

Міністерство освіти і науки України
Національний технічний університет України
«Київський політехнічний інститут»

Лабораторна робота №2
з дисципліни «Комп'ютерні мережі»

**«Аналіз просування даних по стеку TCP/IP
з використанням аналізатора трафіку
Wireshark. Транспортний і мережевий
рівні»**

Виконала студентка групи: KB-11

ПІБ: Михайліченко Софія Віталіївна

Перевірив: _____

Мета роботи:

Засвоєння функцій модулів транспортного та мережевого рівнів стеку протоколів TCP/IP, структури заголовків протоколів TCP та UDP, псевдозаголовку, аналіз фрагментів протоколу TCP за допомогою аналізатора мережевого трафіку Wireshark.

План виконання лабораторної роботи:

1. Ознайомитись та засвоїти теоретичні відомості, викладені в методичному посібнику до лабораторної роботи.
2. Виконати завдання до лабораторної роботи.

Завдання:

1. За допомогою програми Wireshark необхідно виконати захоплення даних сеансу FTP і визначити значення полів заголовків протоколу TCP при передачі файлів з використанням протоколу FTP між хост-комп'ютером і анонімним FTP-сервером. Під'єднання до анонімного FTP-серверу і завантаження файлу виконується за допомогою браузера.

2.

2.1 Ознайомитись з можливостями фільтрації даних за різними ознаками, зокрема, за MAC-адресою відправника і отримувача. Фільтр створюється за описаною вище методикою. Відповідно до рекомендацій викладача сформувати фільтр за MAC-адресою.

2.2 Розглянути результат інкапсуляції при передачі даних. В захоплених пакетах виділити службову інформацію (заголовки) всіх блоків даних, а також, за наявності, кінцевика.

2.3 Використовуючи фільтр відображення `tcp.flags.syn == 1` відібрати сегменти-запити, які містять встановлений прапорець SYN у заголовку та сегменти-відповіді, які містять встановлені прапорці SYN та ACK. Провести аналіз поля Options заголовку TCP. Яке значення MSS використовується в з'єднанні, що аналізується?

2.4 За допомогою меню «Statistics» необхідно отримати і додати до звіту таку

інформацію:

- кількість захоплених пакетів та байтів;
- середня швидкість передачі даних (в бітах за секунду);
- середній розмір пакета;
- час, протягом якого здійснювалось захоплення трафіку;
- вивести таблицю Ethernet Conversations та пояснити вміст її рядків;
- вивести IO Graphs, за допомогою якого визначити пікову швидкість передачі даних протягом інтервалу, що підлягає

аналізу.

За результатами роботи зробити висновки.

Короткі теоретичні відомості:

При передачі даних по стеку TCP/IP повідомлення розбивається на менші частини — сегменти, залежно від обмежень каналу передачі. Кожен рівень додає заголовки, що містять службову інформацію для коректної передачі даних. На прикладному рівні використовуються такі протоколи, як FTP, Telnet, DNS, які обирають між передачею даних у вигляді безперервного потоку або окремих повідомлень.

1. Протокол UDP. Заголовок UDP-сегменту

UDP (User Datagram Protocol) використовується для швидкої передачі даних без встановлення з'єднання. Він не гарантує доставку і не відстежує порядок повідомлень. Заголовок UDP-дейтаграми містить такі поля:

- **Source Port** (порт відправника),
- **Destination Port** (порт отримувача),
- **Length** (довжина дейтаграми),
- **Checksum** (контрольна сума).

UDP підходить для коротких повідомлень або коли точна доставка не є критичною.

2. Протокол TCP. Заголовок TCP-сегменту

TCP (Transmission Control Protocol) забезпечує надійну передачу даних. Його заголовок складається з таких основних полів:

- **Source Port і Destination Port** — порти відправника та отримувача.
- **Sequence Number** — порядковий номер першого байта в сегменті.
- **Acknowledgment Number** — підтверджує отримання даних.
- **Flags** — прапорці для керування з'єднанням (SYN, ACK, FIN, RST тощо).
- **Window Size** — кількість байтів, які можуть бути передані без підтвердження.
- **Checksum** — контрольна сума для перевірки цілісності даних.

TCP використовує тристороннє квітування (SYN-SYN/ACK-ACK) для встановлення з'єднання, а також процес коректного закриття з'єднання після передачі даних. TCP гарантує, що дані доставляються в правильному

порядку, без втрат і дублювань.

3. Мережевий рівень і протокол IP

На мережевому рівні функціонує протокол **IP** (Internet Protocol), який відповідає за передачу даних у вигляді дейтаграм між пристроями. Заголовок IP-дейтаграми містить:

- **Version** — версія протоколу IP (IPv4 або IPv6),
- **Source Address** і **Destination Address** — IP-адреси відправника і отримувача,
- **Time to Live (TTL)** — лічильник, що зменшується на кожному маршрутизаторі і запобігає зацикленню пакету,
- **Protocol** — вказує на протокол транспортного рівня (TCP або UDP). Протокол IP не гарантує доставку даних і не перевіряє послідовність.

Протокол IP відповідає за передачу даних через мережу, але не гарантує їх надійної доставки або збереження послідовності.

Wireshark — це інструмент для аналізу мережевого трафіку, що дозволяє захоплювати пакети і досліджувати їх заголовки. За допомогою фільтрів можна виділяти необхідні сегменти для детального аналізу.

Порядок виконання роботи:

Завдання 1.

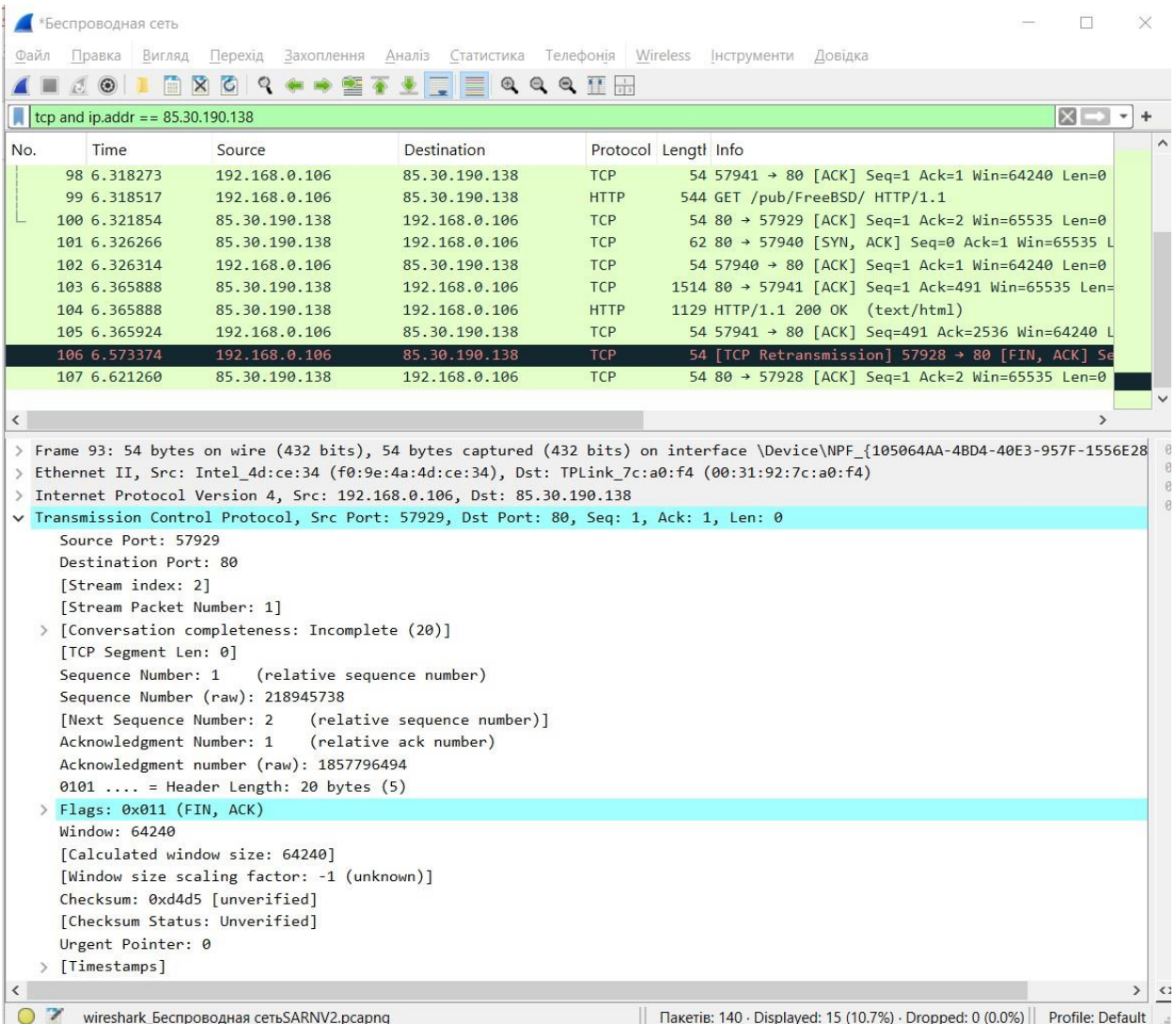
Підключаємось до FTP-сервера через браузер за адресою

<http://ftp4.freebsd.org/pub/FreeBSD/>:

Index of /pub/FreeBSD/

File Name ↓	File Size ↓	Date ↓
Parent directory/	-	-
development/	-	2024-Oct-12 12:15
doc/	-	2024-Oct-07 23:00
ports/	-	2022-Nov-10 15:01
releases/	-	2024-Oct-12 12:15
snapshots/	-	2024-Aug-22 08:10
README.TXT	4259	2015-May-07 16:18
TIMESTAMP	35	2024-Oct-12 12:15
dir.sizes	2929	2024-Oct-12 10:00

Для виконання даного завдання активізуємо режим захоплення даних з використанням програми Wireshark. Тобто починаємо захоплення і відкриваємо файл README.TXT. Після завантаження файлу завершуємо захоплення. Застосуємо до отриманих даних фільтр *tcp and ip.addr == 85.30.190.138*:



Wireshark interface showing a network capture filtered by `tcp and ip.addr == 85.30.190.138`. The packet list shows a TCP retransmission (packet 106) from 192.168.0.106 to 85.30.190.138. The packet details pane shows the structure of the TCP segment, including source and destination ports, sequence numbers, and flags (FIN, ACK).

Packet 106: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface \Device\NPF_{105064AA-4BD4-40E3-957F-1556E28...}

Ethernet II, Src: Intel_4d:ce:34 (f0:9e:4a:4d:ce:34), Dst: TPLink_7c:a0:f4 (00:31:92:7c:a0:f4)

Internet Protocol Version 4, Src: 192.168.0.106, Dst: 85.30.190.138

Transmission Control Protocol, Src Port: 57929, Dst Port: 80, Seq: 1, Ack: 1, Len: 0

Source Port: 57929
Destination Port: 80
[Stream index: 2]
[Stream Packet Number: 1]
[Conversation completeness: Incomplete (20)]
[TCP Segment Len: 0]
Sequence Number: 1 (relative sequence number)
Sequence Number (raw): 218945738
[Next Sequence Number: 2 (relative sequence number)]
Acknowledgment Number: 1 (relative ack number)
Acknowledgment number (raw): 1857796494
0101 = Header Length: 20 bytes (5)
Flags: 0x011 (FIN, ACK)
Window: 64240
[Calculated window size: 64240]
[Window size scaling factor: -1 (unknown)]
Checksum: 0xd4d5 [unverified]
[Checksum Status: Unverified]
Urgent Pointer: 0
[Timestamps]

Source Port (Порт отримувача): 80 – це порт, через який відправник ініціює

з'єднання, і зазвичай використовується для HTTP-запитів.

Destination Port (Порт відправника): 57929 – це порт, до якого спрямовано з'єднання на стороні отримувача.

Sequence Number (Номер послідовності): 1 – це порядковий номер першого октету в даному сегменті.

Acknowledgment Number (Номер підтвердження): 1 – це наступний очікуваний октет від отримувача.

Flags (Прапорці): 0x011 (ACK)

ACK (Acknowledgment): Позначає, що поле підтвердження є дійсним.

Window Size (Розмір вікна): 64240 – це значення вікна зміщення. Вказує на кількість байт, які відправник може відправити до отримання підтвердження.

Urgent Pointer (Вказівник термінованого пакета): 0 – вказівник, який використовується тільки в разі встановлення прапорця URG. У даному випадку не використовується.

Це TCP-сегмент з використанням стандартного HTTP-з'єднання на порт 80 із зазначеним підтвердженням та встановленим розміром вікна 64240.

Завдання 2.

Відкриємо панель відомостей про пакети та подивимось MAC-адресу:

```
> Frame 93: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface \Device\NPF_{105064AA-4BD4-40E3-957F-1556F...}
  Ethernet II, Src: Intel_4d:ce:34 (f0:9e:4a:4d:ce:34), Dst: TPLink_7c:a0:f4 (00:31:92:7c:a0:f4)
    > Destination: TPLink_7c:a0:f4 (00:31:92:7c:a0:f4)
    > Source: Intel_4d:ce:34 (f0:9e:4a:4d:ce:34)
    Type: IPv4 (0x0800)
    [Stream index: 0]
```

Використаємо її, як фільтр (*eth.addr == 00:31:92:7c:a0:f4*):

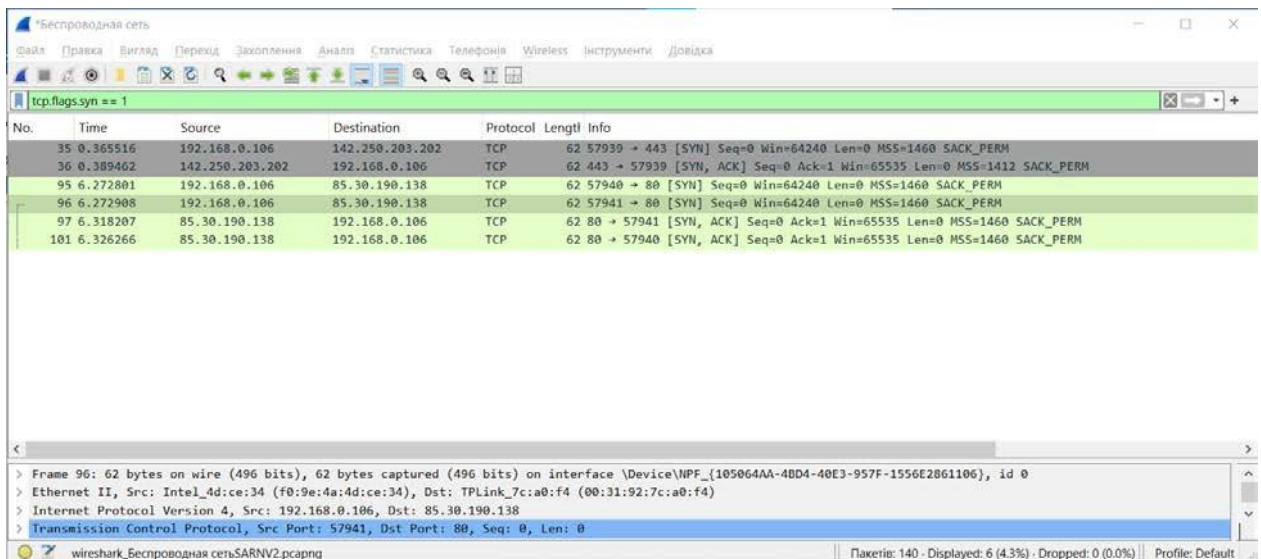
Wireshark interface showing a packet list filtered by MAC address. The filter `eth.addr == 00:31:92:7c:a0:f4` is applied. The packet list shows various protocols including TCP, SSL, and DNS. The packet details pane shows the structure of a selected packet.

No.	Time	Source	Destination	Protocol	Length	Info
24	0.271811	192.168.0.106	149.154.167.41	TCP	54	57935 → 443 [ACK] Seq=426 Ack=24461 Win=64480
25	0.273020	149.154.167.41	192.168.0.106	SSL	1294	Continuation Data
26	0.273020	149.154.167.41	192.168.0.106	SSL	1294	Continuation Data
27	0.273050	192.168.0.106	149.154.167.41	TCP	54	57935 → 443 [ACK] Seq=426 Ack=26941 Win=64480
28	0.275576	149.154.167.41	192.168.0.106	SSL	1026	Continuation Data
29	0.290438	192.168.0.106	149.154.167.41	SSL	239	Continuation Data
30	0.324537	149.154.167.41	192.168.0.106	TCP	60	443 → 57935 [ACK] Seq=27913 Ack=611 Win=65535
31	0.362596	192.168.0.106	192.168.0.1	DNS	87	Standard query 0x5b52 A safebrowsing.googleap
32	0.362850	192.168.0.106	192.168.0.1	DNS	87	Standard query 0x9d31 HTTPS safebrowsing.goog
33	0.364778	192.168.0.1	192.168.0.106	DNS	144	Standard query response 0x9d31 HTTPS safebrow
34	0.365171	192.168.0.1	192.168.0.106	DNS	246	Standard query response 0x5b52 A safebrowsing
35	0.365516	192.168.0.106	142.250.203.202	TCP	62	57939 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1
36	0.389462	142.250.203.202	192.168.0.106	TCP	62	443 → 57939 [SYN, ACK] Seq=0 Ack=1 Win=65535
37	0.389525	192.168.0.106	142.250.203.202	TCP	54	57939 → 443 [ACK] Seq=1 Ack=1 Win=64240 Len=0
38	0.390012	192.168.0.106	142.250.203.202	TLSv1.3	1855	Client Hello (SNI=safebrowsing.googleapis.com
39	0.406593	142.250.203.202	192.168.0.106	TCP	60	443 → 57939 [ACK] Seq=1 Ack=1413 Win=65535 Le

Wireshark details pane for Frame 29:

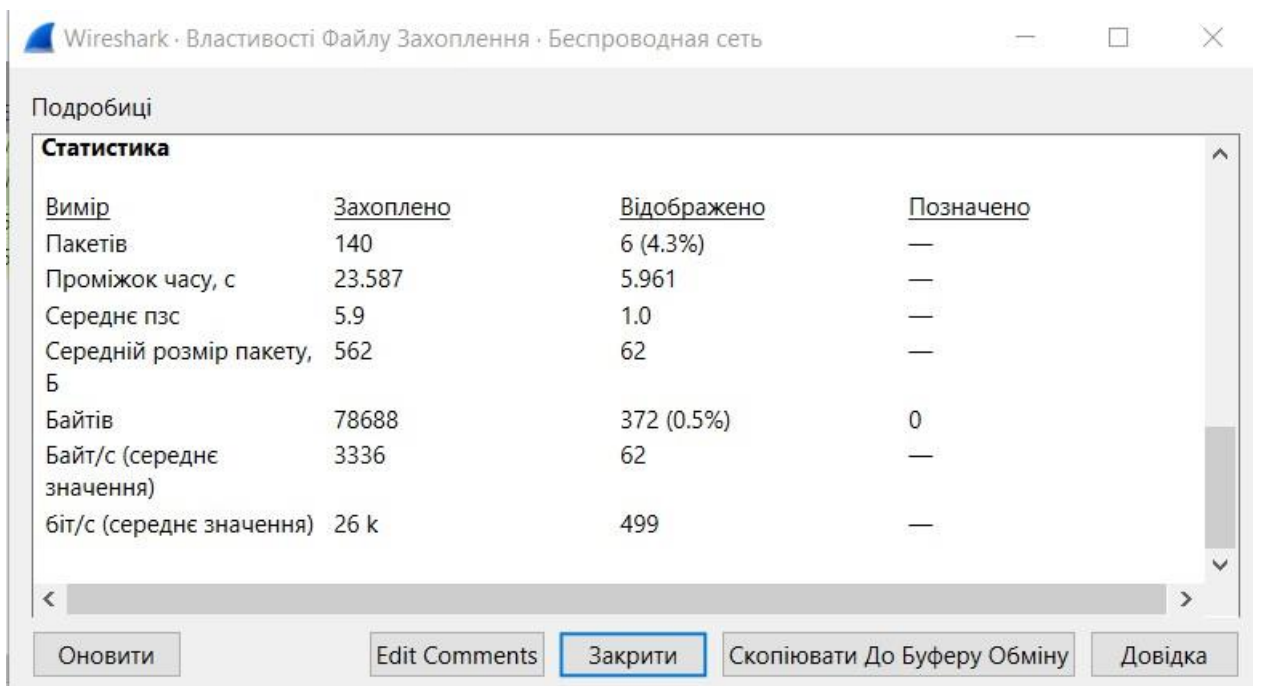
```
> Frame 29: 239 bytes on wire (1912 bits), 239 bytes captured (1912 bits) on interface \Device\NPF_{105064AA-4BD4-40E3-957F-1556F...}
  Ethernet II, Src: Intel_4d:ce:34 (f0:9e:4a:4d:ce:34), Dst: TPLink_7c:a0:f4 (00:31:92:7c:a0:f4)
  Internet Protocol Version 4, Src: 192.168.0.106, Dst: 149.154.167.41
```

Використаємо фільтр *tcp.flags.syn == 1*:



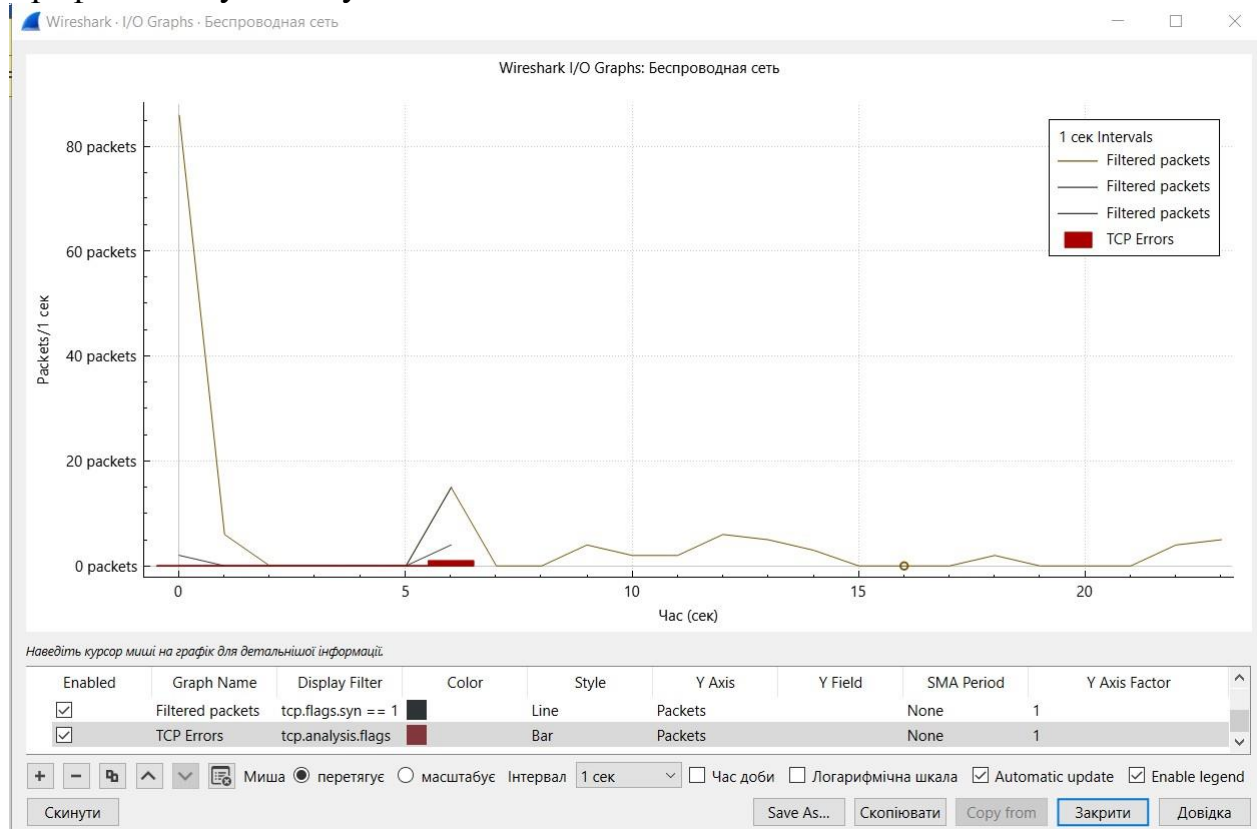
Як бачимо, значення максимального розміру сегмента (MSS) дорівнює 1460. Значення $MSS = 1460$ байт є стандартним для більшості мереж, оскільки воно базується на стандартному розмірі MTU (Maximum Transmission Unit) для Ethernet. Якщо MSS дорівнює 1460, це означає, що TCP-сегмент може передавати до 1460 байтів корисних даних в одному пакеті, не фрагментуючи його. MSS використовується для оптимізації передачі даних. Якщо пакет перевищує розмір MTU, він розбивається на частини (фрагментується), що може призвести до затримок і неефективності.

Скріншоти даних з меню Статистика, де вказано інформацію про наш файл захоплення:



Пакетів – 140; Проміжок часу – 23.5875с; Середній розмір пакету – 562 байт; Байтів ~ 78688;
Середня швидкість передачі даних – 3336 біт/с.

Графіки вводу-виводу:



На основі графіка з інтерфейсу Wireshark, пік швидкості передачі даних (в кількості пакетів) відбувається на початку сесії, приблизно в перші секунди. У цей момент спостерігається максимальне значення графіка — близько 80 пакетів на секунду.

Щоб точніше визначити пікову швидкість передачі даних, потрібно звернути увагу на найвищу точку кривої, яка є на початку графіка, коли передається максимальна кількість пакетів за одиницю часу.

Таблиця Ethernet Conversations, де вказано MAC-адреси пристроїв, між якими відбувався обмін даними. Тут ми бачимо власну адресу, яка фігурувала в завданнях вище, та адресу домашнього роутера:

Wireshark - Conversations - Беспроводная сеть

Conversation Settings

☐ Визначення імен

☐ Absolute start time

☒ Limit to display filter

Скопіювати

Follow Stream...

Graph...

Протокол

☒ UDP

☐ USB

☐ ZigBee

Filter list for specific type

Ethernet · 1

IPv4 · 2

IPv6

TCP · 3

UDP

Address A	Address B	Пакетів	Bytes	Stream ID	Total Packets	Percent Filtered	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration
00:31:92:7c:a0:14	10:9e:4a:4d:ce:34	6	372 байти	0	140	4.29%	3	186 байти	3	186 байти	0.000000	23.5875

Закрити

Довідка

Висновки:

Під час проведення цієї лабораторної роботи ми дослідили процес передачі даних у стеку TCP/IP. Застосовуючи фільтри на основі MAC- та IP-адрес, було розглянуто теоретичний матеріал і виконано ретельний аналіз захопленого файлу. За допомогою розділу "Статистика" визначено кількість прийнятих пакетів, зафіксовано час їх отримання, а також розраховано середню і максимальну швидкість передачі даних. Крім того, було виявлено та проаналізовано кінцеві точки з'єднання.