, при необхідності це досягається заповненням нульовими байтами.

Біти	0-15	16-31
0-31	Порт відправника (Source port)	Порт отримувача (Destination port)
32-63	Довжина датаграми (Length)	Контрольна сума (Checksum)
64-...	Дані (Data)	

Рис.18. Структура пакету UDP

TCP (англ. transmission control protocol – протокол управління передачею) – один із основних протоколів Інтернет, який призначено для управління передачею даних. Механізм TCP забезпечує створення потоку даних з попереднім встановленням з’єднання, забезпечує повторний запит даних у випадку їх втрати та усуває дублювання при отриманні двох копій одного пакету. Такий механізм роботи TCP, на відміну від UDP, гарантує цілісність переданих даних і повідомляє відправника про результати передачі. Реалізації TCP зазвичай вбудовані у ядро ОС.

Для користувача передача даних з використанням протоколу TCP виглядає як потокова. У дійсності, TCP забезпечує обмін пакетами даних. TCP-пакет має визначену структуру, яку показано на Рис. 19.

Біти	0-3	4-9	10-15	16-31
0-31	Порт відправника			Порт отримувача
32-63	Порядковий номер			
64-95	Номер підтвердження			
96-127	Довжина заголовку	резерв	Прапорці	Розмір Вікна
128-159	Контрольна сума			Покажчик важливості
160-191	Опції (необов’язкові, але практично використовуються кожен раз)			
192-...	Дані			

Рис.19. Структура пакету TCP

Порт відправника та порт отримувача є 16-бітними полями та мають значення номерів портів відправника і адресату TCP-пакету. Подібно до UDP для TCP визначено певні порти, які асоційовано з деякими типами сервісів та серверами (табл.3). Для більшості портів програмне забезпечення використовує одні і ті самі номера, але з різними транспортними протоколами.

Таблиця 3. Деякі загальновідомі порти протоколу TCP

Порт	Ім’я сервісу або протоколу	Документ RFC	Опис
7	echo	792	див. табл.2
9	DISCARD	863	див. табл.2
11	SYSTAT	866	див. табл.2
13	DAYTIME	867	див. табл.2
19	CHARGEN	864	див. табл.2
20	FTP	959	FTP-DATA — для передавання даних FTP
21	FTP	959	FTP (File Transfer Protocol) — протокол передавання файлів у TCP-мережі. Порт 21 призначено для FTP-команд
25	SMTP	821	SMTP (Simple Mail Transfer Protocol) — протокол передавання електронної пошти у мережі
53	DNS	1034	см. табл.2
80	HTTP	1945	HTTP (HyperText Transfer Protocol) — протокол прикладного рівня для передавання даних
110	POP3	1939	POP3 (Post Office Protocol Version 3) — стандартний протокол, що використовується клієнтами електронної пошти для отримання повідомлень з віддаленого серверу
123	NTP	5905	NTP (Network Time Protocol) — використовується для синхронізації часу

Наступне 32-бітне поле (біти 32-63) є порядковим номером або номером у послідовності (sequence number). Його значення визначає положення даних TCP-пакету усереднені вихідного потоку даних, що існує в межах поточного логічного з’єднання. У момент встановлення логічного з’єднання і перший, і другий вузол взаємодії генерують свої початкові номери у послідовності. Основні вимоги до цього полю полягають у виключені повторень у проміжку часу, протягом якого TCP-пакет може знаходитись у мережі (час життя IP-пакета). Вузли обмінюються цими початковими номерами і підтверджують їх отримання. Під час відправлення TCP-пакетів з даними поле "номер в послідовності" має суму початкового номеру та кількості байт, що вказують на раніше передані дані.

Номер підтвердження (acknowledgement number) є 32-бітним полем, яке визначає кількість прийнятих даних із вхідного потоку до TCP-модулю, що створює TCP-пакет.

За ним йде чотирьох-бітове поле, що вказує на довжину заголовку TCP-пакету у 32-бітових словах та яке використовується для визначення початку розміщення даних у TCP-пакеті.

Поле «Прапорці» вміщує ознаки встановлення 6 прапорців: URG, ACK, PSH, RST, SYN, FIN.

Якщо біт прапорця URG встановлено у 1, то це означає, що TCP-пакет має важливі (urgent) дані. Встановлення у 1 прапорця ACK означає, що TCP-пакет має у полі "номер підтвердження" вірні дані.

Встановлення у 1 прапорця PSH буде вимагати невідкладної передачі прикладній програмі даних TCP-пакету, для якої їх адресовано. Підтвердження для TCP-пакету, що має значення 1 прапорця PSH, означає, що усі попередні TCP-пакети досягли адресату.

Встановлення у 1 прапорця RST означає або відповідь на отримання невірного TCP-пакету, або запит на перевстановлення логічного з'єднання.

Прапорець SYN, встановлений у 1, означає, що TCP-пакет є запитом на логічне з'єднання. Отримання пакету з встановленим прапорцем SYN повинно підтверджуватись вузлом-отримувачем.

Прапорець FIN, встановлений у 1, означає, що TCP-пакет є запитом на закриття логічного з'єднання та є ознакою кінця потоку даних, що передаються в цьому напрямку. Отримання пакету з встановленим прапорцем FIN повинно підтверджуватись вузлом-отримувачем.

Поле «Розмір вікна» є 16-бітовим полем, яке показує кількість байт інформації, що може прийняти до свого внутрішнього буферу TCP-модуль, який відправляє іншому вузлу цей TCP-пакет. Це поле використовується TCP-модулем приймача потоку даних для управління інтенсивністю цього потоку. Якщо, встановити значення поля «Розмір вікна» 0, то можна повністю зупинити передавання даних. Передавання потім можна поновити тільки, якщо розмір вікна прийме достатньо велике значення. Максимальний розмір вікна залежить від реалізації. У деяких реалізаціях максимальний розмір може встановлюватись системним адміністратором. Визначення оптимального розміру вікна є однією з найбільш складних задач реалізації протоколу TCP.

Поле «Контрольна сума» - 16-бітове поле, значення якого визначається як контрольна сума TCP-заголовку, даних пакету та псевдозаголовку, якщо він є.

Поле «Покажчик важливості» - 16-бітове поле, визначає зміщення на першого байту у тілі TCP-пакету, починає послідовність важливих (urgent) даних.

Додаткові дані заголовку – це послідовність полів довільної довжини, що описують необов'язкові дані заголовку. Протокол TCP визначає три типи додаткових даних заголовку:

- кінець списку полів додаткових даних;
- пусто (No Operation);

- максимальний розмір пакету.

Додаткові дані останнього типу надсилаються у TCP-заголовку у момент встановлення логічного з’єднання для вказування на готовність TCP-модуля щодо приймання пакетів довше 536 байтів. В UNIX-реалізаціях довжина пакету зазвичай визначається максимальною довжиною IP-сегменту для мережі.

Етапи TCP-взаємодії

Основною відмінністю TCP від UDP є те, що протокол TCP виконує додаткову задачу — забезпечує надійне доставлення даних. Для рішення цієї задачі протокол TCP використовує метод просування даних з встановлення логічного з’єднання. Логічне з’єднання дає можливість учасникам обміну стежити за тим, щоб дані не було втрачено, змінено або продубльовано, а також щоб вони прийшли до отримувача у тому порядку, в якому їх було відправлено.

Взаємодія вузлів з використанням протоколу TCP має три етапи:

- встановлення логічного з’єднання;
- обмін даними;
- закриття з’єднання.

Протокол TCP встановлює логічне з’єднання між прикладними процесами, у кожному з’єднанні приймають участь тільки два процеси. TCP-з’єднання є дуплексним – кожний з учасників цього з’єднання може одночасно отримувати та відправляти дані. При встановлені логічного з’єднання модулі TCP домовляються між собою про параметри процедури обміну даними. У протоколі TCP кожна сторона з’єднання відправляє протилежній стороні наступні параметри:

- максимальний розмір сегменту, який вона може прийняти;
- максимальний об’єм даних (можливо декілька сегментів), які вона дозволяє іншій стороні передавати у свою сторону, навіть якщо інша сторона ще не отримала квитанцію на попередню порцію даних (розмір вікна);
- початковий порядковий номер байту, з якого вона починає відлік потоку даних у рамках даного з’єднання.

У результаті переговорного процесу модулів TCP з обох сторін визначаються параметри з’єднання. Деякі з них залишаються постійними протягом усього сеансу зв’язку, а інші адаптивно змінюються. Наприклад, у залежності від завантаження буферу отримувача, а також надійності роботи мережі динамічно змінюється розмір вікна відправника.

З’єднання встановлюється за ініціативи клієнтської частини додатку. При необхідності виконати обмін даними із серверною частиною додаток-клієнт звертається до нижчого за рівнем протоколу TCP, який у відповідь на це звернення відправляє сегмент-запит на встановлення з’єднання за протоколом TCP, який працює на стороні серверу (Рис.20). У запиті є прапорець SYN, який встановлено у 1.

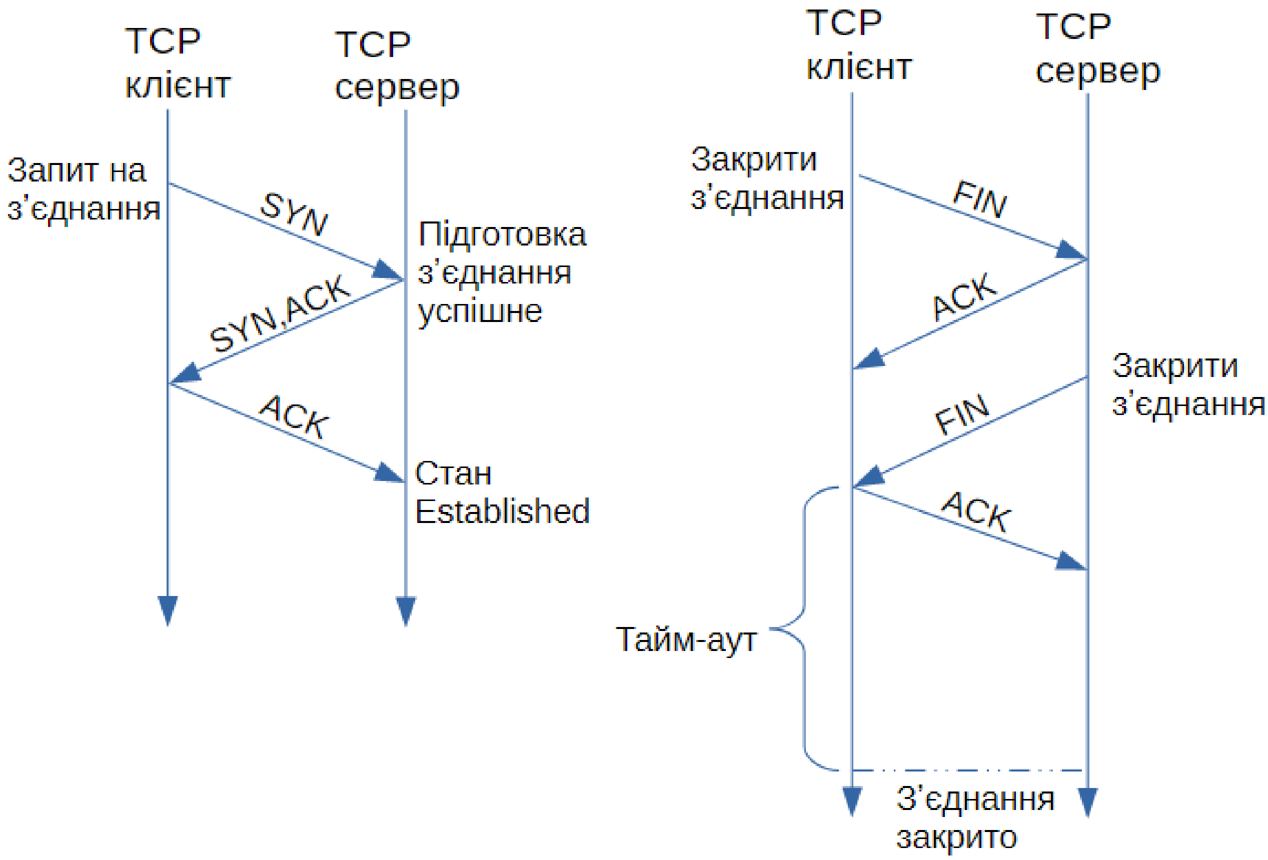


Рис.20. Процедура встановлення та розриву логічного з'єднання при нормальному перебігу процесу.

Після отримання запиту, модуль TCP на стороні серверу звертається до операційної системи щодо забезпечення певними системними ресурсами для організації буферів, таймерів, лічильників. Ці ресурси пов'язуються із з'єднанням з моменту його створення і до моменту розриву. Якщо на стороні серверу усі необхідні ресурси були отримані і всі необхідні дії виконано, то модуль TCP відправляє клієнту сегмент з прапорцями ACK та SYN.

У відповідь клієнт посилає сегмент з прапорцем ACK та переходить у стан встановленого логічного з'єднання (стан ESTABLISHED). Коли сервер отримує прапорець ACK, він також переходить у стан ESTABLISHED. На цьому процедура встановлення з'єднання закінчується і сторони можуть переходити до обміну даними.

Етап встановлення з'єднання отримав назву "потрійне рукостискання" (three-way handshake).

З'єднання може бути розірване у будь-який момент з ініціативи будь-якої сторони – клієнтської або серверної. Для цього клієнт та сервер повинні обмінятись сегментами FIN та ACK, у послідовності, яку показано на Рис. 20, де ініціатором виступає клієнт. З'єднання вважається закритим після деякого часу, на протязі якого сторона-ініціатор впевнюється, що її заключний сигнал ACK отримано нормально і він не викликав ніяких «аварійних» повідомлень з боку серверу. Сокет одночасно може приймати участь у декількох з'єднаннях.

Схема нумерації TCP передбачає привласнення порядкового номеру кожному надісланому у з’єднанні байту даних. У заголовку сегмента вказується номер першого байта поля даних цього сегмента. Від приймача необхідне підтвердження приймання сегмента. Якщо ACK не приходить за інтервал тайм-ауту, дані передаються повторно. Такий механізм називається позитивним підтвердженням з ретрансляцією (positive acknowledgment with retransmission). У підтверджені TCP, що надсилається передавачу, вказується номер наступного байту, що очікує приймач. Наслідком такого механізму є те, що у випадку пересилання останнього сегменту даних (наприклад, файлу), з номером першого байту поля даних n приймач відправить підтвердження з номером n+1. Тоді перший сегмент з наступними даними (наступний файл) буде мати номер першого байту поля даних n+1. Тому необхідна достатньо велика розрядність поля, що зберігає номер першого байту поля даних і поля, що зберігає номер підтвердження.

Протокол TCP реалізує механізми управління потоком, що надходять до вузла-приймача даних. Під час встановлення з’єднання кожний з вузлів визначає простір пам’яті для вхідного буферу з’єднання і повідомляє про це іншу сторону. Вікно приймання (receive window) є будь-яким простором у вхідному буфері, що ще не зайняте даними. Звільнення буферу від прийнятих даних виконує додаток отримувача. Цей процес залежить від продуктивності та завантаження вузлу отримувача. Кожний відправлений приймачем ACK має відомості про поточний стан вікна приймання, у залежності від якого регулюється потік даних від відправника. Зазвичай підтвердження ACK відправляються не на кожний надісланий сегмент, а на неперервний блок з декількох сегментів, який зібрано у вікні приймання. Це дозволяє не відкидати сегменти, що надійшли не за порядком відправлення, а впорядковувати їх у відповідності з послідовністю номерів та розмірам сегментів. Вузол-відправник повинен відстежувати кількість вже відправлених даних, на які надійшло підтвердження та поточний розмір приймального вікна отримувача.

Завдання до лабораторної роботи №5

1. Підготуйте аналізатор мережевих протоколів WireShark до захоплення тільки пакетів UDP, налаштував відповідний фільтр. Виконайте захоплення декількох пакетів UDP.
2. Виконайте аналіз захоплених пакетів. Наведіть у звіті дамп заголовків мережевого та транспортного рівнів одного пакету. Укажіть поле в IP-заголовку, що вказує на транспортний протокол UDP. Поясніть значення полів заголовку захопленого UDP-пакету.
3. Для захоплення та аналізу TCP-пакетів використовуйте будь-який браузер Internet. Запустіть його. Далі підготуйте аналізатор мережевих протоколів WireShark до захоплення тільки пакетів TCP, налаштував відповідний фільтр. Переведіть аналізатор у режим захоплення пакетів. У браузері уведіть адресу www.znu.edu.ua та натисніть Enter. Аналізатор

захопить множину пакетів, серед яких перші три виконують процедуру встановлення логічного з’єднання («потрійне рукостискання»).

4. Знайдіть пакети, що виконують встановлення з’єднання, та наведіть дамп їх заголовків транспортного рівня та поясніть значення їх полів.
5. Підготуйте звіт.

Запитання для самоперевірки

1. Що зветься сокетом (socket)?
2. Для чого призначено порти з номерами від 0 до 1023?
3. Яка структура UDP пакету?
4. За яких умов обміну даними у мережі використання UDP пакетів є найбільш обґрунтованим?
5. Чи можна фрагментувати IP пакети, що несуть UDP пакети? Дайте пояснення.
6. Яка найбільша довжина блоку даних, що передається за допомогою пакету UDP?
7. У чому принципова різниця між протоколами UDP та TCP?
8. Для рішення яких задач краще використовувати протокол TCP?
9. Яка структура пакету TCP?
10. На що вказують поля “Порядковий номер” та “Номер підтвердження” в пакеті TCP?
11. На що вказує поле “Розмір вікна”? Коли змінюється його значення?
12. Які прапорці і для чого використовуються в пакеті TCP?
13. Яка послідовність дій на початку обміну даними за протоколом TCP?
14. Чи можуть одночасно для TCP та UDP використовуватись порти з однаковими номерами? Дайте пояснення.

ТЕМА 5. ПРОТОКОЛИ ПРИКЛАДНОГО РІВНЯ

Лабораторна робота №6. Протокол передачі гіпертексту HTTP

Тема: Протокол передачі гіпертексту HTTP.

Мета: Вивчити принципи роботи та використання протоколу HTTP, формування HTTP запитів та структуру HTTP відповіді.

Теоретичні відомості

HTTP (HyperText Transfer Protocol – протокол передавання гіпертексту) було розроблено як основу World Wide Web. Його робота ґрунтується на принципі взаємодії клієнт-сервер: програма-клієнт установлює TCP з'єднання із сервером (стандартний номер для порту http-серверу 80) та надсилає йому запит HTTP. Сервер оброблює цей запит та надсилає клієнту HTTP відповідь. Документи, що регламентують використання протоколу HTTP, можна знайти за наступним посиланням: http://www.w3.org/Protocols/.

Структура HTTP-запиту

HTTP-запит складається із заголовку запиту та його тіла, які розділяються пустим рядком (Рис.21). Тіло запиту може бути відсутнім.

Рядок запиту (Request Line)
Заголовки (Message Headers)
Пустий рядок-поділювач
Тіло повідомлення (Entity Body)

Рис.21. Загальна структура запиту HTTP

Рядок запиту (Request Line) вказує :

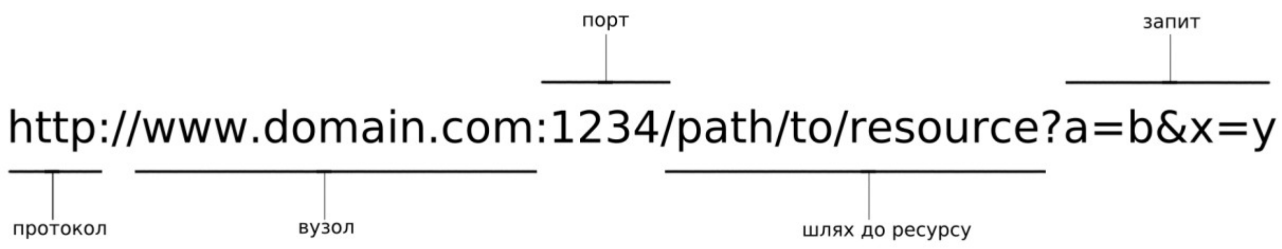
- метод передавання;
- URL-адресу, до якої необхідно звернутись;
- версію протоколу HTTP.

Методами передавання можуть бути:

- GET, якщо необхідно запитати дані разом із HTTP-заголовками;
- HEAD, якщо необхідні HTTP-заголовки без даних;
- POST, якщо необхідно відправити дані на сервер.

Для запиту за методом POST обов'язковими параметрами є тип тіла запиту (Content-Type) та довжина тіла запиту у байтах (Content-Length).

URL (англ. Uniform Resource Locator) це єдиний вказівник на ресурс. Структура URL-адреси наступна:



де протокол – тип протоколу, за яким робиться запит; вузол – ім’я серверу в системі DNS; порт – порт, який прослуховує серверна програма; шлях до ресурсу – шлях до певного ресурсу або обробника запитів на стороні серверу; запит – параметри або вирази запиту.

Для HTTP-серверів в якості протоколів в URL можна вказувати або http для звичайних з’єднань, або https – для більш безпечного обміну даними. За замовчуванням HTTP-сервери використовують порт за номером 80.

Заголовки (Message Headers) описують тіло повідомлення, передають різноманітні параметри, інші відомості та інформацію. Параметри передаються у наступному форматі:

```
Ім’я_параметру: значення_параметру
```

Наступні параметри частіше використовуються у HTTP-запиті:

Connection (з’єднання) – може приймати значення Keep-Alive або close.

Keep-Alive ("залишити живим") позначає, що після видачі даного документу з’єднання із сервером не розривається, і можна відправляти інші запити. Більшість браузерів працюють саме у режимі Keep-Alive, тому що він дозволяє за одне з’єднання із сервером отримати HTML-сторінку та, наприклад, рисунки до неї. Якщо режим Keep-Alive встановлено, то він зберігається до першої помилки або до явного вказування на черговому запиті Connection: close.

Close ("закрити") – з’єднання закривається після відповіді на даний запит.

User-Agent – його значенням є "кодове позначення" браузеру, наприклад:
Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:72.0) Gecko/20100101 Firefox/72.0
У цьому параметрі також часто вказується операційна система клієнта.

Accept – список типів, що підтримуються браузером, які наводяться у порядку їх уподобання цим браузером. Наприклад, для Firefox/72.0 цей параметр може мати наступні значення:
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8

Referer – URL, з якого перейшли на даний ресурс.

Host – ім’я вузла, на якому знаходиться ресурс у запиті. Це корисно для випадку, коли на сервері є декілька віртуальних серверів з однією IP адресою. У цьому випадку ім’я віртуального серверу визначається за допомогою цього поля.

Accept-Language – мова, що підтримується. Цей параметр має значення для серверу, який може видавати один і то1 самий документ у різних мовних версіях. Наприклад: Accept-Language: ru-RU,ru;q=0.8,en-US;q=0.5,en;q=0.3

Приклад простого HTTP-запиту:

`GET / HTTP/1.1` – перший рядок заголовку запиту;
`Host: www.znu.edu.ua` - другий рядок заголовку запиту
- пустий рядок

Структура HTTP-відповіді

Формат HTTP-відповіді є схожим на формат HTTP-запиту: він токож має заголовок та тіло, які розмежовуються пустим рядком. Заголовок також складається з основного рядка та рядків з параметрами. Основний рядок запиту складається із 3-х полів, що поділяються пробілами: версія протоколу, код помилки (кодове позначення стану виконання запиту) та словесний опис помилки. Коди 2xx позначають успішне виконання запиту, 3xx — відповідає тій чи іншій формі перенаправлення, 4xx - позначає помилку з вини клієнта, 5xx — помилка на сервері або у сценарії.

Нижче наведено параметри HTTP-відповіді, що використовуються найчастіше.

Date – дата обробки запиту.
Server – назва веб-серверу.
ETag є міткою ресурсу для статичних документів. Зазвичай вона подібна контрольній сумі або підпису файлу.
X-Powered-By – для динамічних документів вказує на додаток, який було використано для формування документу. Наприклад: X-Powered-By: PHP/5.0.3
Connection – подібний до відповідного параметру запиту.
Content-Type (тип вмісту) – має позначення типу вмісту відповіді. У залежності від значення Content-Type браузер сприймає відповідь як HTML-сторінку, картинку gif або jpeg, як файл, який необхідно зберегти на диску, або щось інше, що потребує певних визначених дій від нього. Широковживаними типами вмісту є: text/html - текст у форматі HTML (веб-сторінка); text/plain - простий текст; image/jpeg - картинка у форматі JPEG; image/gif — картинка у форматі GIF; application/octet-stream - потік даних для запису на диск. Часто в параметрі Content-Type через крапку з комою також вказується тип кодування документу.

Наприклад:

```
Content-Type: text/html; charset=windows-1251
```

Content-Length - довжина вмісту відповіді у байтах.
Last-Modified – дата останньої зміни документу.

Завдання до лабораторної роботи №6

1. Запустіть програму WireShark. Налаштуйте фільтр програми для захоплення пакетів з даними HTTP. Запустіть будь-який браузер, що встановлено на комп'ютері. Переведіть WireShark у режим захоплення пакетів з HTTP. Укажіть у рідку адреси браузеру http://www.znu.edu.ua та натисніть Enter. У програмі WireShark повинні з'явитись декілька захоплених пакетів. За заголовками HTTP визначте які з них є HTTP-запитами, а які HTTP-відповідями. Наведіть у звіті HTTP-заголовки одного сеансу «запиту-відповіді».
2. За допомогою утиліти telnet встановіть TCP з'єднання за 80-м портом з вузлом http://www.znu.edu.ua або іншим за вказівкою викладача. Для цього запустіть утиліту telnet ("Пуск" -> "Виконати" -> "cmd" -> "telnet"), встановіть режим "локальне відлуння", для того щоб бачити набраний текст. Для з'єднання з вузлом за 80-м портом введіть команду ***open www.znu.edu.ua***. Після цього можна вводити HTTP-запит, приклад якого показано в теоретичній частині. Наведіть скрин-шоти виконання завдань у звіті.
3. Використайте дані, що отримано за допомогою програми-сніферу, сформуйте HTTP-запит до серверу http://www.znu.edu.ua (можна використовувати декілька перших рядків заголовку). Після отримання HTTP-відповіді її можна скопіювати у буфер. Для цього виділіть текст за допомогою лівої клавіші миші та натисніть праву кнопку миші в межах вікна. У результаті виділений текст скопіюється у буфер і його можна буде вставити в будь-який редактор за допомогою комбінації клавіш ***Ctrl+V***. Після встановлення з'єднання відправте HTTP-запит на отримання HTTP-заголовків для головної сторінки сайту без отримання даних. Необхідно, щоб код відповіді був або 2** (три цифри, де перша 2), або 3**, або 5**. Якщо код починається з цифри 4, то було допущено помилку. У першому рядку запиту слід вказувати відносну адресу головної сторінки (тобто /), у заголовку запиту обов'язково потрібно передати параметр Host. У звіті наведіть **Screenshot** копії HTTP-запитів і відповідей серверу.
4. Повторіть сеанс зв'язку для Web-сервера www.education.zp.ua. Наведіть аналіз отриманих HTTP-відповідей та заповніть таблицю:

	Сервер 1	Сервер 2
Код відповіді		
Web-сервер (назва, версія)		
Операційна система сервера		
Статична чи динамічна сторінка Для статичної сторінки вказати Etag, для динамічної – додаток, який створював HTML-сторінку у відповідь		
Кодування документу (якщо є)		
Чи закриває з'єднання сервер одразу, чи очікує нових відповідей?		

Запитання для самоперевірки

1. Які порти прослуховує Web-сервер?
2. Яка структура HTTP-запиту до серверу?
3. Задля якої мети у HTTP-запиті використовується параметр Connection?
4. Яку інформацію несе у HTTP-запиті параметр Host ?
5. Який вигляд має простий HTTP-запит?
6. Яка структура HTTP-відповіді від серверу?
7. Які коди можуть бути у HTTP-відповіді та що вони означають?

Лабораторна робота №7. Система DNS

Тема: Вивчення системи DNS. Утиліта NSLOOKUP.
Мета: Вивчити принципи та призначення системи DNS.

Теоретична частина

Комп’ютери, що мають підключення до комп’ютерної мережі часто називають хостами (від англ. host), а для їх ідентифікації у мережі використовують *імена хоста*. Ім’я хоста – це певний мнемонічний, зручний для сприйняття людиною запис. Наприклад, znu.edu.ua, cnn.com, www.yahoo.com, gaia.cs.umass.edu, surf.eurecom.fr. У той же час для комп’ютерів прийнятними є тільки числові форми ідентифікації. Зараз для ідентифікації хостів в Інтернет широко використовується IP-адреса протоколу IPv4 – сукупність чотирьох однобайтових чисел з певною ієрархічною структурою. Символьне ім’я хоста є зручним для людини, вони легко запам’ятовується, але алгоритми обробки IP-адреси є більш простими для комп’ютерних систем, мереж та маршрутизаторів. Також важливим є те, що зв’язок (асоціація) між IP-адресою та символьним ім’ям робиться не назавжди. Зміна провайдера Інтернет або хостингової компанії буде змінювати IP-адресу але символьне (доменне) ім’я ресурсу, наприклад, вебсайту якоїсь організації скоріш за все буде залишатись тим самим, якщо, безумовно, ця організація не захоче його змінити. Крім того сам провайдер може інколи змінювати доступні йому діапазони IP-адрес, що також викликає зміну IP-адрес у споживачів його сервесів. Тому принципово важливим є здатність системи до зберігання символьної адреси при ймовірній зміни IP-адреси.

На початку створення та існування Internet (ARPANET) її зростання було помірним, тому реляція символьних імен хостів та IP-адреси могла централізовано контролюватись та підтримуватись Мережевим Інформаційним Центром (NIC – Network Information Center) за допомогою єдиного файлу (hosts.txt). Адміністратор будь-якого хоста або певної організації періодично робив копію цього файлу з використанням або електронного передавання файлів, або навіть якогось носія, наприклад, дискети. Нові ресурси додавались не часто, тому цей файл був невеликим і міг не змінюватись навіть декілька тижнів. Механізм визначення мережевої адреси за символьним ім’ям хоста полягав у звернені до файлу hosts.txt, свіжа копія якого переносилась у певний каталог на кожний комп’ютер, що підключався до мережі. Така система працювала достатньо добре поки ARPANET була порівняно невеликою мережею. Коли темп зростання та зміни складу мережі суттєво збільшились, частота звернення за оновленим файлом hosts.txt із NIC також суттєво зросла. Крім того стало необхідним відокремити управління локальними іменами і адресами у різних організаціях та компаніях, а також постійно вносити зміни до файлу hosts.txt. Тому централізована схема підтримки файлу із записами відповідності імен хостів та IP-адрес стала недостатньо практичною та повільною і потребувала зміни принципів у цьому питані. У 1983 році Пол

Мокапетріс — науковець із Каліфорнійського університету, запропонував створити автоматизовану систему доменних імен DNS (Domain Name System).

Перші результати розробки DNS було опубліковано у 1983 році в RFC 882 та 883. Після експериментів з декількома реалізаціями, DNS було формально визначено у RFC 1034 та 1035 у 1987 році. Положення, специфікації, протоколи та принципи використання DNS продовжують розвиватись і у наш час, що закріплюється відповідними стандартами у документах RFC (https://datatracker.ietf.org/).

DNS базується на двох основних концепціях:

- на розподіленій базі даних, що зберігає узагальнені записи про ресурси мережі (resource records) та має децентралізоване управління;
- на певній схемі іменування, зо базується ієрархічно структурованих доменних іменах.

DNS є розподіленою базою даних. Це дозволяє локально контролювати окремі сегменти загальної бази даних. Дані у кожному сегменті мають доступ через мережу з використанням технології клієнт-сервер. Адекватна продуктивність досягається через використання механізмів копіювання (replication) та кешування (caching).

Програми, що реалізують серверну частину DNS, звуться серверами імен (name servers). Сервер імен має інформацію про деякий сегмент загальної бази даних DNS, для якого він є повноважним сервером, та робить її доступною для клієнтів — програми вирішувачі (resolvers). Програми вирішувачі (resolvers) зазвичай надають бібліотечні функції, які генерують запити та насилають їх через мережу до серверів імен. DNS-перетворювач виконує дві основні функції. Функція *gethostbyname()* перетворює доменне ім’я у IP-адресу, а функція *gethostbyaddr()* перетворює IP-адресу у доменне ім’я. Перетворювач взаємодіє з одним або декількома DNS-серверами для виконання цих функцій від імені додатку. Для виконання цього процесу перетворювач повинен мати налаштування принаймні на один DNS-сервер. Для комп’ютера в мережі Internet зазвичай вказується список IP-адрес, кожна з яких відповідає певному DNS-серверу. Мережеві адміністратори намагаються розташовувати локальні DNS-сервери ближче до клієнтів, що використовують їх для запитів. Наприклад, в університеті комп’ютерна мережа, що об’єднує комп’ютери класів, кафедр, деканатів, відділів, також має у своєму складі локальні DNS-сервери, що обробляють запити вирішувачів від цих комп’ютерів. Зазвичай провайдер Інтернет також розташовує свої DNS-сервери у мережі, до якої підключаються його клієнти.

Кожна одиниця даних розподіленої бази даних DNS індексується за ім’ям. Ці імена, за своєю суттю, є своєрідними вказівниками в інвертованому дереві простору доменних імен.

Кожний вузол (node) у просторі імен має власну мітку без крапки тому, що крапка (".") використовується в якості роздільників міток. Максимальна довжина мітки може складати 63 байта. Максимальна довжина доменного імені (сума усіх міток та роздільників) дорівнює 255 байтам. Кореневий домен має мітку нульової довжини (null). Повне доменне ім’я кожного вузлу у дереві — це послідовність міток на шляху від цього вузла до кореня (Рис.22). За згодою, мітки, що складають доменне ім’я читають зліва на право, починаючи з нижньої, найбільш віддаленої від кореня, і завершуючи самою верхньою, яка безпосередньо знаходиться перед коренем.

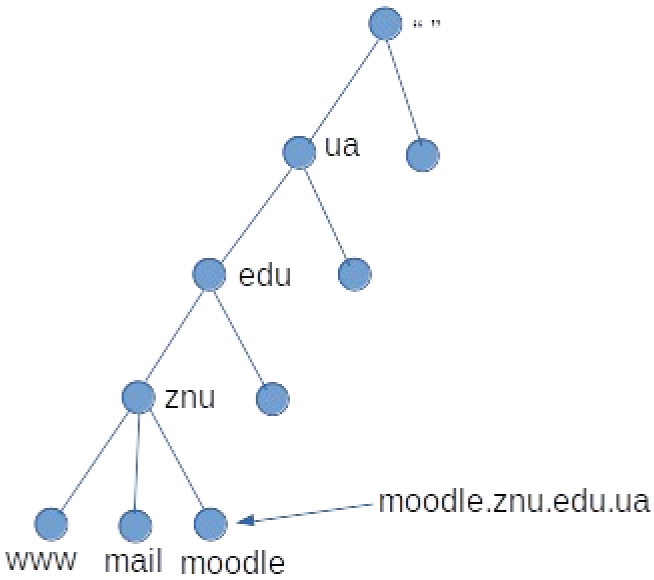


Рис.22. Приклад побудови доменного імені.

Для позначення кореневого домену в доменному імені вузла використовується символ крапки (".") наприкінці імені. Доменні імена, які закінчуються крапкою, звуться абсолютними доменними іменами (Рис.23).

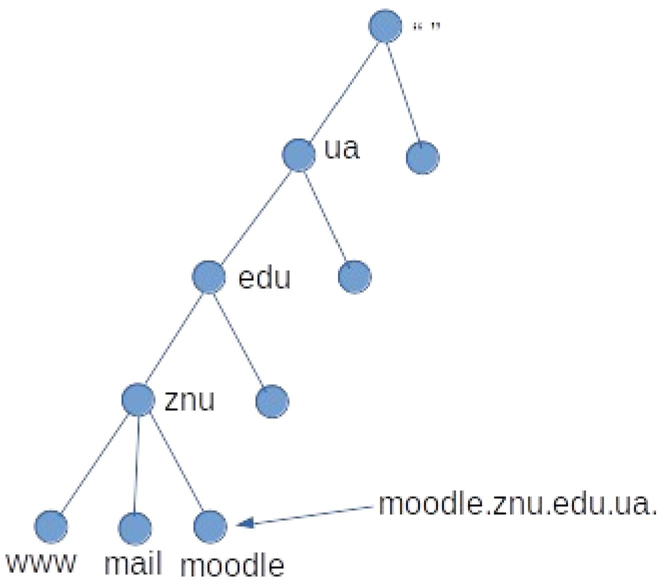


Рис.23. Приклад абсолютного доменного імені.

Абсолютне доменне ім’я зв’язане з коренем, тому воно однозначно специфікує місце знаходження вузла в ієрархії. Імена без крапки наприкінці інколи інтерпретуються як ті, що пов’язані з деяким доменом, який відрізняється від кореневого. Абсолютне доменне ім’я також зветься повністю кваліфікованим доменним ім’ям (fully-qualifed domain name або FQDN).

Домен - це деяке піддерево простору доменних імен. Доменне ім’я домену – це ім’я самого верхнього вузлу домену. Наприклад, верхнім вузлом домену "znu.edu.ua" є вузол, який має доменне ім’я "znu.edu.ua" (Рис.24).

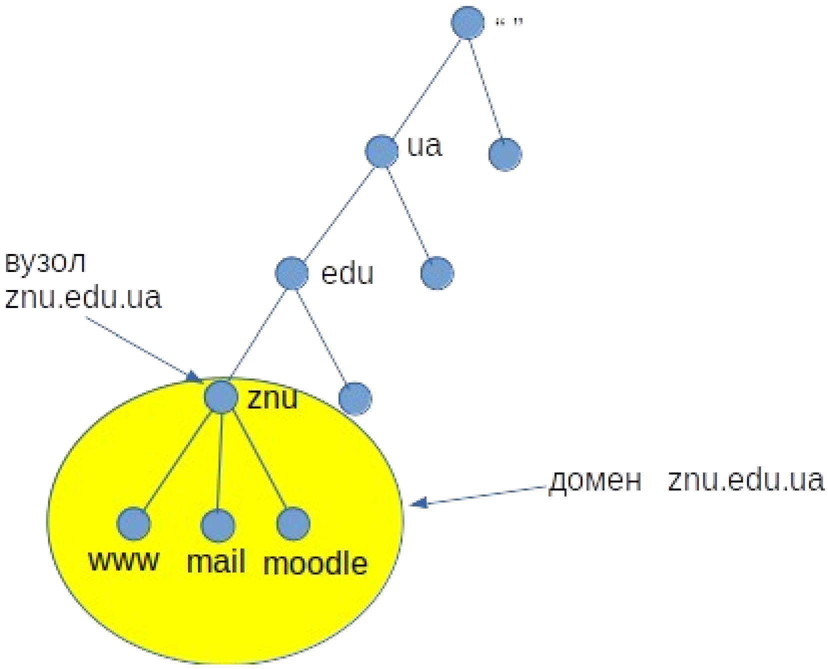


Рис.24. Верхній вузол в ієрархії має ім’я домену.

Кожне доменне ім’я може знаходитись у декількох піддеревах і, відповідно, у декількох доменах. Наприклад, доменне ім’я "moodle.znu.edu.ua" є частиною доменів "znu.edu.ua", "edu.ua", а також домену "ua". Хост-машини також є вузлами DNS-дерева, і також, за суттю доменної ієрархії, є доменами. Домен має усі хост-машини, доменні імена яких знаходяться всередині цього домену, тобто домен є суфіксом у їх іменах. Хост-машини у домені зв’язані логічно (інколи географічно або за організаційною ознакою) і не зобов’язані належати до певної мережі або класу IP-адрес. В одному домені можуть знаходиться хост-машини із різних мереж, різних стран і навіть різних континентів. Доменні імена всередині домену можуть бути іменами хост-машин або посилатись на структурну інформацію про піддомени. Одне і те саме доменне ім’я може використовуватись як для позначення хост-машини, так і для домену. Наприклад, ім’я "hp.com" є іменем домену компанії Hewlett-Packard і іменем хост-машини, яка здійснює маршрутизацію пошти між HP та Internet. Або, ім’я "znu.edu.ua" є іменем домену компанії та іменем хост-машини, яка отримує запити на веб-сервер університету. Тип інформації, яку необхідно знайти про хост-машину або піддомен, залежить від контексту використання доменного імені.

Простір імен DNS має ієрархічну структуру і містить множину вкладених доменів, кожен з яких представляє адміністративно пов'язаний набір хост-машин Internet. Безпосередньо нижче кореня цієї ієрархії знаходиться набір доменів верхнього рівня, які відповідають або певним типам організацій, або пов’язані з географічним розташуванням ресурсу. Географічне розташування ресурсу є не стільки фізичним, скільки логічним з точки зору формуванням доменного імені. Наприклад, такий домен верхнього рівня як “tv” є географічним, позначає державу Тувалу і належить її уряду. Але цей домен за контрактами з урядом Тувалу використовується в усьому світі телевізійними та медійними компаніями, реальні ресурси яких розташовуються не в Тувалу.

Дані, які асоціюються з доменними іменами, знаходяться у ресурсних записах (resouce records або RR). Записи поділяються на класи. У сучасній DNS практично використовується єдиний клас – IN (Internet). Також записи поділяються на типи. Тип записи визначає тип інформації, яка може зберігатись у просторі доменних імен. Кожний тип записи має визначений формат. В Internet-класі є адресні записи, записи серверів імен, записи вказівників та інші. Серед них є дві обов’язкові, відповідні записи повинні бути у будь-якій зоні, інакше вона буде важитись некоректною. Обов’язковими типами є SOA та NS.

Ресурсний запис є одиницею зберігання та передавання інформації в DNS і має наступні поля:

- ім’я (Name), ім’я домену, до якого відноситься цей запис;
- TTL (Time Ti Live), заданий час зберігання запису неавторизованим сервером;
- тип (Type), параметр, який визначає призначення та формат запису в полі даних (Rdata);
- клас (Class), тип мережі передавання даних;
- довжина поля даних (Rdlen);
- поле даних (Rdata), вміст та формат залежать від типу запису.

Найчастіше використовуються такі типи ресурсних записів:

- SOA (Start of Authority), "початок повноважень" або початок зони відповідальності, вміщує базові параметри доменної зони: ім’я первинного DNS-серверу та контактну адресу електронної пошти;
- A (IPv4 Address Record), адресний запис, який зв’язує доменне ім’я з IPv4 адресою хоста;
- AAAA (IPv6 Address Record), адресний запис, який зв’язує доменне ім’я з IPv6 адресою хоста;
- CNAME (Canonical Name Record), канонічний запис імені, який використовується для перенаправлення на інше доменне ім’я;
- MX (Mail Exchange) поштовий обмінник, який посилається на поштовий сервер, що обслуговує домен;
- NS (Name Server), сервер імен, що посилається на DNS-сервер, який відповідає за домен;
- TXT, текстове описання домену, часто необхідно для виконання специфічних задач, наприклад, підтвердження права власності на домен при прив’язуванні його до поштового сервісу;
- PTR (Point to Reverse), запис вказівки, що зв’язує IP адресу хоста з доменом, часто використовується поштовими сервісами.

З 1995 р. в основі DNS знаходиться система кореневих (root) серверів, яка складається з первинного (primary) сервера a.root-servers.net та його реплік (secondary). З 1997 і по сьогодення таких реплік 12, з іменами b.root-servers.net, c.root-servers.net і т.д. до m.root-servers.net. Їх список, розподіл у світі та інша інформація доступна на офіційному сайті www.root-servers.org. Коли було створено «скритий» мастер-сервер a.root-servers.net став одним із 13 рівноправних кореневих серверів, які управляються 12 незалежними організаціями-операторами, що несуть за них відповідальність:

a — VeriSign Global Registry Services;
b — University of Southern California, Information Sciences Institute;
c — Cogent Communications;
d — University of Maryland;
e — NASA Ames Research Center;
f — Internet Systems Consortium, Inc.;
g — US DoD Network Information Center;
h — US Army Research Lab;
i — Netnod;
j — VeriSign Global Registry Services;
k — RIPE NCC;
l — ICANN;
m — WIDE Project.

З метою підвищення продуктивності та стійкості системи, у відповідь на вибухове зростання інтернету і кількості звернень до доменів, з 2003 року було впроваджено технологію anycast, що дозволила операторам зробити множину дзеркал кореневих серверів, які розташовуються ближче до користувачів. Зараз число таких дзеркал становить 1467 (https://root-servers.org/).

Кореневі сервери DNS є критичним компонентом системи, тому що забезпечують доступ до кореневої зони DNS. Корнева зона має інформацію про усі домени верхнього (першого) рівня. Ця інформація вказує клієнту на які сервери DNS слід відправити запит для визначення повного доменного імені. Якщо інформація про домен у запиті не була раніше збережена у кеші клієнта, то його запит починається із звернення до кореневого серверу і буде обробленим найближчим його дзеркалом.

Якщо є необхідність змінити адресацію у кореневій зоні, наприклад, змінити склад серверів, які обслуговують домен, то адміністратор домену верхнього рівня надсилає підписаний ЕЦП запит, який ретельно перевіряється та авторизується спеціальним аудитором - IANA. Авторизований запит передається оператору VeriSign, який має право вносити зміни на скритому мастер-сервері DNS, після чого зміни розповсюджуються через захищений протокол на всі кореневі сервера.

Практична частина

Стандартною утилітою, що широко використовується для перевірки сервісів DNS-серверів, є **nslookup**. Утиліта **nslookup** інсталюється разом із протоколом TCP/IP і використовується із командного рядка. Використання nslookup.exe має декілька особливостей. Насамперед, у параметрах протоколу TCP/IP має бути вказаним принаймні один сервер DNS. Для визначення встановлених налаштувань DNS можна використати команду ipconfig /all (в операційній системі Windows), або команду resolvectl dns (в операційній системі Linux).

Утиліта nslookup може використовуватись у двох режимах: інтерактивному або неінтерактивному. Неінтерактивний режим використовується, якщо відповідь може бути у вигляді одного набору даних. Для запуску nslookup у неінтерактивному режимі використовується наступним синтаксис:

nslookup [-параметри] [вузол] [сервер]

Для запуску nslookup в інтерактивному режимі у командному рядку виконується команда nslookup:

в операційній системі Windows

```
C:\> nslookup
Default Server:  nameserver1.domain.com
Address:  10.0.0.1
>
```

в операційній системі Linux

```
(base) :~$ nslookup
> server
Default server: 127.0.0.53
Address: 127.0.0.53#53
>
```

Список доступних команд nslookup можна отримати за допомогою команди help або «?», а також в операційній системі Linux можна виконати man nslookup. Нижче наведено приклад результату виконання команди help:

```
Commands:   (ідентифікатори записано у верхньому регістрі, [] позначають опції)

 NAME                    - виводить інформацію про хост/домен NAME з використанням
                           серверу за замовчуванням
 NAME1 NAME2       - та сама операція, але використовується NAME2 як сервер
 help or ?         - виводить інформацію про стандартні команди
 set OPTION        - задає опції

    all                 - виводить параметри поточного серверу та хоста
    [no]debug           - виведення інформації налагодження
    [no]d2              - виведення повної інформації налагодження
    [no]defname         - додати ім’я домену до усіх запитів
    [no]recurse         - запит про рекурсивну відповідь на запит
    [no]search          - використовувати список пошуку доменів
    [no]vc              - завжди використовувати віртуальну схему
```

```
    domain=NAME         - встановити ім’я домену за замовчуванням NAME
    srchlist=N1[/N2/.../N6] — встановити домен N1 і список пошуку N1, N2, і т.д.
    root=NAME           - встановити кореневий сервер NAME
    retry=X             - встановити число повторень X
    timeout=X           - встановити інтервал часу очікування X секунд
     type=X                 - встановити тип запиту (наприклад, A, ANY, CNAME, MX,
                        NS, PTR, SOA, SRV)
    querytype=X         - те саме, що і type
      class=X                 - встановити клас запиту (наприклад, IN (Internet), ANY)
    [no]msxfr           - використовувати швидку зону MS для передавання
     ixfrver=X            - поточна версія, що використовується в запитах IXFR

  server  NAME          -  встановити  сервер  за  замовчуванням  NAME, використовувати
                   поточний сервер за замовчування
   lserver  NAME         -  встановити  сервер  за  замовчуванням  NAME, використовувати
                   первинний сервер
 root              - зробити поточний сервер за замовчуванням кореневим сервером
 ls [opt] DOMAIN [> FILE] — список адрес у домені DOMAIN (опціонально: виведення
                   у FILE)

    -a          -  список канонічних імен та псевдонімів
    -d          -  список усіх записів
    -t TYPE     -  список записів вказаного типу (наприклад, A, CNAME,
                   MX, NS, PTR, and so on)

 view FILE       - сортування файлу 'ls' і вивід його вмісту за допомогою pg
 exit            - вихід із програми
```

Реалізація nslookup суттєво залежить від типу та версії операційної системи. Якщо дані, які введено у командному рядку, не відповідають синтаксису команди nslookup, то вони інтерпретуються як ім’я вузла і робиться спроба визначити це ім’я за допомогою серверу за замовчуванням. Для переривання виконання команди в інтерактивному режимі використовується комбінація CTRL+C, а для завершення роботи nslookup в інтерактивному режимі – використовується команда exit.

Для зміни параметрів використання nslookup застосовується команда set. Для визначення поточного значення параметрів використовується команда set all. Наприклад:

```
> set all
Default server: 127.0.0.53
Address: 127.0.0.53#53

Set options:
  novc                  nodebug         nod2
  search                recurse
  timeout = 0           retry = 3       port = 53       ndots = 1
  querytype = A         class = IN
  srchlist =
```

Для того, щоб знайти в адресному просторі домену дані різних типів, використовуються команди set type або set q[uerytype]. Наприклад:

```
> set q=mx
> znu.edu.ua
Server:         127.0.0.53
Address:        127.0.0.53#53

Non-authoritative answer:
znu.edu.ua      mail exchanger = 10 mx1.znu.edu.ua.
znu.edu.ua      mail exchanger = 20 mx2.znu.edu.ua.
```

Позначення “Non-authoritative answer” вказує на те, що відповідь було отримано не від DNS-серверу зони відповідальності. Тобто така відповідь надходить від найближчого проміжного серверу, який отримав відповідний запит та мав у своєму кеші необхідний запис. Це відбувається тому, що записи про ресурси на DNS-серверах зони відповідальності залишаються незмінними тривалий час і кешування записів на інших DNS-серверах є звичайною практикою розвантаження DNS. Але, якщо потрібно отримати відповідь від DNS-серверу зони відповідальності, то необхідно задати ім’я потрібного сервера за допомогою команди server або lserver. Команда lserver визначає адрес сервера, на який потрібно відправляти запити, використовуючи локальний сервер. Команда server, за замовчуванням, використовує для отримання цієї адреси поточний сервер. Наприклад, якщо потрібно отримати автоматизовану відповідь для доменного імені www.znu.edu.ua, то першим визначимо DNS-сервер, що відповідає за зону znu.edu.ua:

```
> set q=NS
> znu.edu.ua
Server:         127.0.0.53
Address:        127.0.0.53#53

Non-authoritative answer:
znu.edu.ua      nameserver = ns2.trifle.net.
znu.edu.ua      nameserver = ns.znu.edu.ua.
```

Наступним визначимо його IP-адресу:

```
> set q=A
> ns.znu.edu.ua
Server:         127.0.0.53
Address:        127.0.0.53#53

Non-authoritative answer:
Name:   ns.znu.edu.ua
Address: 212.111.202.6
```

Тепер за допомогою команди server змінюємо DNS-сервер, до якого будуть надсилатись запити:

```
> server 212.111.202.6
Default server: 212.111.202.6
Address: 212.111.202.6#53
```

Після цього усі запити будуть надсилатись саме до заданого DNS-серверу. Отримаємо відповідь для www.znu.edu.ua:

```
> www.znu.edu.ua
Server:         212.111.202.6
Address:        212.111.202.6#53

Name:   www.znu.edu.ua
Address: 212.111.202.6
Name:   www.znu.edu.ua
Address: 81.90.230.250
```

У відповіді відсутнє позначення “Non-authoritative answer”.

Утиліта nslookup також дозволяє отримати повний список вузлів домену за допомогою команди ls. Загальний синтаксис команд ls:

```
ls [- a | d | t type] domain [> filename]
```

Якщо команда ls виконується без аргументів, то буде отримано список усіх серверів - імені та адрес в домені. Параметр -a дозволяє отримати список канонічних імен і псевдонімів, параметр -d — отримати список усіх записів, а параметр -t – виконати фільтрацію за типом записів.

На деяких серверах DNS передавання зон дозволено тільки для авторизованих адрес або мереж. При спробі отримати данні зони з такого серверу з’явиться наступне повідомлення про помилку:

*** Can't list domain *example.com*.: Query refused

Завдання до лабораторної роботи №7

1. За допомогою утиліти nslookup визначте IP адреси 3 будь-яких Web-серверів в Інтернет. Занесіть у звіт інформацію у вигляді: DNS-ім'я - IP-адреса.
2. Командою set type=PTR (або set querytype=PTR) переведіть nslookup у режим трансляції IP адрес у DNS імена. Визначте імена серверів по наступним IP-адресам: 3.20.100.150; 128.84.21.199; 91.198.174.192; 212.8.40.1. Наведіть у звіті результат роботи програми.
3. Командою set type=NS (або set querytype=NS) переведіть nslookup у режим визначення DNS-імен по IP адресам та завантажте список кореневих серверів імен Інтернету, для чого в рядку команди задайте "." (крапку). Результат занесіть у звіт.
4. Подібним чином отримайте список серверів імен домену ua, для чого задайте у якості команди "ua." (ua точка). Результат занесіть у звіт.
5. Подібним чином отримайте список серверів імен домену zp.ua, для чого задайте у якості команди zp.ua. (zp.ua точка). Результат занесіть у звіт.
6. За допомогою nslookup виконайте визначення існуючого імені хосту в його IP адресу (наприклад, znu.edu.ua), одночасно захопіть та проаналізуйте пакети з повідомленнями DNS, для чого використовуйте програму WireSharck. Занесіть у звіт типи транспортних протоколів та номери портів, що використовуються у повідомленнях DNS. Занесіть до звіту зміст пакету запиту до DNS-серверу і його відповідь у вигляді: Ім'я поля - Розмір у байтах - Значення.
7. За допомогою nslookup виконайте визначення неіснуючого імені хосту в його IP адресу (наприклад, таку адресу: 123.znu.edu.ua), одночасно захоплюйте та аналізуйте пакети з повідомленнями DNS. Занесіть у звіт запит до DNS-серверу та його відповідь у вигляді: Ім'я поля - Розмір у байтах - Значення.
8. Виконайте трансляцію існуючої IP адреси у ім'я хосту (наприклад, 192.168.1.1), одночасно захоплюйте та аналізуйте пакети з повідомленнями DNS. Занесіть у звіт запит до DNS-серверу та його відповідь у вигляді: Ім'я поля - Розмір у байтах - Значення.
9. Виконайте трансляцію неіснуючої IP адреси у ім'я хосту (наприклад, 192.168.240.244), одночасно захоплюйте та аналізуйте пакети з повідомленнями DNS. Занесіть у звіт запит до DNS-серверу та його відповідь у вигляді: Ім'я поля - Розмір у байтах - Значення.

Запитання для самоперевірки

1. Які RFC документи присвячено DNS системі? Які питання в них розглянуто?
2. Надайте пояснення наступних записів про ресурси: SOA, PTR, A, MX, CNAME.

3. Яке призначення утиліти nslookup?
4. Які порти використовуються для DNS сервера і клієнта?
5. Які транспортні протоколи використовуються у DNS системі? Якими RFC документами це регламентується?
6. Яке призначення і як працює програма вирішувач (resolver)?
7. Як іменуються root-сервера системи DNS?
8. Чим відрізняються авторитативна та неавторитативна відповіді?
9. Скільки рівнів DNS забезпечують ім’я www.znu.edu.ua? Надайте пояснення.
10. Наведіть приклад PTR запису та поясніть її.

ТЕМА 6. ПРИНЦИПИ ОРГАНІЗАЦІЇ ГЛОБАЛЬНИХ МЕРЕЖ ТА ІНТЕРНЕТ

Лабораторна робота №8. Маршрутизація в IP-мережах

Тема: Маршрутизація в IP-мережах.

Мета: Вивчити принципи маршрутизації в IP-мережах, конфігурування роутерів та використання маршрутизації у локальних комп'ютерних мережах.

Теоретична частина

Безкласова модель (CIDR)

Найчастіше локальна мережа має декілька мережевих сегментів. Як правило такі мережеві сегменти з'єднуються окремими пристроями – маршрутизаторами, або, як їх найчастіше називають адміністратори мереж і це вже увійшло до вживання у науковій та навчальній літературі, роутерами.

У класовій моделі адресації старші біти IP-адреси визначають належність цієї адреси до того, чи іншого класу. Головною проблемою класової моделі є нераціональність використання простору адрес в мережі. Сучасна адресація в IP мережах є безкласовою (CIDR - Classless Internet Direct Routing, пряма безкласова маршрутизація в Інтернет), положення межі мережа-вузол у IP-адресі є довільним і визначається за допомогою 32-бітової маски мережі (netmask), яка додатково вказується до IP-адреси.

Мережева маска конструюється за наступним правилом:

- на позиціях, що відповідають мережі в IP-адресі, біти мають значення 1;
- на позиціях, що відповідають номеру вузлу в IP-адресі, біти мають значення 0.

Для зручності запису IP-адреса у моделі CIDR часто представляється у вигляді a.b.c.d / n, де a.b.c.d – IP адреса, n – кількість біт IP адреси, що відносяться до мережевої частини.

Наприклад: 137.158.128.0 / 17.

Маска мережі для цієї адреси має 17 одиниць, що вказує на мережеву частина, а наступні 15 нулів — на вузлову частину. У октетовому представлені це має вигляд:

11111111.11111111.10000000.00000000 = 255.255.128.0.

Якщо представити IP-адресу у двійковому вигляді і побітово виконати її множення на мережеву маску, то отримаємо адресу мережі (у вузловій частині будуть нульові біти). Номер вузлу у цій мережі можна отримати, якщо побітово виконати множення IP-адреси на інверсію маски мережі.

Наприклад:

IP = 205.37.199.134 / 26

або теж саме,
IP = 205.37.199.134   netmask = 255.255.255.192

Запишемо у двійковій формі:
```
IP  =     11001101 00100101 11000111 10000110
маска =   11111111 11111111 11111111 11000000
```

Виконуємо множення побітово та отримуємо адресу мережі (вузлова частина відповідає нульовим бітам у масці мережі):

```
network = 11001101 00100101 11000111 10000000
```

або теж саме в октетовому представлені:

205.37.199.128 / 26

та у повній формі

205.37.199.128 netmask 255.255.255.192.

Вузлова частина наданої IP адреси дорівнює 000110, або 6. Таким чином, 205.37.199.134 / 26 адресує вузол номер 6 у мережі 205.37.199.128 / 26. У класовій моделі IP-адреса 205.37.199.134 вказувала б на вузол 134 у мережі класу C 205.37.199.0. Але, задавання маски мережі (або кількості біт у мережній частині) дозволяє визначати належність адреси до безкласової моделі та її поділ на мережеву та вузлову частини.

Мережі класів A, B, C у безкласовій моделі представляються за допомогою масок, відповідно, 255.0.0.0 (або /8), 255.255.0.0 (або /16) та 255.255.255.0 (або /24).

Завдання до лабораторної роботи №8

1. У програмі симуляторі мереж Packet Tracer складіть мережу Рис.25.
2. Варіанти розподілу IP-адрес наведено у табл.4.
3. Виконайте розрахунки для кожного мережевого сегменту та наведіть у звіті для кожної підмережі:
   - діапазон адрес,
   - мережеву маску у десятковому представлені,
   - адреси вузлів,
   - адреси роутерів,
   - адресу підмережі,
   - широкомовну адресу підмережі.
4. Виконайте конфігурування, відповідно до свого варіанту. На схемі використовуйте тільки по два вузли у кожній мережі, крім 4-ої (один вузол). Для їх адресації оберіть перші дві адреси із обчисленого

діапазону, а для роутерів — останню.

5. Перевірте зв’язок між вузлами у кожній підмережі та між мережами за допомогою команд ping та traceroute.
6. Підготуйте звіт за пунктами 1-5 та наведенням схеми і відповідних налаштувань, зроблених у Packet Tracer.

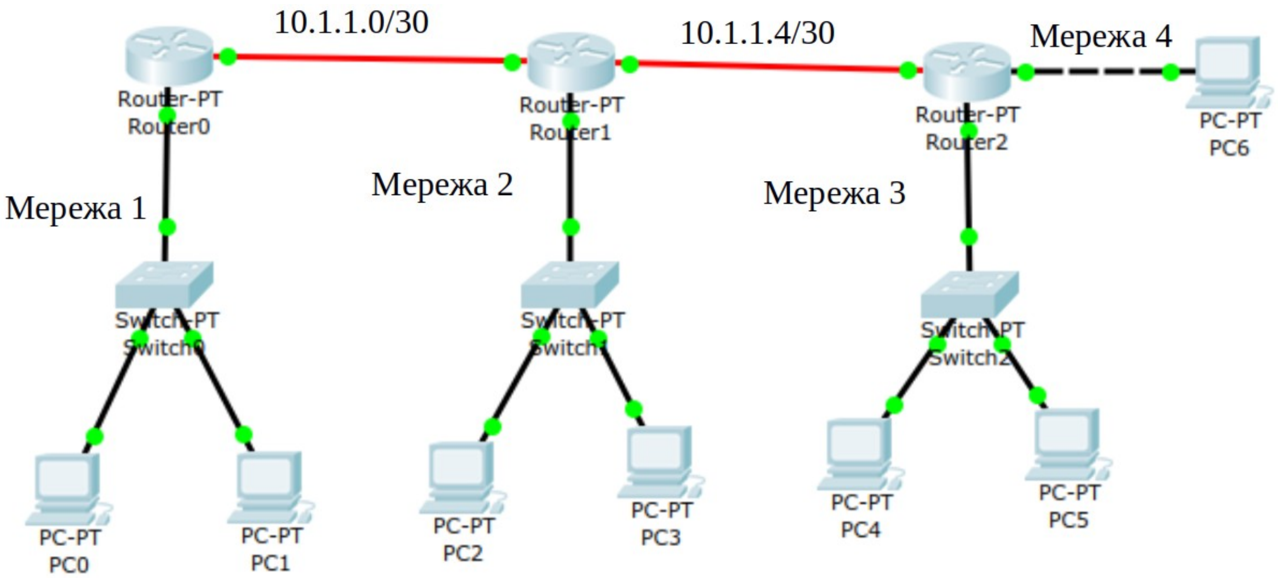


Рис.25.

Таблиця 4. Варіанти завдань

варіант	**Мережа 1**	**Мережа 2**	**Мережа 3**	**Мережа 4**
1	192.168.1.8/29	192.168.1.16/28	192.168.1.32/27	94.5.15.132/30
2	192.168.2.64/29	192.168.2.72/29	192.168.2.128/27	136.18.139.64/29
3	192.168.10.16/28	192.168.10.32/28	192.168.10.64/29	210.35.77.80/29
4	172.18.1.8/29	172.18.1.32/27	172.18.1.64/29	170.72.5.96/29
5	172.21.7.0/25	172.21.7.128/27	172.21.7.160/28	155.34.221.160/29
6	192.168.53.80/29	192.168.53.88/29	192.168.53.128/27	122.89.111.240/29
7	192.168.12.16/28	192.168.12.32/29	192.168.12.80/29	196.8.11.168/29
8	172.28.15.8/29	172.28.15.32/27	172.28.15.64/26	85.181.35.136/29
9	172.16.100.24/29	172.16.100.64/26	172.16.100.128/26	142.39.67.144/30
10	172.24.24.152/29	172.24.24.160/29	172.24.24.168/29	56.18.99.96/29
11	10.118.10.24/29	10.118.10.32/27	10.118.10.64/26	200.88.55.32/29
12	10.85.19.16/28	10.85.19.32/28	10.85.19.64/26	72.68.170.8/29
13	172.22.31.96/29	172.22.31.8/29	172.22.31.88/29	149.64.57.48/29
14	192.168.61.48/28	192.168.61.64/28	192.168.61.80/28	167.39.200.80/29
15	172.30.26.168/29	172.30.26.144/29	172.30.26.128/29	130.86.45.88/29

Запитання для самоперевірки

1. За допомогою якої маски можна поділити мережу класу А на дві підмережі? Дайте пояснення.
2. Як поділити мережу класу В на 4 частини?

3. Скільки вузлів може бути у мережі класу B та у мережі, утвореної її поділом на 4 частини?
4. На скільки підмереж можна поділити мережу класу C?
5. На скільки підмереж буде поділено мережу класу C за допомогою маски 255.255.255.224?
6. Які IP адреси мають наступні підмережі:
   - 10.0.0.64 / 26
   - 172.16.1.160 / 27
   - 192.168.10.192 / 255.255.255.240 ?
7. Які широкомовні адреси (broadcast) мають наступні підмережі:
   - 10.1.10.128 / 27
   - 172.20.100.176 / 255.255.255.240
   - 192.168.1.192 / 255.255.255.224 ?
8. Визначте адреси підмереж, в яких знаходяться комп'ютери з наступними мережевими налаштуваннями:
   - 10.1.100.155 / 255.255.255.224
   - 172.31.8.221 / 255.255.255.240
   - 192.168.30.250 / 26 ?
9. Комп'ютери з якими IP адресами будуть суміжними у мережі для комп'ютера з мережевими налаштуваннями 192.168.12.97 / 28?
10. Що означає наступний запис 172.31.8.35 / 255.255.255.255 ?

Лабораторна робота №9. Програмування простого мережевого клієнту

Тема: Програмування простого мережевого клієнту.
Мета: Вивчення принципів програмної реалізації мережевої взаємодії.

Сокет (англ. socket – роз’єм) це назва програмного інтерфейсу для забезпечення обміну даними між процесами. Процеси при такому обміні можуть виконуватись як на одному комп’ютері, так і на різних, що мають зв’язок між собою через мережу. Сокет є абстрактним об’єктом, що представляє кінцеву точку з’єднання.

Кожний процес може створювати сокет, що слухає (серверний сокет) та прив’язувати його до певного порту операційної системи (в UNIX непривілейовані процеси не можуть використовувати порти менші 1024). Процес, що слухає, зазвичай знаходиться у циклі очікування, тобто просинається при появі нового з’єднання. При цьому зберігається можливість перевірити наявність з’єднань на поточний момент, встановити тайм-аут для операції та інше.

Кожний сокет має свою адресу. ОС сімейства UNIX можуть підтримувати багато типів адрес, но обов’язковими є INET-адреси та UNIX-адреси. Якщо зв’язати сокет з UNIX-адресою, то буде створено спеціальний файл (файл сокету) за заданим шляхом, через який зможуть взаємодіяти будь-які локальні процеси через читання/запис з нього. Сокети типу INET мають доступ з мережі і потребують виділення номеру порту.

Зазвичай клієнт явно під’єднується до слухача, після чого будь-яке читання або запис через його файловий дескриптор будуть передавати дані між ним та сервером.

Таблиця 5. Основні функції

Загальні	**Опис**
Socket	Створити новий сокет і повернути файловий дескриптор
Send	Відправити дані через мережу
Receive	Отримати дані із мережі
Close	Закрити з’єднання
Серверні	
Bind	Зв’язати сокет з IP-адресою та портом
Listen	Повідомити про намір отримувати з’єднання. Слухає порт та очікує коли буде встановлено з’єднання
Accept	Прийняти запит на встановлення з’єднання
Клієнтські	
Connect	Встановити з’єднання

Метод socket() створює кінцеву точку з’єднання і повертає файловий дескриптор. Метод приймає три аргументи:

1. **domain** — вказує на сімейство протоколів сокету, що створюється
    - **AF_INET**, для позначення мережевого протоколу IPv4
    - **AF_INET6**, для IPv6
    - **AF_UNIX**, для локальних сокетів, використовує файл
2. **type** — вказує на тип сокету
    - **SOCK_STREAM**, надійна потокоорієнтований сервіс або потоковий сокет
    - **SOCK_DGRAM**, сервіс датаграм або датаграмний сокет
    - **SOCK_RAW**, протокол поверх мережевого рівня
3. **protocol**

Протоколи позначаються символьними константами з префіксом **IPPROTO_*** (наприклад, **IPPROTO_TCP** або **IPPROTO_UDP**). Значення protocol=0 інтерпретується як налаштування на використання протоколу за замовчуванням для вказаного типу з'єднання.

Функція повертає ціле число, яке вказує на привласнений дескриптор. У разі виникнення помилки функція повертає значення −1.

Нижче наведено приклад сокет-клієнту реалізований на C++ для Linux:

```
#include <iostream>
#include <sys/socket.h>
#include <arpa/inet.h>
#include <sys/types.h>
#include <string.h>
#include <resolv.h>
#include <unistd.h>
#define PORT_SOCK 80

using namespace std;
int main()
{
    struct sockaddr_in dest;
    char *host = "172.217.19.100";
    int isocket;
    isocket = socket(PF_INET,SOCK_STREAM,0);
    cout << "isocket = " << isocket << endl;
    bzero(&dest,sizeof(dest));
    dest.sin_family = AF_INET;
    dest.sin_port = htons(PORT_SOCK);
    inet_aton(host, &dest.sin_addr);
    int icon;
    icon = connect(isocket,(struct sockaddr*)&dest, sizeof(dest));
    cout << "icon = " << icon << endl;
    char buffer_wr[]=
{'G','E','T','','/','','H','T','T','P','/','1','.','0','\n','\n'};
    int legth = sizeof(buffer_wr)/sizeof(buffer_wr[0]);
    cout << "buffer_wr:" << buffer_wr << endl;
    write(isocket,buffer_wr,legth);
    int bytes_read;
    char buffer[1024];
    bytes_read = read(isocket,buffer,1024);
    cout << "bytes_read:" << bytes_read << endl;
    cout << "buffer:" << buffer << endl;
    return 0;
}
```

Наведений код слід компілювати з використанням, наприклад, такого рядку:

```
$ g++ main.cpp
```

Також для відтворення коду та компіляції можна використати інтегровані системи розробки, наприклад, Code::Blocks IDE.

Створений клієнт буде надсилати запит на встановлення сокету за IP адресою 172.217.19.100 на порт 80 — це веб-сервер www.google.com. Після утворення сокету до серверу буде надіслано запит: GET /HTTP/1.0 \n \n. У результаті веб-сервер повинен надіслати код підтвердження правильності запиту та вміст самої відповіді.

Також подібний сокет-клієнт можна створити засобами мови програмування Java:

```
package com.example;

import java.io.InputStream;
import java.io.OutputStream;
import java.net.Socket;

public class Example {

    public static void main(String[] args) {
        int c;
        try{
        Socket s = new Socket("www.google.com", 80);
        InputStream in = s.getInputStream();
        OutputStream out = s.getOutputStream();
        String str = "GET / HTTP/1.0"+"\n"+"\n";
        byte buf[]= str.getBytes();
        out.write(buf);
        while((c = in.read())!=-1){
            System.out.print((char) c);
        }
        }catch(Exception e){
            System.out.println(e);
        }
    }
}
```

Особливістю цього варіанту є його кросплатформність, тобто створена програма буде працювати на будь-якому комп’ютері, де встановлено середовище виконання Java

Завдання до лабораторної роботи №9

1. Відтворіть наведені приклади сокет-клієнтів.
2. Запустіть Wireshark у режимі захоплення пакетів за вказаними у клієнтах адресами. Виконайте захоплення пакетів при запуску та роботі кожного із наведених клієнтів. Визначить послідовність пакетів та їх роль у процедурі обміну.
3. Визначте IP-адресу свого комп'ютера. Змініть у програмах адресу, за якою звертається клієнт, на адресу свого комп'ютера. За допомогою програми Wireshark встановіть які пакети було відправлено клієнтом у запит і отримано у відповідь.
4. Оберіть 5 відомих Вам портів у діапазоні до 1024 (крім порту 80). Обґрунтуйте свій вибір. Внесіть необхідні зміни у програм клієнтів для формування правильного запиту до серверів за обраними портами.
5. За допомогою програми Wireshark вивчіть відповіді на запити клієнту до кожного серверу за обраними портами та заповніть наступну таблицю:

Номер порту та назва серверу	Транспортний протокол, що використовується у запиті	Заголовки пакетів транспортного у запиту та відповіді

6. За якими портами на було отримано відповіді?
7. Підготуйте звіт.

Запитання для самоперевірки

1. Що називається сокетом? Наведіть приклади.
2. Який RFC документ визначає поняття "socket"?
3. Який RFC документ визначає поняття "websocket"?
4. Якими процедурами (методами) реалізується у клієнті сокет-запит?
5. Як реалізується у сокет клієнті читання відповіді від серверу?
6. Які транспортні протоколи використовують сокет-клієнти?
7. Як відбувається процес встановлення сокет зв'язку?
8. Як довго може тривати встановлений сокет зв'язок?

ВИКОРИСТАНА ЛІТЕРАТУРА

1. Гордєєв О. О., Гордєєва Д. В., Колдовський М. В. Комп'ютерні мережі : навчальний посібник для студентів вищих навчальних закладів. Суми : ДВНЗ "УАБС НБУ", 2011. 250 с.
2. Микитишин А. Г., Митник М. М., Стухляк П. Д., Пасічник В. В. Комп'ютерні мережі : навчальний посібник. Львів : «Магнолія 2006», 2017. 256 с.
3. Микитишин А. Г., Митник М. М., Стухляк П. Д., Пасічник В. В. Комп'ютерні мережі : навчальний посібник. Книга 2 Львів : «Магнолія 2006», 2017. 328 с.
4. Тарнавський Ю. А., Кузьменко І. М. Організація комп'ютерних мереж : підручник для студ. спеціальності 121 «Інженерія програмного забезпечення» та 122 «Комп'ютерні науки». Київ : КПІ ім. Ігоря Сікорського, 2018. 259 с.
5. Олифер В., Олифер Н. Компьютерные сети. Принципы, технологии, протоколы : учебник для вузов. 5-е изд. Санкт-Петербург : Питер, 2016. 992 с.
6. Таненбаум Э., Уэзеролл Д. Компьютерные сети. 5-е изд. Санкт-Петербург : Питер, 2012. 960 с.

РЕКОМЕНДОВАНА ЛІТЕРАТУРА

Основна:

7. Гордєєв О. О., Гордєєва Д. В., Колдовський М. В. Комп’ютерні мережі : навчальний посібник для студентів вищих навчальних закладів. Суми : ДВНЗ “УАБС НБУ”, 2011. 250 с.
8. Микитишин А. Г., Митник М. М., Стухляк П. Д., Пасічник В. В. Комп’ютерні мережі : навчальний посібник. Львів : «Магнолія 2006», 2017. 256 с.
9. Микитишин А. Г., Митник М. М., Стухляк П. Д., Пасічник В. В. Комп’ютерні мережі : навчальний посібник. Книга 2 Львів : «Магнолія 2006», 2017. 328 с.
10. Тарнавський Ю. А., Кузьменко І. М. Організація комп’ютерних мереж : підручник для студ. спеціальності 121 «Інженерія програмного забезпечення» та 122 «Комп’ютерні науки». Київ : КПІ ім. Ігоря Сікорського, 2018. 259 с.

Додаткова:

1. Антонов В. М. Сучасні комп’ютерні мережі. Київ : «МК-Прес», 2005. 480 с.
2. Колпаков В. В., Данькевич А. О., Корж А. П., Борзенкова С. В. Промислові комп’ютерні мережі : конспект лекцій для студентів напряму підготовки «Автоматизоване управління технологічними процесами». Київ : НТУУ «КПІ», 2016. 71 с.
3. Олифер В., Олифер Н. Компьютерные сети. Принципы, технологии, протоколы : учебник для вузов. 5-е изд. Санкт-Петербург : Питер, 2016. 992 с.
4. Таненбаум Э., Уэзеролл Д. Компьютерные сети. 5-е изд. Санкт-Петербург : Питер, 2012. 960 с.
5. James F. Kurose, Keith W. Ross. Computer Networking: A Top-Down Approach (2012).
6. Jesin A. Packet Tracer Network Simulator (2014).
7. Cisco CCNA Routing and Switching ICND2 200-101 Official Cert Guide (2013).

Інформаційні ресурси:

1. Cisco Packet Tracer – Networking Simulation Tool. URL: https://www.netacad.com/courses/packet-tracer
2. Wireshark. URL: https://www.wireshark.org/
3. RFC. URL: https://www.ietf.org/standards/rfcs/
4. RFC 4960. URL: https://www.rfc-editor.org/info/rfc4960
5. IEEE 802 LAN/MAN Standards Committee. URL: https://www.ieee802.org/

Навчальне видання
(українською мовою)

Горбенко Віталій Іванович
Лісняк Андрій Олександрович
Панасенко Євген Валерійович

ОРГАНІЗАЦІЯ КОМП’ЮТЕРНИХ МЕРЕЖ

Методичні рекомендації до лабораторних занять
для здобувачів ступеня вищої освіти бакалавра
спеціальності 126 «Інформаційні системи та технології»
освітньо-професійної програми «Інформаційні системи та технології»

Рецензент *С. М. Гребенюк*
Відповідальний за випуск *А. О. Лісняк*
Коректор *Є. В. Панасенко*