

**МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ  
 НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ   
імені ІГОРЯ СІКОРСЬКОГО»**

**Навчально-науковий інститут прикладного системного аналізу  
Кафедра системного проектування**

**Звіт**

**про виконання практичної роботи №2  
з дисципліни «Комп’ютерні мережі»**

Виконав:  
студент III курсу, групи ДА-21  
Сімков Микита Дмитрович

Прийняв:

к.н. Гіоргізова-Гай В. Ш.

Київ – 2025

**ПОСТАНОВКА ЗАДАЧІ**

**Мета роботи:**

* Ознайомлення із засобами перехоплення, збереження і аналізу мережевого трафіку за допомогою відомих аналізаторів мережевого трафіку як Wireshark;
* проведення аналізу мережевого сеансу транспортного протоколу TCP.

**Завдання:**

Ознайомтеся з теоретичним відомостями.

Топологія мережі для Виконання роботи:

Інтернет –> маршрутизатор (шлюз за замовчанням) –> ПК

1. Завантажте та установки Wireshark, якщо він ще не встановлений.

2. Запустіть браузер, і в ньому відкриється домашня сторінка.

3. Запустіть програму Wireshark.

*4. Щоб почати роботу*, виберіть інтерфейс. Щоб точно дізнатися IP-адресу і MAC-адресу бажаного інтерфейсу (мережевого адаптера) на ПК, відкрийте вікно командного рядка і введіть ipconfig/all. Запишіть IP-адресу і MAC-адресу інтерфейсу свого комп’ютера.

5. Почніть захоплення пакетів, вибравши в меню Capture команду Start.

6. Завантажте контент з веб-сайту браузером, адреса (http://gaia.cs.umass.edu/

або іншого віддаленого сайту з вітальною сторінкою без безлічі картинок).

7. Після того, як браузер відобразить сторінку, згорніть вікно сайту і поверніться в програму Wireshark.

8. Зупиніть захоплення пакетів, вибравши в меню Capture команду Stop.

*9. Використовуючи меню* Statistics (Capture file properties, Protocol

Hierarchy, Packet leanghs) визначте статистичні дані по захопленим пакетам:

* процентне співвідношення трафіку різних протоколів в мережі;
* середню швидкість біт / сек;
* середню швидкість байт / сек;
* мінімальний, максимальний та середній розміри пакету;
* ступінь використання смуги пропускання каналу (завантаження мережі);
* подивіться по пункту End Points (Ethernet і IPv4), випишіть і поясніть;
* проаналізуйте по IO Gragh поведінку вашого мережевого інтерфейсу.

10. Дослідження Сесії HTTP, ТСР HTTP-повідомлення обміну з HTTP-сервером за адресою gaia.cs.umass.edu або іншого повинні бути в списку захоплених пакетів. Але там буде присутній також безліч інших типів пакетів. Навіть якщо крім завантаження веб-сторінки ви більше нічого не робили, все одно на вашому комп'ютері працює безліч інших процесів і протоколів, а в мережі завжди відбувається набагато більше подій, ніж можна припустити. Якщо комп'ютер був включений недавно і ще не використовувався для доступу до Інтернет, в захоплених даних ви зможете побачити весь процес, включаючи протокол дозволу адрес (ARP), службу доменних імен (DNS) і тристороннє рукостискання TCP. Якщо ПК вже мав запис ARP для вирішення IP-адреси шлюзу за замовчуванням в МАС адресу, тоді він відразу створить DNS запит для дозволу доменного імені сайту (наприклад, gaia.cs.umass.edu) в IP-адресу.

*11. За даними ARP запиту* і відповіді запишіть: IP-адреса і МАС адреса шлюзу вашого комп'ютера.

*12. За даними DNS пакетів* запишіть: IP-адреса DNS-сервера і IP-адреса веб-сервера сайту.

13. Для відображення тільки HTTP-повідомлень у вікні списку пакетів вкажіть значення http (всі імена протоколів в Wireshark пишуться в нижньому регістрі) в поле фільтра відображення. Якщо ви захопили не одну сесію HTTP, скористайтеся фільтрами відображення. В поле фільтра введіть http && ip.addr == x.x.x.x, де x.x.x.x - IP-адреса веб-сервера сайту, і натисніть клавішу Enter.

*14. Визначте, скільки часу пройшло* від моменту відправки повідомлення GET протоколу HTTP до отримання відповідного повідомлення OK? (За замовчуванням, значення поля Time у вікні списку є час в секундах від початку трасування. Ви можете поміняти вид цього поля за вашим бажанням, вибравши в меню View пункт Time Display Format (Формат відображення часу) і потім вказавши відповідне подання часу).

15. Знайдіть повідомлення GET протоколу HTTP, відправлене з вашого комп'ютера на HTTP-сервер (gaia.cs.umass.edu, www.google.com/ або інший), яке містить також введену вами адресу gaia.cs.umass.edu або інший. Виділіть знайдений рядок з повідомленням HTTP GET, у вікні деталей заголовків з'явиться інформація по заголовкам всіх нижчих протоколів, в які вкладено GET-повідомлення протоколу HTTP (міститься в сегменті TCP, який, в свою чергу, знаходиться (інкапсулюється) в IP- дейтаграмі, яка інкапсулюється в кадр Ethernet).

*16. Користуючись кнопками + і -* в лівій частині вікна деталей заголовків, по черзі відкривайте інформацію про кадри Ethernet, дейтаграми IP, сегменті TCP, повідомлення HTTP, скопіюйте в протокол і поясніть (коротко опишіть) призначення основних полів.

*17. Проаналізуйте і опишіть повну сесію HTTP* (без деталей кожного заголовка) і TCP (детально), використовуючи інформацію з поля списку пакетів і Flow Graph з меню Statistics. Внизу вікна Flow Graph є перемикач Flow Type для перемикання відображення загальної сесії HTTP і тільки TCP. Збережіть діаграму взаємодії машин. Опис сесії TCP повинен включати:

* IP-адреси джерела і призначення;
* номера портів джерела і призначення;
* процедуру встановлення з'єднання, передачі даних і розриву з'єднання (номери пакетів і значення прапорців управління TCP);
* метод ковзного вікна (зміна значень відносних послідовних номерів і номерів підтвердження і розміру вікна). Це краще видно не на графі, а у полі списку пакетів.

*18. Дослідіть сесію* інструментами з TCP Stream Graph з меню Statistics. Поясніть графіки, покажіть зміну довжини ковзаючого вікна у відправника і одержувача (див. приклади в файлі «Робота з Wireshark»).

19. Збережіть результати захоплення в файлі для можливості подальшого перегляду.

**ХІД РОБОТИ**

Після завантаження Wireshark обираю мережевий інтерфейс Ethernet.

Також запишу свій IPv4 та фізичний адрес:

Фізичний адрес: 00-D8-61-19-0F-D8

IPv4: 192.168.88.20

Проведу захоплення пакетів:

A screenshot of a computer

AI-generated content may be incorrect.

Рис. 1 – захоплені пакети.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screen shot of a graph

AI-generated content may be incorrect.

Рис. 2-5 - Capture File Properties; Protocol

Hierarchy; Packet Lengths; I/O Graphs

Процентне співвідношення протоколів за кількістю пакетів:

* **IPv4:** 98.9% (1703 пакети) — основна частина трафіку використовує IPv4.
* **TCP:** 56% (965 пакетів) — найбільший внесок серед транспортних протоколів.
* **UDP:** 42.8% (738 пакетів) — помітна частка, ймовірно, через QUIC.
* **QUIC:** 35.4% (609 пакетів) — сучасний протокол для швидкого з’єднання.
* **DNS:** 7.08% (122 пакети) — типовий рівень для обробки доменних запитів.
* **TLS:** 20.7% (357 пакетів) — шифрування трафіку, наприклад, для HTTPS.
* **ARP:** 0.58% (10 пакетів) — для отримання MAC-адрес.
* **IPv6:** 0.17% (3 пакети) — незначний обсяг трафіку через IPv6.

Аналіз за обсягом даних (Bytes):

* **TLS:** 59.8% — найбільша частина трафіку через зашифровані з’єднання (537,161 байт).
* **QUIC:** 31.3% — вагомий внесок через UDP, що показує сучасний веб-трафік (281,099 байт).
* **TCP:** 2.26% — хоча кількість пакетів велика, обсяг даних відносно невеликий (20,280 байт).
* **DNS:** 2.08% — важливо для розв’язання доменів (18,704 байт).
* **Ethernet:** 2.83% — невелика частка, оскільки сам заголовок Ethernet короткий.
* **IPv4:** 3.79% — частка IP-адрес та маршрутної інформації (34,060 байт).

Висновки:

* Основна частина трафіку проходить через **TCP** та **UDP** — це класичний сценарій для веб-запитів (HTTPS, QUIC).
* Протоколи **QUIC** та **TLS** домінують за обсягом даних, що свідчить про активне використання сучасних зашифрованих з’єднань.
* **IPv6** майже не використовується — більшість трафіку йде через **IPv4**.
* **ARP** та **LLC** займають мізерну частку — це очікувано, адже ці протоколи лише допоміжні.

144,5 пзс та 75к байт за секунду

Мінімальний розмір пакету 54, максимальний 14454, середній 521

Так як швидкість мережі 100Mb за секунду, то середня навантаженість

завантаження мережі.

Пункт End Points:

A screenshot of a computer

AI-generated content may be incorrect.

Рис. 6 – EndPoints

Активні учасники трафіку

* Найбільше трафіку (1702 пакети, 897135 байт) проходить через вузол **00:d8:61:19:0f:d8** — він активно відправляє та приймає дані, це основний пристрій у мережі – мій комп’ютер.
* **cc:2d:e0:1a:6e:d3** теж генерує значний трафік (1696 пакетів), передаючи майже рівномірно у двох напрямках, це мій роутер – основний маршрутизатор в цій мережі.

Широкомовні та мультикаст-адреси

* **01:00:5e:00:00:fb** і **33:33:00:00:00:fb** — це мультикаст-адреси, що використовуються для групового зв’язку (наприклад, для DNS-сервісів, mDNS).
* **01:80:c2:00:00:00** — це адреса, пов’язана зі Spanning Tree Protocol (STP), що використовується для запобігання петлям у комутаційній мережі.

Односторонній трафік

* **50:91:e3:3d:6c:4d** передав 10 пакетів, але нічого не отримав — можливо, це ARP-запити або трансляція даних без відповіді.

Аналіз поведінки мережевого інтерфейсу за I/O Graph у Wireshark

A screen shot of a graph

AI-generated content may be incorrect.

A screen shot of a graph

AI-generated content may be incorrect.

Рис. 7, 8 – I/O Graphs

Оскільки захоплення було розпочато майже зі стартом системи, можна побачити підвищення кількості пзс, також всередині можна побачити червоний графік з днс запитами, це пов’язано з вмиканням браузера та відкриттям сторінки. Також є червона лінія ("TCP Errors"), яка показує що наявні помилки в TCP з’єднаннях,

Зайдемо на сторінку <http://gaia.cs.umass.edu/> і подивимось як будуть проходити запити.  
A screenshot of a computer

AI-generated content may be incorrect.

Рис. 9 – HTTP запити.

Як можна побачити спочатку був створений GET запит, а потім отримано код 200 та відповідь.

Подивимось на TCP:

A screenshot of a computer

AI-generated content may be incorrect.

Рис. 10 – Потрійне рукостискання TCP.

На 2.49 було відправлено запит на встановлення зв’язку між комп’ютером і веб-сервером. На 2.60 було отримано відповідь на запит. В наступному запиті на 2.60 було підтверджено встановлення зв’язку.

ARP запити:

A screenshot of a computer

AI-generated content may be incorrect.

Рис. 11 – ARP запити.

Sender MAC address: MicroStarINT\_19:0f:d8 (00:d8:61:19:0f:d8)

Sender IP address: 192.168.88.20

DNS запити:

A screenshot of a computer

AI-generated content may be incorrect.

Рис. 12 – DNS запити

IP серверу – 128.119.245.12, DNS сервер у нашому випадку – 192.168.0.1, тобто шлюз.

A screenshot of a computer

AI-generated content may be incorrect.

Рис. 13 – відповідь DNS сервера.

Відповідь прийшла від DNS сервера. В відповіді можна побачити, що для сайту: Address: 128.119.245.12 та для DNS сервера: Address: 217.20.184.1.

Подивимось запити до веб-серверу.

A screenshot of a computer

AI-generated content may be incorrect.

Рис. 14 – Фільтр HTTP запитів за IP адресою веб-серверу.

Різниця між GET та 200 ОК – 0.11мс.

Подивлюсь інформацію про http запит:

Frame 499: 507 bytes on wire (4056 bits), 507 bytes captured (4056 bits) on interface \Device\NPF\_{CCE4C272-E3F4-466C-90A5-263CA64E9393}, id 0

Section number: 1

Interface id: 0 (\Device\NPF\_{CCE4C272-E3F4-466C-90A5-263CA64E9393})

Interface name: \Device\NPF\_{CCE4C272-E3F4-466C-90A5-263CA64E9393}

Interface description: Ethernet

Encapsulation type: Ethernet (1)

Arrival Time: Mar 3, 2025 11:37:30.157379000 Финляндия (зима)

UTC Arrival Time: Mar 3, 2025 09:37:30.157379000 UTC

Epoch Arrival Time: 1740994650.157379000

[Time shift for this packet: 0.000000000 seconds]

[Time delta from previous captured frame: 0.000103000 seconds]

[Time delta from previous displayed frame: 0.000103000 seconds]

[Time since reference or first frame: 2.602741000 seconds]

Frame Number: 499

Frame Length: 507 bytes (4056 bits)

Capture Length: 507 bytes (4056 bits)

[Frame is marked: False]

[Frame is ignored: False]

[Protocols in frame: eth:ethertype:ip:tcp:http]

[Coloring Rule Name: HTTP]

[Coloring Rule String: http || tcp.port == 80 || http2]

Ethernet II, Src: MicroStarINT\_19:0f:d8 (00:d8:61:19:0f:d8), Dst: Routerboardc\_1a:6e:d3 (cc:2d:e0:1a:6e:d3)

Destination: Routerboardc\_1a:6e:d3 (cc:2d:e0:1a:6e:d3)

.... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)

.... ...0 .... .... .... .... = IG bit: Individual address (unicast)

Source: MicroStarINT\_19:0f:d8 (00:d8:61:19:0f:d8)

.... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)

.... ...0 .... .... .... .... = IG bit: Individual address (unicast)

Type: IPv4 (0x0800)

[Stream index: 0]

Internet Protocol Version 4, Src: 192.168.88.20, Dst: 128.119.245.12

0100 .... = Version: 4

.... 0101 = Header Length: 20 bytes (5)

Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)

0000 00.. = Differentiated Services Codepoint: Default (0)

.... ..00 = Explicit Congestion Notification: Not ECN-Capable Transport (0)

Total Length: 493

Identification: 0x6368 (25448)

010. .... = Flags: 0x2, Don't fragment

0... .... = Reserved bit: Not set

.1.. .... = Don't fragment: Set

..0. .... = More fragments: Not set

...0 0000 0000 0000 = Fragment Offset: 0

Time to Live: 128

Protocol: TCP (6)

Header Checksum: 0x0000 [validation disabled]

[Header checksum status: Unverified]

Source Address: 192.168.88.20

Destination Address: 128.119.245.12

[Stream index: 13]

Transmission Control Protocol, Src Port: 64938, Dst Port: 80, Seq: 1, Ack: 1, Len: 453

Source Port: 64938

Destination Port: 80

[Stream index: 6]

[Stream Packet Number: 4]

[Conversation completeness: Incomplete, DATA (15)]

..0. .... = RST: Absent

...0 .... = FIN: Absent

.... 1... = Data: Present

.... .1.. = ACK: Present

.... ..1. = SYN-ACK: Present

.... ...1 = SYN: Present

[Completeness Flags: ··DASS]

[TCP Segment Len: 453]

Sequence Number: 1 (relative sequence number)

Sequence Number (raw): 3109134500

[Next Sequence Number: 454 (relative sequence number)]

Acknowledgment Number: 1 (relative ack number)

Acknowledgment number (raw): 2353453855

0101 .... = Header Length: 20 bytes (5)

Flags: 0x018 (PSH, ACK)

000. .... .... = Reserved: Not set

...0 .... .... = Accurate ECN: Not set

.... 0... .... = Congestion Window Reduced: Not set

.... .0.. .... = ECN-Echo: Not set

.... ..0. .... = Urgent: Not set

.... ...1 .... = Acknowledgment: Set

.... .... 1... = Push: Set

.... .... .0.. = Reset: Not set

.... .... ..0. = Syn: Not set

.... .... ...0 = Fin: Not set

[TCP Flags: ·······AP···]

Window: 513

[Calculated window size: 131328]

[Window size scaling factor: 256]

Checksum: 0x9020 [unverified]

[Checksum Status: Unverified]

Urgent Pointer: 0

[Timestamps]

[Time since first frame in this TCP stream: 0.108134000 seconds]

[Time since previous frame in this TCP stream: 0.000103000 seconds]

[SEQ/ACK analysis]

[iRTT: 0.108031000 seconds]

[Bytes in flight: 453]

[Bytes sent since last PSH flag: 453]

TCP payload (453 bytes)

Hypertext Transfer Protocol

GET / HTTP/1.1\r\n

Request Method: GET

Request URI: /

Request Version: HTTP/1.1

Host: gaia.cs.umass.edu\r\n

Connection: keep-alive\r\n

Upgrade-Insecure-Requests: 1\r\n

User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/133.0.0.0 Safari/537.36\r\n

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,\*/\*;q=0.8,application/signed-exchange;v=b3;q=0.7\r\n

Accept-Encoding: gzip, deflate\r\n

Accept-Language: uk-UA,uk;q=0.9,en-US;q=0.8,en;q=0.7\r\n

\r\n

[Response in frame: 512]

[Full request URI: http://gaia.cs.umass.edu/]

Пояснення запиту:

Комп’ютер (MAC: 00:d8:61:19:0f:d8, IP: 192.168.88.20) намагається підключитися до веб-сервера gaia.cs.umass.edu (IP: 128.119.245.12) через HTTP (порт 80).

Що відбувається:

1. Ethernet рівень:
   * Пакет йде через роутер (MAC: cc:2d:e0:1a:6e:d3).
2. IP рівень:
   * Джерело: 192.168.88.20 (ПК).
   * Призначення: 128.119.245.12 (сервер в інтернеті).
3. TCP:
   * Комп’ютер встановлює TCP-з'єднання (SYN, SYN-ACK, ACK).
   * Надсилає HTTP-запит методом GET для отримання головної сторінки.
4. HTTP:
   * Браузер (Chrome 133) просить головну сторінку сайту.
   * Вказує заголовки: приймає HTML, підтримує gzip-стиснення, вказує мову (українська та англійська).

Повний запит: <http://gaia.cs.umass.edu/>. З’єднання було ініційовано комп’ютером — це вихідний трафік, а не відповідь від сервера. Відповідь прийде у наступних кадрах (посилання на відповідь: Frame 512).

Тепер переглянемо:

A screenshot of a computer screen

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

Рис. 15,16 – flow Graph

Проаналізуємо запити до сервера:

* Порт джерела: 64938 (випадковий порт, призначений клієнтом)
* Порт призначення: 80 (порт HTTP)

1. З’єднання починається з TCP 3-way handshake:  
   SYN (Frame 476):  
   Клієнт 192.168.88.20 відправляє SYN-пакет до сервера 128.119.245.12 на порт 80.
   * Seq = 0
   * Window: 64240
   * MSS = 1460
   * Window Scale = 8
   * SACK permitted
2. SYN-ACK (Frame 497):  
   Сервер відповідає SYN-ACK пакетом, підтверджуючи отримання SYN.
   * Seq = 0
   * Ack = 1
   * Win = 29200
   * MSS = 1460
   * Window Scale = 7
3. ACK (Frame 498):  
   Клієнт надсилає фінальне підтвердження ACK.
   * Seq = 1
   * Ack = 1
   * Win = 513

Після 3 повідомлення TCP з’єднання встановлено.

**Передача HTTP-запиту**

Після встановлення з’єднання клієнт надсилає запит на отримання сторінки (Frame 273):

* Метод: GET
* URI: /
* Версія: HTTP/1.1
* Хост: gaia.cs.umass.edu
* User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/133.0.0.0 Safari/537.36\r\n
* Заголовки: Accept, Accept-Encoding, Connection: keep-alive

Запит міститься у TCP-сегменті з Seq = 1, Ack = 1, Len = 453.

**Отримання HTTP-відповіді**

Сервер відповідає у Frame 512:

* Статус: 200 OK
* Тип контенту: text/html
* Довжина контенту: 2651 байт
* Сервер: Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips PHP/7.4.33
* Кодування: UTF-8
* Час відповіді: 0.11 секунди

HTTP-відповідь передається в кількох TCP-сегментах:

* Frame 510 (1460 байт)
* Frame 511 (1460 байт)
* Frame 512 (91 байт, завершення відповіді)

Клієнт підтверджує отримання кожного сегмента через ACK-пакети.

Оскільки під час захоплення пакетів сайт не було закрито, не було захоплено пакет FIN, ACK для закриття з’єднання.

**Завершення TCP-з'єднання**

Після отримання контенту відбувається процес завершення TCP-з'єднання:

1. Клієнт надсилає FIN.
2. Сервер підтверджує ACK.
3. Сервер надсилає FIN.
4. Клієнт підтверджує ACK, завершуючи сесію.

**Використання механізму ковзного вікна**

TCP використовує механізм Sliding Window для більш ефективного керування потоком даних.

* Початковий розмір вікна клієнта: 131328
* Початковий розмір вікна сервера: 30336
* Дані передаються у фрагментах 1460 байт, що відповідає MSS (Maximum Segment Size).
* Клієнт надсилає ACK після отримання всіх блоків.

Завдяки ковзному вікну оптимізується швидкість передачі даних.

Графіки з TCP Stream Graph:

A graph with numbers and a line

AI-generated content may be incorrect.

Рис. 17 – Графік Stevens.

Кожна точка відповідає отриманому ACK. Чіткі сходинки означають збільшення переданих байтів у відповідь на ACK.

A graph with numbers and a line

AI-generated content may be incorrect.

Рис. 18 –Sequence Numbers.

Зростання кривої означає передачу нових сегментів даних.

Плоскі ділянки вказують на відсутність нових сегментів у передаванні.

На графіку видно стрибкоподібне зростання, що свідчить про надсилання пакетів великими порціями по кожному новому з'єднанню.

A graph with numbers and lines

AI-generated content may be incorrect.

Рис. 19 – Графік Throughput.

Відображає обсяг переданих даних у часі. Плоскі ділянки – це періоди відсутності передавання (очікування відповіді). Зростання на початку – пришвидшення передачі.

A graph with numbers and a line

AI-generated content may be incorrect.

Рис. 20 – Графік RTT.

Відображає зміну часу RTT у мілісекундах. Лінійне зростання RTT свідчить про стабільну затримку в мережі.

A graph with numbers and a line

AI-generated content may be incorrect.

Рис. 21 – Графік Window Scaling.

Показує зміну розміру приймального вікна. Збільшення вікна означає, що приймач може обробляти більші обсяги даних.

**Висновки**

В лабораторній роботі було проведено захоплення та аналіз мережевого трафіку після запуску комп’ютера та під час використання браузера. Також було наглядно проаналізовано з’єднання та передачу даних між клієнтом та сервером.

Під час роботи вдалося зафіксувати TCP-з'єднання через процедуру Three-way Handshake (SYN, SYN-ACK, ACK), дослідити механізм ковзного вікна та масштабування вікна. Аналіз номерів послідовностей показав стабільну передачу даних без значних затримок чи втрат. Вимірювання RTT засвідчило сталість часу проходження пакетів, а графік пропускної здатності відобразив зростання швидкості передавання з подальшою стабілізацією.

Окрім цього, було проаналізовано HTTP-сесію, де час відправки запиту до отримання відповіді склав 0,11 с — прийнятний показник для веб-додатків. Ідентифікація ARP- і DNS-запитів підтвердила коректний процес резолюції IP-адрес та маршрутизації в локальній мережі.

Таким чином, дослідження підтвердило ефективність роботи TCP у межах даної мережевої взаємодії. Використання Wireshark продемонструвало можливість не лише аналізувати роботу транспортного протоколу, а й виявляти потенційні проблеми з передачею даних, що є важливим для підтримання стабільності та продуктивності мережі.