ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ імені ІВАНА ФРАНКА

Факультет прикладної математики та інформатики

Комп'ютерні інформаційні мережі

ЛАБОРАТОРНА РОБОТА №7

Виконав:

Ст. *Лук'янчук Денис*

Група ПМі-33

2025

Тема: “Аналіз IP-пакетів і повідомлень керуючих протоколів. Утиліти для діагностики мережі на мережевому рівні”

Мета роботи: Здобути практичні навички з інтерпретації IP-пакетів і повідомлень керуючих протоколів, а також використання консольних утиліт для діагностики мережі на мережевому рівні.

Хід роботи

1. Ознайомився з базовою мережевою конфігурацією свого комп’ютера, виконавши в консолі команду ipconfig:

```
PS C:\Users\denys> ipconfig

Windows IP Configuration


Wireless LAN adapter Підключення через локальну мережу* 1:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Wireless LAN adapter Підключення через локальну мережу* 2:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Wireless LAN adapter Wi-Fi:

   Connection-specific DNS Suffix  . : mshome.net
   Link-local IPv6 Address . . . . . : fe80::1c1c:ef55:3a64:716d%8
   IPv4 Address. . . . . . . . . . . : 192.168.0.227
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.0.1

Tunnel adapter Teredo Tunneling Pseudo-Interface:

   Connection-specific DNS Suffix  . :
   IPv6 Address. . . . . . . . . . . : 2001:0:284a:364:cc6:c88e:d2a6:a64b
   Link-local IPv6 Address . . . . . : fe80::cc6:c88e:d2a6:a64b%16
   Default Gateway . . . . . . . . . : ::

Ethernet adapter vEthernet (WSL (Hyper-V firewall)):

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::5136:21ac:1b05:415b%42
   IPv4 Address. . . . . . . . . . . : 172.21.128.1
   Subnet Mask . . . . . . . . . . . : 255.255.240.0
   Default Gateway . . . . . . . . . :
```

2. Для отримання більш детальної конфігурації виконав команду ipconfig /all. Звернув увагу на такі нові поля, як MAC-адреса і адреса DNS-сервера:

```
Wireless LAN adapter Wi-Fi:

   Connection-specific DNS Suffix  . : mshome.net
   Description . . . . . . . . . . . : Realtek RTL8852BE WiFi 6 802.11ax PCIe Adapter
   Physical Address. . . . . . . . . : D0-39-57-41-8F-41
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes
   Link-local IPv6 Address . . . . . : fe80::1c1c:ef55:3a64:716d%8(Preferred)
   IPv4 Address. . . . . . . . . . . : 192.168.0.227(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Lease Obtained. . . . . . . . . . : 3 листопада 2025 р. 16:45:16
   Lease Expires . . . . . . . . . . : 3 листопада 2025 р. 18:45:16
   Default Gateway . . . . . . . . . : 192.168.0.1
   DHCP Server . . . . . . . . . . . : 192.168.0.1
   DHCPv6 IAID . . . . . . . . . . . : 80755031
   DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-2E-F8-86-91-D0-39-57-41-8F-41
   DNS Servers . . . . . . . . . . . : fe80::404f:5eb0:880a:f3%8
                                       192.168.0.1
   NetBIOS over Tcpip. . . . . . . . : Enabled
   Connection-specific DNS Suffix Search List :
                                       mshome.net
```

3. *Опишіть призначення команд ipconfig /renew та ipconfig /release*:

- **ipconfig /all** – показує повну інформацію про мережеві налаштування (IP, маску, шлюз, MAC, DNS).
- **ipconfig /release** – відключає поточну IP-адресу, отриману від DHCP-сервера.
- **ipconfig /renew** – запитує нову IP-адресу від DHCP-сервера.

4. Переглянув активні TCP-з’єднання за допомогою команди netstat:

```
PS C:\Users\denys> netstat

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    127.0.0.1:49350        kubernetes:51887       CLOSE_WAIT
  TCP    127.0.0.1:49676        kubernetes:49677       ESTABLISHED
  TCP    127.0.0.1:49677        kubernetes:49676       ESTABLISHED
  TCP    127.0.0.1:49679        kubernetes:49680       ESTABLISHED
  TCP    127.0.0.1:49680        kubernetes:49679       ESTABLISHED
  TCP    127.0.0.1:51821        kubernetes:49350       TIME_WAIT
  TCP    127.0.0.1:51824        kubernetes:49350       TIME_WAIT
  TCP    127.0.0.1:51825        kubernetes:49350       TIME_WAIT
  TCP    127.0.0.1:51827        kubernetes:49350       TIME_WAIT
  TCP    127.0.0.1:51829        kubernetes:49350       TIME_WAIT
  TCP    127.0.0.1:51830        kubernetes:49350       TIME_WAIT
  TCP    127.0.0.1:51831        kubernetes:49350       TIME_WAIT
  TCP    127.0.0.1:51832        kubernetes:49350       TIME_WAIT
  TCP    127.0.0.1:51833        kubernetes:49350       TIME_WAIT
  TCP    127.0.0.1:51834        kubernetes:49350       TIME_WAIT
  TCP    127.0.0.1:51835        kubernetes:49350       TIME_WAIT
  TCP    127.0.0.1:51836        kubernetes:49350       TIME_WAIT
  TCP    127.0.0.1:51837        kubernetes:49350       TIME_WAIT
  TCP    127.0.0.1:51838        kubernetes:49350       TIME_WAIT
  TCP    127.0.0.1:51839        kubernetes:49350       TIME_WAIT
  TCP    127.0.0.1:51840        kubernetes:49350       TIME_WAIT
  TCP    127.0.0.1:51841        kubernetes:49350       TIME_WAIT
  TCP    127.0.0.1:51842        kubernetes:49350       TIME_WAIT
  TCP    127.0.0.1:51843        kubernetes:49350       TIME_WAIT
  TCP    127.0.0.1:51844        kubernetes:49350       TIME_WAIT
  TCP    127.0.0.1:51845        kubernetes:49350       TIME_WAIT
  TCP    127.0.0.1:51846        kubernetes:49350       TIME_WAIT
  TCP    127.0.0.1:51847        kubernetes:49350       TIME_WAIT
  TCP    127.0.0.1:51850        kubernetes:49350       TIME_WAIT
  TCP    127.0.0.1:51853        kubernetes:49350       TIME_WAIT
  TCP    127.0.0.1:51854        kubernetes:49350       TIME_WAIT
  TCP    127.0.0.1:51855        kubernetes:49350       TIME_WAIT
  TCP    127.0.0.1:51856        kubernetes:49350       TIME_WAIT
  TCP    127.0.0.1:51857        kubernetes:49350       TIME_WAIT
  TCP    127.0.0.1:51858        kubernetes:49350       TIME_WAIT
  TCP    127.0.0.1:51859        kubernetes:49350       TIME_WAIT
  TCP    127.0.0.1:51861        kubernetes:49350       TIME_WAIT
  TCP    127.0.0.1:51863        kubernetes:49350       TIME_WAIT
  TCP    127.0.0.1:51864        kubernetes:49350       TIME_WAIT
  TCP    127.0.0.1:51865        kubernetes:49350       TIME_WAIT
```

5. Для отримання статистики про отримані/відправлені пакети виконав команду netstat -e.

```
C:\Users\denys>netstat -e
Interface Statistics

                           Received            Sent

Bytes                    3118299470      2557080663
Unicast packets             2320274         1433472
Non-unicast packets           29822          214938
Discards                          0               0
Errors                            0               1
Unknown protocols                 0
```

6. Запустив Wireshark у режимі адміністратора.

7. Від’єднався від мережі.

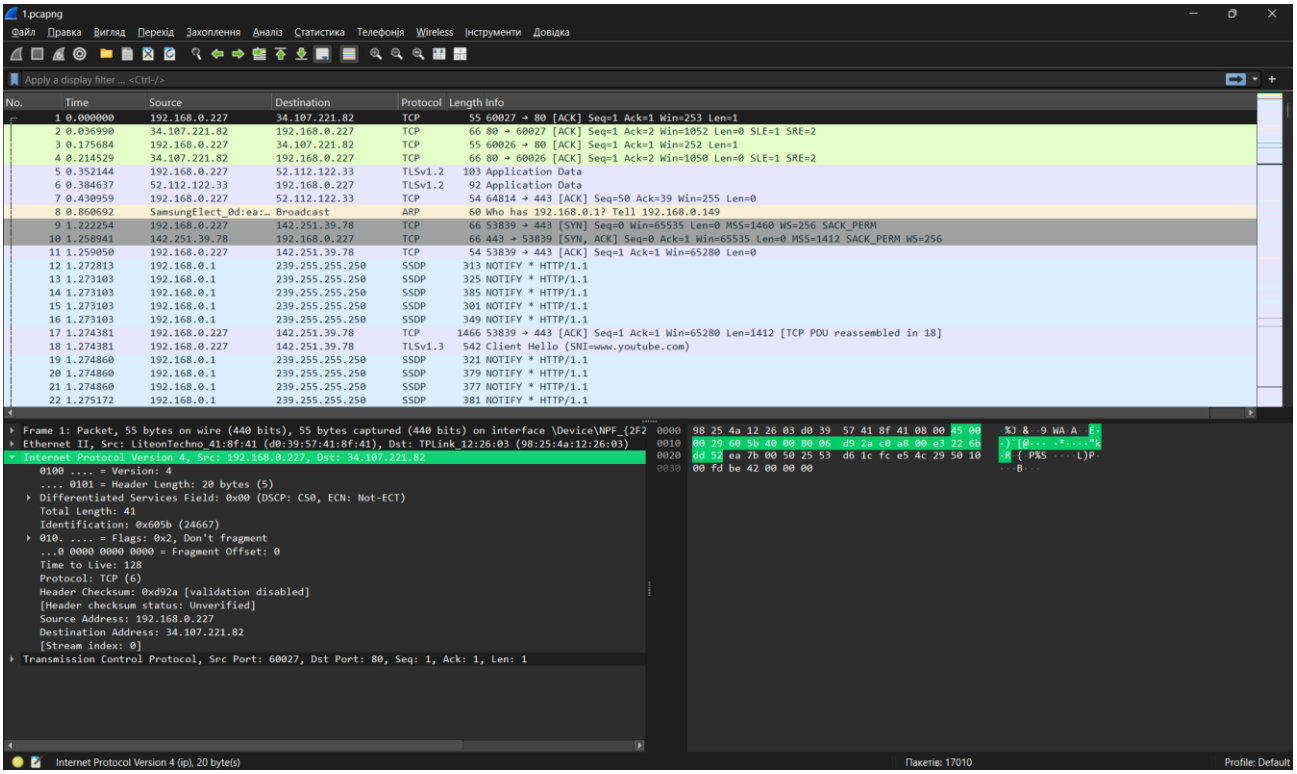
8. Почав захоплення пакетів усіх інтерфейсів.

9. Під ’єднався до мережі.

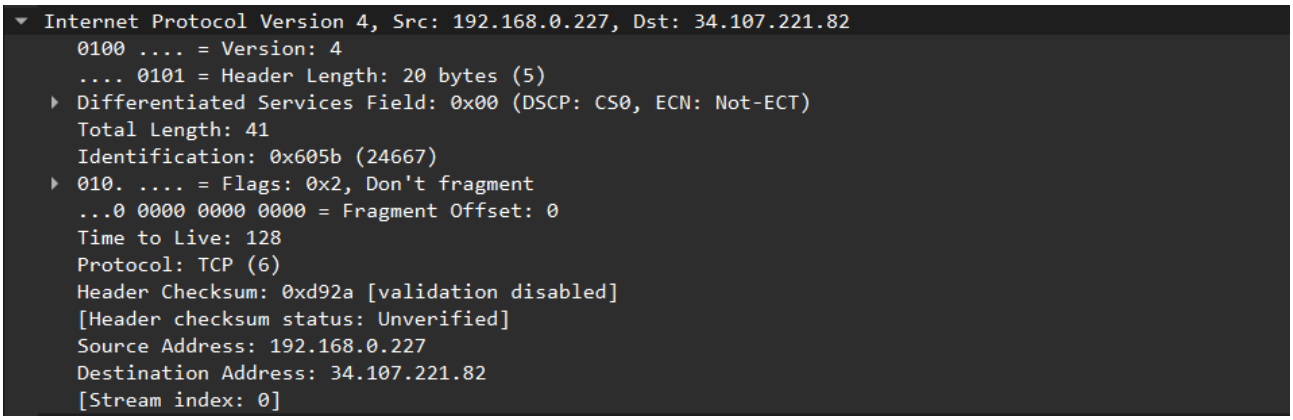
10. Здійснив активність у браузері.

11. Закінчив захоплення пакетів та зберіг результат у файл.

12. Вибрав пакет для аналізу, клацнув на рядку Internet Protocol version 4 … в області “Ієрархічний вміст пакета”. В області “Бітове подання” підсвічуються біти, які відповідають заголовку IP-пакета:



13. Розгорнув заголовок, та описав інформацію, яку надає кожне його поле



Під час аналізу пакета, отриманого у Wireshark, були виявлені такі основні поля заголовка IPv4:

- **Version** – значення 4, що вказує на використання протоколу IPv4.
- **Header Length** – біти 0101, які у десятковій системі дорівнюють 5. Оскільки кожна одиниця означає 4 байти, маємо 5 × 4 = 20 байт заголовка.
- **Differentiated Services Field** – 0x00. Це колишнє поле ToS, яке використовується для визначення пріоритету пакетів у мережі. У цьому випадку всі біти дорівнюють нулю, тобто трафік передається у звичайному режимі Best Effort, без пріоритезації.
- **Total Length** – 41 байт. Це повний розмір пакета разом із заголовком і даними.
- **Identification** – 0x605b (24667). Це унікальний номер, за яким фрагменти пакета можуть бути об’єднані під час збору.
- **Flags** – 0x2 (Don’t Fragment). У цьому полі зазначено, що пакет не можна фрагментувати під час передавання.
- **Fragment Offset** – 0, тобто пакет не є фрагментом і передається цілком.
- **Time to Live (TTL)** – 128. Це кількість маршрутизаторів, через які може пройти пакет, перш ніж буде відкинутий.
- **Protocol** – TCP (значення 6), що означає, що всередині IPv4-пакета передаються TCP-сегменти.
- **Header Checksum** – контрольна сума, що використовується для перевірки правильності переданого заголовка.
- **Source Address** – 192.168.0.227. Це локальна IP-адреса відправника, тобто

пристрій знаходиться у приватній мережі (LAN).

- **Destination Address** – 34.107.221.82. Це публічна адреса, яка належить зовнішньому серверу, розміщеному у хмарі.

14. *Дайте відповідь на запитання "Біти 0100 поля Версія (Version) дають десяткове число 4, яке відповідає протоколу IPv4. Чому тоді біти 0101 (десяткове число 5) поля Довжина заголовку (Header Length) відповідають значенню 20 байт, а не 5 байт?"*

Біти 0101 у полі довжини заголовка відповідають числу 5, але це не означає 5 байт. У протоколі IPv4 довжина заголовка задається у словах по 4 байти, тому 5 × 4 = 20 байт. Це стандартна довжина заголовка без додаткових опцій.

15. *Дайте відповідь на запитання (на прикладі свого пакету, а не на прикладі пакету, який досліджується тут) "Який розмір корисних даних?"*

Повна довжина пакета – 41 байт, довжина заголовка – 20 байт. Отже, розмір корисних даних становить 41 − 20 = 21 байт. Це ті дані, які передаються всередині TCP.

16. *Дайте відповідь на запитання (на прикладі свого пакету, а не на прикладі пакету, який досліджується тут) "Що Ви можете сказати про одержувача та відправника за виглядом їхніх IP-адрес?"*

Відправник має адресу 192.168.0.227, що належить до приватного діапазону IP-адрес (локальна мережа).
Одержувач – 34.107.221.82 – це публічна IP-адреса, розташована в інтернеті. За інформацією з відкритих джерел, ця адреса належить інфраструктурі Google Cloud, тому можна зробити висновок, що користувач у локальній мережі звертається до зовнішнього інтернет-сервера.

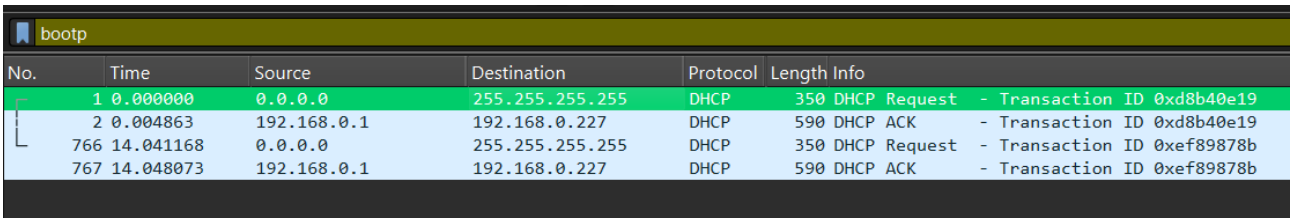
17. *Відшукайте у відкритих джерелах інформацію про поле Тип обслуговування (раніше воно називалося ToS, тепер DSCP). Опишіть, з яких частин воно складається і яку інформацію несуть біти кожної частини:*

Поле складається з 8 бітів, які поділяються так:

- 6 біт (DSCP – Differentiated Services Code Point) – визначають клас обслуговування або пріоритет трафіку.
- 2 біти (ECN – Explicit Congestion Notification) – використовуються для повідомлення про перевантаження мережі без втрати пакетів.

У цьому пакеті значення поля 0x00, тобто DSCP = 000000 і ECN = 00, що означає звичайну доставку без спеціального пріоритету.

18. Дослідив протокол динамічної конфігурації хостів (DHCP). Для відображення пакетів цього протоколу скористався фільтром bootp:



19. Пояснив IP-адреси відправника у DHCP-запиті:

У повідомленні **DHCP Request** клієнт звертається до DHCP-сервера з проханням підтвердити або оновити IP-адресу.

- **Адреса відправника:** 0.0.0.0 — комп'ютер ще не має власної IP-адреси, тому тимчасово використовує цю.
- **Адреса отримувача:** 255.255.255.255 — широкомовна адреса, яка дозволяє пакету дійти до всіх пристроїв у локальній мережі, включно з DHCP-сервером.

Таким чином, клієнт надсилає запит на отримання або оновлення IP-адреси для себе.

20. Пояснив IP-адреси отримувача у DHCP-запиті:

У повідомленні **DHCP ACK** DHCP-сервер підтверджує, що клієнту видано або подовжено дію IP-адреси.

- **Адреса відправника:** 192.168.0.1 — це адреса DHCP-сервера, яким, як правило, є **роутер**.
- **Адреса отримувача:** 192.168.0.227 — це IP-адреса клієнта, яку сервер підтвердив для використання в мережі.

Після отримання ACK клієнт остаточно приймає видану адресу й може працювати в локальній мережі або в інтернеті.

21. Розгортаючи по черзі опції, зазначив, яка інформація передається у DHCP-запиті:

Option 53 – DHCP Message Type (Request) Містить тип DHCP-повідомлення. Значення 3 означає Request — запит клієнта на отримання або поновлення IP-адреси.

Option 61 – Client Identifier Ідентифікує клієнта. У полі зазначено тип інтерфейсу і його MAC-адресу: d0:39:57:41:8f:41.

Option 50 – Requested IP Address (192.168.0.227) Клієнт просить видати саме цю IP-адресу.

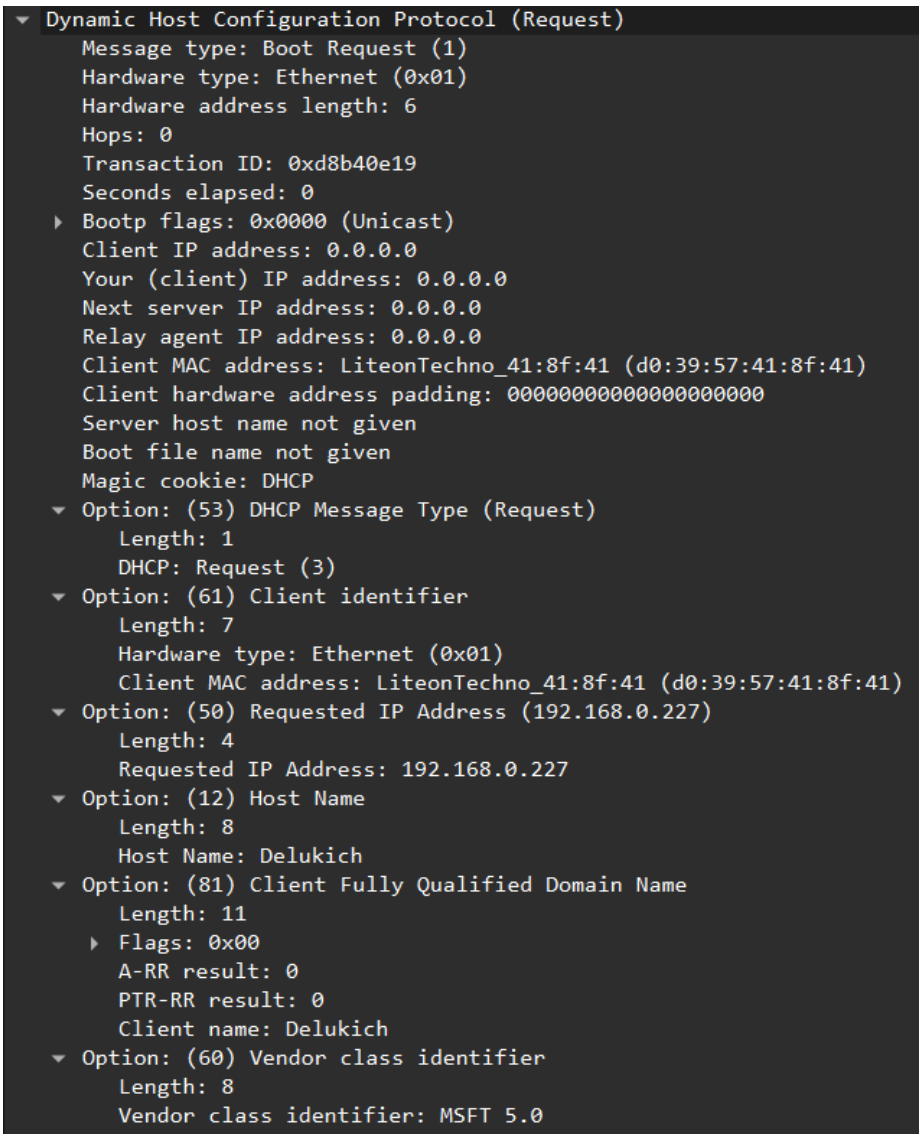
Option 12 – Host Name: Delukich Ім'я комп'ютера, яке клієнт повідомляє серверу. У цьому випадку — Delukich.

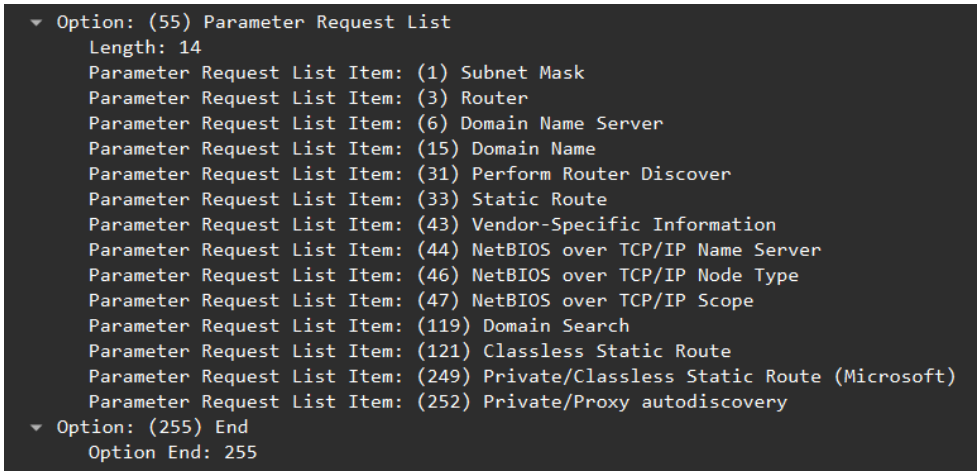
Option 81 – Client Fully Qualified Domain Name (FQDN) Містить повне доменне ім'я клієнта та параметри оновлення записів DNS.
Тут також указано Host Name: Delukich і прапорці A-RR, PTR-RR (результат 0 означає, що записи DNS не оновлювались автоматично).

Option 60 – Vendor Class Identifier: MSFT 5.0 Поле, у якому клієнт повідомляє, яке програмне забезпечення або операційна система виконує запит. Значення MSFT 5.0 означає, що клієнт — Windows-пристрій.

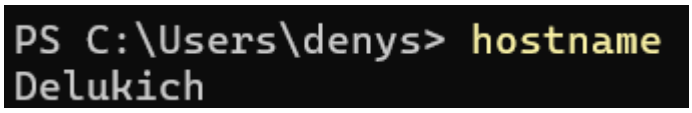
Option (55) – Parameter Request List, яке у моєму випадку має довжину 14 елементів.

Option (255) – End, яка означає завершення передавання DHCP-опцій у цьому повідомленні.





22. Виконав в консолі команду hostname і переконався, що ім’я комп’ютера збігається з іменем у DHCP-запиті та з іменем у **Option 12**.



23. Розгортаючи по черзі опції, зазначив, яка інформація передана у DHCP-відповіді:

Option 53 – DHCP Message Type (ACK) Визначає тип повідомлення. Значення 5 означає Acknowledgement (ACK) — підтвердження з боку сервера, що клієнт може використовувати надану IP-адресу.

Option 54 – DHCP Server Identifier (192.168.0.1) Вказує IP-адресу DHCP-сервера, який обслуговує цей запит. У моєму випадку це 192.168.0.1 — локальний маршрутизатор.

Option 51 – IP Address Lease Time Містить час оренди IP-адреси, тобто тривалість, на яку сервер надає адресу клієнту.
Значення 7200 секунд — це 2 години. Після закінчення цього часу клієнт має оновити оренду.

Option 6 – Domain Name Server Сервер повідомляє адресу DNS-сервера, який клієнт має використовувати для перетворення доменних імен у IP. У моєму випадку DNS = 192.168.0.1.

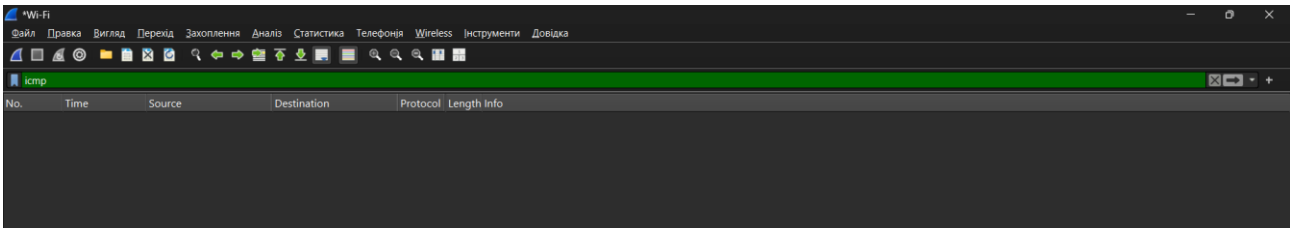
Option 1 – Subnet Mask Визначає маску підмережі — 255.255.255.0. Вона задає, яка частина IP-адреси відповідає мережі, а яка — хостам.

Option 3 – Router Вказує адресу шлюзу за замовчуванням — 192.168.0.1, тобто роутер, через який відбувається вихід у зовнішню мережу.

Option 255 – End Позначає завершення списку DHCP-опцій.

```
▼ Dynamic Host Configuration Protocol (ACK)
    Message type: Boot Reply (2)
    Hardware type: Ethernet (0x01)
    Hardware address length: 6
    Hops: 0
    Transaction ID: 0xd8b40e19
    Seconds elapsed: 0
  ▶ Bootp flags: 0x0000 (Unicast)
    Client IP address: 0.0.0.0
    Your (client) IP address: 192.168.0.227
    Next server IP address: 0.0.0.0
    Relay agent IP address: 0.0.0.0
    Client MAC address: LiteonTechno_41:8f:41 (d0:39:57:41:8f:41)
    Client hardware address padding: 00000000000000000000
    Server host name not given
    Boot file name not given
    Magic cookie: DHCP
  ▼ Option: (53) DHCP Message Type (ACK)
      Length: 1
      DHCP: ACK (5)
  ▼ Option: (54) DHCP Server Identifier (192.168.0.1)
      Length: 4
      DHCP Server Identifier: 192.168.0.1
  ▼ Option: (51) IP Address Lease Time
      Length: 4
      IP Address Lease Time: 2 hours (7200)
  ▼ Option: (6) Domain Name Server
      Length: 4
      Domain Name Server: 192.168.0.1
  ▼ Option: (1) Subnet Mask (255.255.255.0)
      Length: 4
      Subnet Mask: 255.255.255.0
  ▼ Option: (3) Router
      Length: 4
      Router: 192.168.0.1
  ▼ Option: (255) End
      Option End: 255
    Padding […]: 0000000000000000000000000000000000000000000000000000
```

24. Дослідив пакети міжмережевого протоколу керуючих повідомлень ICMP. Застосувавши однойменний фільтр icmp дав порожній результат, адже так просто ICMP-пакети не з’являються:



25. Для перехоплення ICMP-пакетів запустив заново Wireshark і після старту захоплення пакетів скористався консольною утилітою ping, яка працює на основі протоколу ICMP:

```
PS C:\Users\denys> ping google.com

Pinging google.com [142.250.186.174] with 32 bytes of data:
Reply from 142.250.186.174: bytes=32 time=33ms TTL=117
Reply from 142.250.186.174: bytes=32 time=37ms TTL=117
Reply from 142.250.186.174: bytes=32 time=35ms TTL=117
Reply from 142.250.186.174: bytes=32 time=35ms TTL=117

Ping statistics for 142.250.186.174:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 33ms, Maximum = 37ms, Average = 35ms
```

Бачу, що сервер google.com з IP-адресою 142.250.185.174 доступний і середній час відклику дорівнює 35 мс.

26. Для отримання IP-адреси за заданим доменним іменем скористатався командою nslookup:

```
PS C:\Users\denys> nslookup google.com
DNS request timed out.
    timeout was 2 seconds.
Server:  UnKnown
Address:  fe80::404f:5eb0:880a:f3

DNS request timed out.
    timeout was 2 seconds.
DNS request timed out.
    timeout was 2 seconds.
DNS request timed out.
    timeout was 2 seconds.
DNS request timed out.
    timeout was 2 seconds.
*** Request to UnKnown timed-out
```

27. Зупинив захоплення пакетів і, відфільтрував їх з допомогою фільтру icmp, переконався, що тепер результат не порожній:

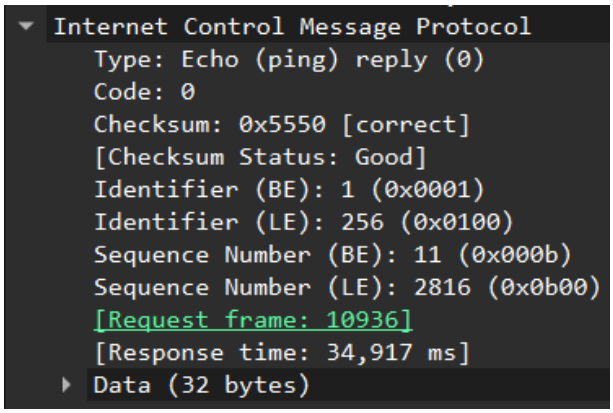
icmp

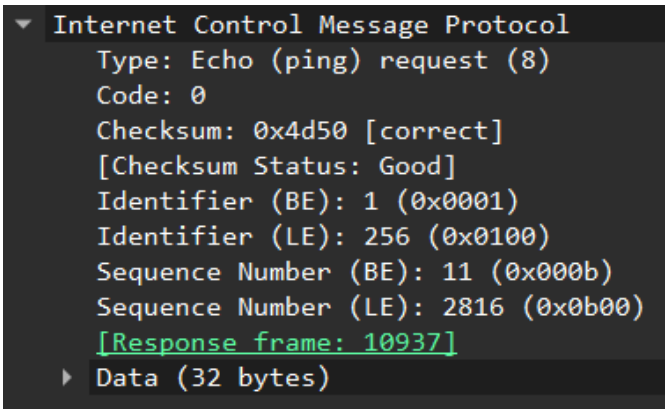
No.	Time	Source	Destination	Protocol	Length	Info
3067	23.386865	192.168.0.227	142.250.186.174	ICMP	74	Echo (ping) request id=0x0001, seq=5/1280, ttl=128 (reply in 3068)
3068	23.436515	142.250.186.174	192.168.0.227	ICMP	74	Echo (ping) reply id=0x0001, seq=5/1280, ttl=117 (request in 3067)
4850	24.396826	192.168.0.227	142.250.186.174	ICMP	74	Echo (ping) request id=0x0001, seq=6/1536, ttl=128 (reply in 4851)
4851	24.429567	142.250.186.174	192.168.0.227	ICMP	74	Echo (ping) reply id=0x0001, seq=6/1536, ttl=117 (request in 4850)
4856	25.407389	192.168.0.227	142.250.186.174	ICMP	74	Echo (ping) request id=0x0001, seq=7/1792, ttl=128 (reply in 4857)
4857	25.441393	142.250.186.174	192.168.0.227	ICMP	74	Echo (ping) reply id=0x0001, seq=7/1792, ttl=117 (request in 4856)
4862	26.429096	192.168.0.227	142.250.186.174	ICMP	74	Echo (ping) request id=0x0001, seq=8/2048, ttl=128 (reply in 4864)
4864	26.462808	142.250.186.174	192.168.0.227	ICMP	74	Echo (ping) reply id=0x0001, seq=8/2048, ttl=117 (request in 4862)

28. *Подумайте і дайте відповідь на запитання "Чому поле TTL у запиті та відповіді мають різні значення?"*

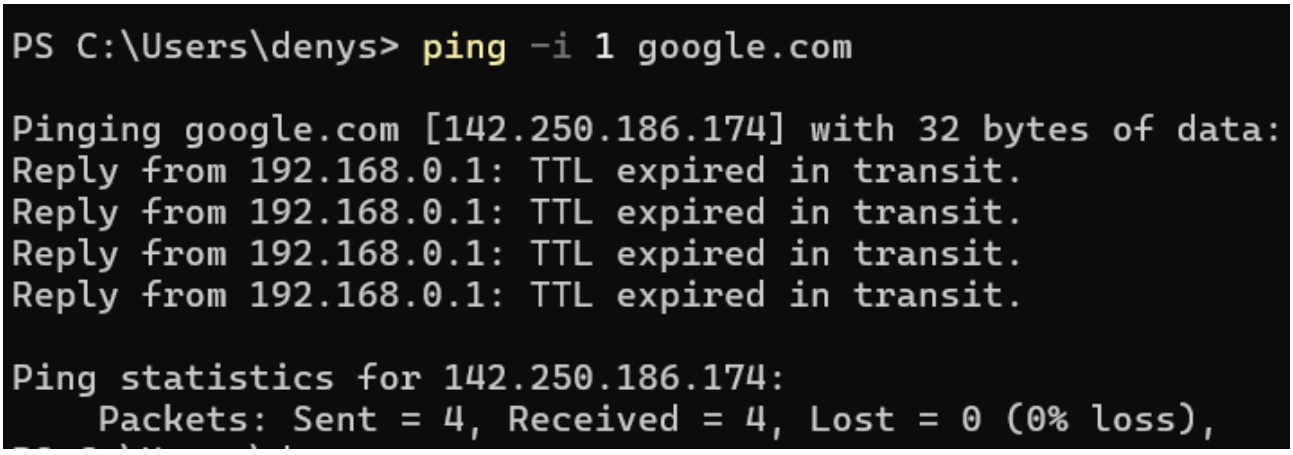
Під час аналізу ICMP-пакетів видно, що поле TTL у запиті й відповіді різне: у запиті — 128, у відповіді — 117.
Це тому, що TTL показує кількість вузлів, через які може пройти пакет. Кожен маршрутизатор зменшує TTL на 1.
Крім того, запит і відповідь формуються різними пристроями, тому мають власні початкові TTL.
Різниця між ними означає, що пакет пройшов приблизно 11 проміжних вузлів у мережі.

29. У попередньому захопленні я отримали 2 типи ICMP-повідомлень: 8 - ехо-запит та 0 - ехо-відповідь:

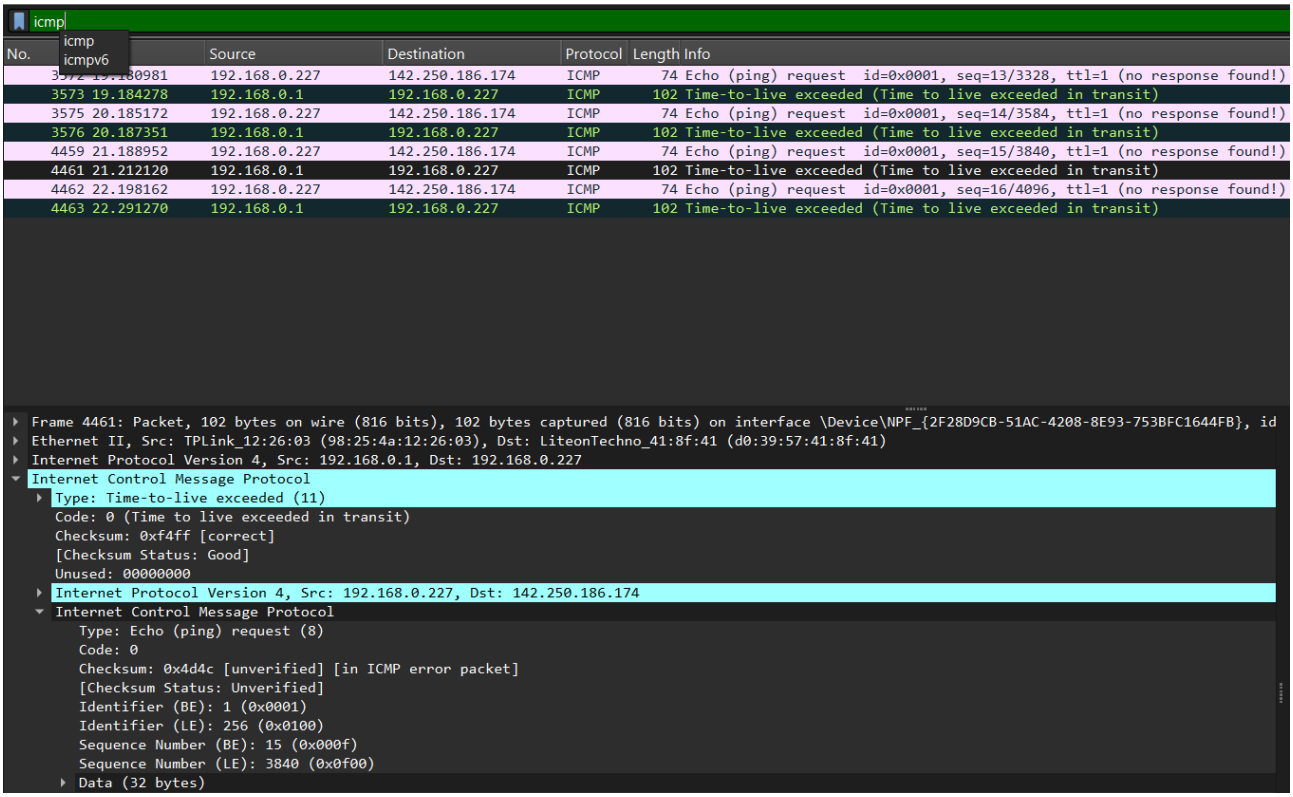




30. Отримав повідомлення іншого типу. Для цього під час запуску утиліти ping задамо час життя пакету TTL = 1:



Бачу, що усі пакети були втрачені. У Wireshark ICMP-відповідь має тип 11 - час життя вийшов:



Під час попереднього захоплення трафіку у відповідях на ICMP-запити (ping) відправником була кінцева адреса сервера — 142.250.186.174 (тобто сам вузол, до якого звертався мій комп’ютер).
У новому захопленні видно, що відповіді надходять від проміжних вузлів з іншими IP-адресами, наприклад 192.168.0.1, а не від кінцевого сервера.

Це пояснюється тим, що цього разу я використовував TTL = 1 у запитах. Через це пакет не може дістатись до кінцевого вузла: він «помирає» вже на першому маршрутизаторі, який і відправляє назад повідомлення “Time to live exceeded in transit”.

Таким чином, відмінність полягає в тому, що:

- у попередніх пакетах відповіді надходили від кінцевого сервера (Echo Reply);
- у поточному прикладі відповіді надходять від проміжних маршрутизаторів (ICMP Time Exceeded), бо TTL закінчується ще під час шляху до призначення.

31. Для визначення маршруту, яким проходить пакет від мене до отримувача, скористатився утилітою tracert. Запустив її у командному рядку, передавши аргументом адресу сайту google.com:

```
PS C:\Users\denys> tracert google.com

Tracing route to google.com [142.250.186.174]
over a maximum of 30 hops:

  1    17 ms    16 ms     3 ms  192.168.0.1
  2     6 ms     4 ms     5 ms  10.1.64.254
  3     *        *        *     Request timed out.
  4     *        *        *     Request timed out.
  5    14 ms    17 ms    13 ms  dtel-ix-2.google.com [193.25.181.62]
  6    12 ms    11 ms    13 ms  74.125.245.83
  7    21 ms    16 ms    95 ms  74.125.245.84
  8    38 ms    33 ms    36 ms  142.251.239.42
  9   150 ms    35 ms    35 ms  142.250.238.131
 10    33 ms    33 ms    33 ms  142.250.214.203
 11    34 ms    34 ms    33 ms  fra24s08-in-f14.1e100.net [142.250.186.174]

Trace complete.
```

Бачу, щоб пакет потрапив до сервера google.com з IP-адресою 142.250.186.174, йому потрібно пройти 11 проміжних маршрутизаторів.

32. Перехопив за допомогою Wireshark ICMP-пакети утиліти tracert/traceroute. Пояснив, як ця утиліта відтворює маршрут до пункту призначення:

Утиліта tracert надсилає серію ICMP-запитів, поступово збільшуючи значення TTL (Time To Live):

- перший пакет має TTL = 1 і «помирає» на першому маршрутизаторі, який надсилає відповідь Time to live exceeded;
- наступний має TTL = 2, тому проходить уже два вузли, і так далі.

Таким чином, кожен маршрутизатор на шляху до призначення відповідає, і програма поступово відтворює весь маршрут до сервера.

33. *Подумайте над запитанням "Чи можна дізнатися маршрут, використовуючи утиліту ping?" та дайте відповідь на нього. Якщо Ваша відповідь "так", то продемонструйте кілька кроків алгоритму.*

Ні, звичайна команда ping не показує маршрут — вона лише перевіряє, чи досяжний сервер і який час відгуку. Щоб дізнатися шлях проходження пакетів, використовується саме tracert, оскільки вони змінюють TTL і фіксують відповіді від кожного проміжного вузла.

Висновок: У ході виконання лабораторної роботи я отримав знання та практичні навички з аналізу мережевих пакетів за допомогою Wireshark, дослідив роботу протоколів IPv4, DHCP та ICMP, навчився визначати параметри мережевої конфігурації, спостерігати процес отримання IP-адреси, а також відстежувати маршрут проходження пакетів за допомогою утиліти tracert.