

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»**

Виконав студент групи: KB-11

ПІБ: Терентьев Іван Дмитрович

Перевірив: _____

Київ 2024

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

Теоретичні відомості

Передача даних у стеку протоколів **TCP/IP** відбувається через кілька рівнів, кожен з яких додає власну службову інформацію у вигляді заголовків. Основна ідея полягає в тому, що дані проходять зверху вниз через рівні моделі OSI, починаючи з прикладного рівня і завершуючи фізичним.

Прикладний рівень:

Прикладний рівень відповідає за сервіси, які використовуються безпосередньо користувачами або додатками. Одним з таких сервісів є **FTP (File Transfer Protocol)**, який широко застосовується для передачі файлів між віддаленими системами. FTP працює за допомогою двох каналів:

1. **Управлінський канал**, що використовує TCP на порті 21 для обміну командами між клієнтом і сервером.
2. **Дані** передаються через окремий канал, який може бути відкритий або сервером (в активному режимі), або клієнтом (у пасивному режимі).

FTP використовує **TCP**, оскільки це надійний транспортний протокол, що гарантує правильну доставку даних без втрат і в правильному порядку.

Транспортний рівень:

Транспортний рівень реалізує функції передачі даних між двома хостами і використовує два основні протоколи: **TCP** і **UDP**.

1. TCP (Transmission Control Protocol):

- TCP є протоколом з'єднання, який гарантує надійну передачу даних. Кожне з'єднання починається з процедури "триетапного рукошлякування" (SYN, SYN-ACK, ACK), під час якого встановлюються всі необхідні параметри для передачі даних.
- **Сегментація** даних: повідомлення поділяється на сегменти, кожен з яких має заголовок, що містить порядковий номер (Sequence Number). Це дозволяє одержувачу відновити правильну послідовність даних навіть у разі їх отримання в іншому порядку.
- TCP також підтримує контроль помилок, використовуючи **контрольну суму (Checksum)**, яка перевіряє цілісність даних. Якщо виявлено помилку, сегмент повторно відправляється.
- **Флаги управління** (SYN, ACK, FIN) використовуються для керування з'єднанням: встановлення, підтримки і завершення сесії.

2. UDP (User Datagram Protocol):

- UDP є ненадійним і простішим у порівнянні з TCP, оскільки не гарантує доставку даних або їх правильний порядок. Він використовується в ситуаціях, де важлива швидкість передачі, а не надійність, наприклад, для потокового відео чи голосових викликів.

Для протоколу FTP використовується лише TCP, оскільки він забезпечує необхідну надійність при передачі файлів.

Мережевий рівень:

Мережевий рівень відповідає за маршрутизацію і доставку пакетів (дейтаграм) між хостами в мережі на основі IP-адрес. Основним протоколом на цьому рівні є **IP (Internet Protocol)**.

- **IP** забезпечує передавання даних між хостами, додаючи заголовок, який містить IP-адресу відправника і отримувача. Протокол IP не гарантує надійність передачі, тому він є ненадійним протоколом без встановлення з'єднання.
- **Фрагментація**: якщо розмір даних перевищує максимальний розмір кадру канального рівня (MTU), IP фрагментує їх на менші частини, які потім збираються на стороні отримувача.

У процесі роботи FTP IP передає дані між клієнтом і сервером через мережу, використовуючи IP-адреси для ідентифікації відправника і отримувача.

Псевдозаголовки та контрольна сума:

Для перевірки коректності переданих даних використовуються контрольні суми, які обчислюються не лише на основі даних заголовка TCP або UDP, але й додаткових даних, таких як IP-адреси відправника і отримувача. Ці дані входять до складу **псевдозаголовку**, що дозволяє гарантувати, що дані будуть доставлені вірному отримувачу.

Аналізатор трафіку Wireshark:

Wireshark є потужним інструментом для аналізу мережевого трафіку. Він дозволяє захоплювати пакети, що передаються через мережу, і аналізувати їх. Під час роботи з FTP за допомогою Wireshark можна досліджувати TCP-сесії, що використовуються для передачі файлів:

- Встановлення з'єднання починається з обміну пакетами SYN, SYN-ACK і ACK, що ініціює сеанс.
- Дані передаються сегментами, кожен з яких містить **Source Port, Destination Port, Sequence Number, Acknowledgment Number, Flags (SYN, ACK, FIN)**.
- Закриття з'єднання відбувається шляхом відправки пакета з прапорцем **FIN**, який сигналізує про завершення передачі.

Наприклад, під час сеансу FTP можна використовувати Wireshark для захоплення трафіку між клієнтом і FTP-сервером. Завдяки фільтрам можна сфокусуватися на конкретних IP-адресах або портах. Це дозволить побачити кожен етап передачі: від запиту на з'єднання до закриття сесії після завершення завантаження файлу.

Формати заголовків TCP та UDP:

- **TCP-заголовок** містить інформацію про порти, порядкові номери, контрольні суми і флаги керування, такі як SYN, ACK і FIN.
- **UDP-заголовок** є значно простішим і містить тільки порти, довжину дейтаграми і контрольну суму.

Встановлення та завершення TCP-з'єднання:

Процес встановлення TCP-з'єднання відбувається за допомогою тристороннього рукостискання (SYN, SYN-ACK, ACK). Після цього відбувається передача даних, яка завершується послідовністю обміну пакетами з прапорцем **FIN** для завершення сеансу.

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.

1. Активізувати режим захоплення даних з використанням програми Wireshark.
2. Завантажити файл довідки README.TXT.

2.1. Під'єднатися до FTP-сервера центру FreeBSD:

`ftp://ftp3.ie.freebsd.org.`

2.2. В розділі pub/FreeBSD знайти і завантажити файл README.TXT (рисунок 2.10).

2.3. Після завершення завантаження файлу зупинити захоплення даних програмою Wireshark.

3. Відкрити головне вікно програми Wireshark.
4. Проаналізувати поля заголовків сегментів TCP.

№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 сформулювати фільтр за MAC-адресою.

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

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?

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 зміст її рядків;
- вивести IO Graphs, за допомогою якого визначити пікову швидкість передачі даних протягом інтервалу, що підлягає аналізу.

5. За результатами роботи зробити висновки.

Хід роботи

The screenshot shows the Wireshark interface with a packet capture of TCP traffic. The packet list at the top shows several packets, including a SYN packet (No. 86) and a SYN-ACK packet (No. 87). The details pane on the right shows the expanded view of the selected packet (No. 86), displaying the TCP header fields: Source (192.168.88.219), Destination (85.30.190.138), Protocol (TCP), Length (66), and Info (Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM). The packet bytes pane on the right shows the raw data of the packet, including the Ethernet II header and the TCP header.

No.	Time	Source	Destination	Protocol	Length	Info
86	6.675366	192.168.88.219	85.30.190.138	TCP	66	53359 → 21 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
87	6.711773	85.30.190.138	192.168.88.219	TCP	66	21 → 53359 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 WS=64 SACK_PERM
127	7.084363	192.168.88.219	85.30.190.138	TCP	66	53360 → 56604 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
128	7.120703	85.30.190.138	192.168.88.219	TCP	66	56604 → 53360 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 WS=64 SACK_PERM
161	7.375390	192.168.88.219	85.30.190.138	TCP	66	53361 → 52615 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
162	7.414195	85.30.190.138	192.168.88.219	TCP	66	52615 → 53361 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 WS=64 SACK_PERM
211	9.986872	192.168.88.219	85.30.190.138	TCP	66	53362 → 50391 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
213	10.024373	85.30.190.138	192.168.88.219	TCP	66	50391 → 53362 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 WS=64 SACK_PERM

Transmission Control Protocol (tcp), 32 байта

Пакеты: 303 · Отображено: 8 (2.6%) · Потеряно: 0 (0.0%)

Профиль: Default

Рис.1 – Розгорнутий фрагмент TCP, встановлення зв'язку [SYN], [SYN, ACK], [SYN]

На рис. 1 можна побачити розгорнутий фрагмент TCP після завантаження файлу з FTP серверу, пакети були відфільтровані за протоколом TCP, IP адресою серверу та за прапором SYN.

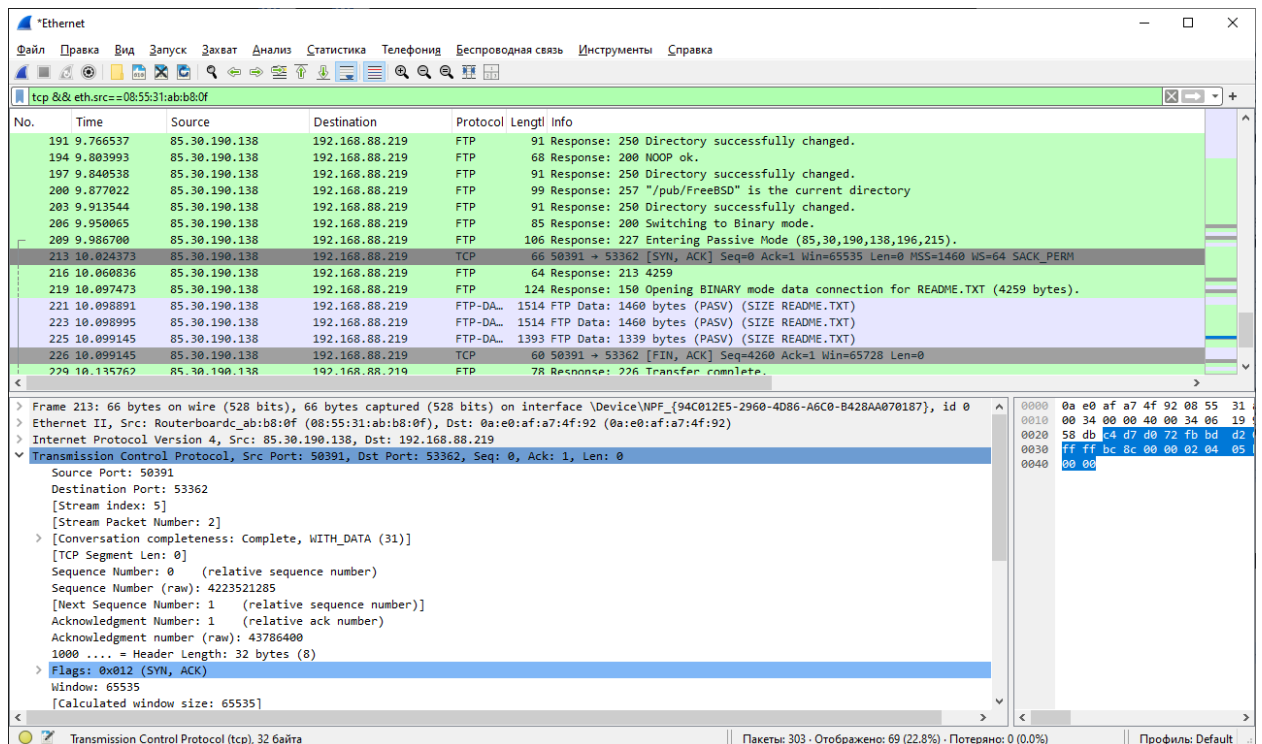


Рис.2 – Розгорнутий фрагмент TCP, пакети відфільтровані за MAC-адресою

На рис. 2 можна побачити приклад фільтрації пакетів за MAC-адресою, а також побачити розмір MSS, що становить 1460.

Фрагмент TCP містить наступні поля:

Порт відправника: 50391 (порт виданий NAT);

Порт одержувача: 53362 (порт виданий NAT);

Порядковий номер: 2 (перший октет у сегменті TCP);

Номер підтвердження: 1 (також перший октет у сегменті TCP);

Прапорці: встановлено прапорець SYN (синхронізація), ACK;

Розмір вікна: 65535 байт;

Максимальний розмір сегменту (MSS): 1460 байт (MTU – 1484 байт, IP та TCP заголовки займають по 20 байт, додаткових опцій немає, тому $MSS = 1500 - 20 - 20 = 1460$).

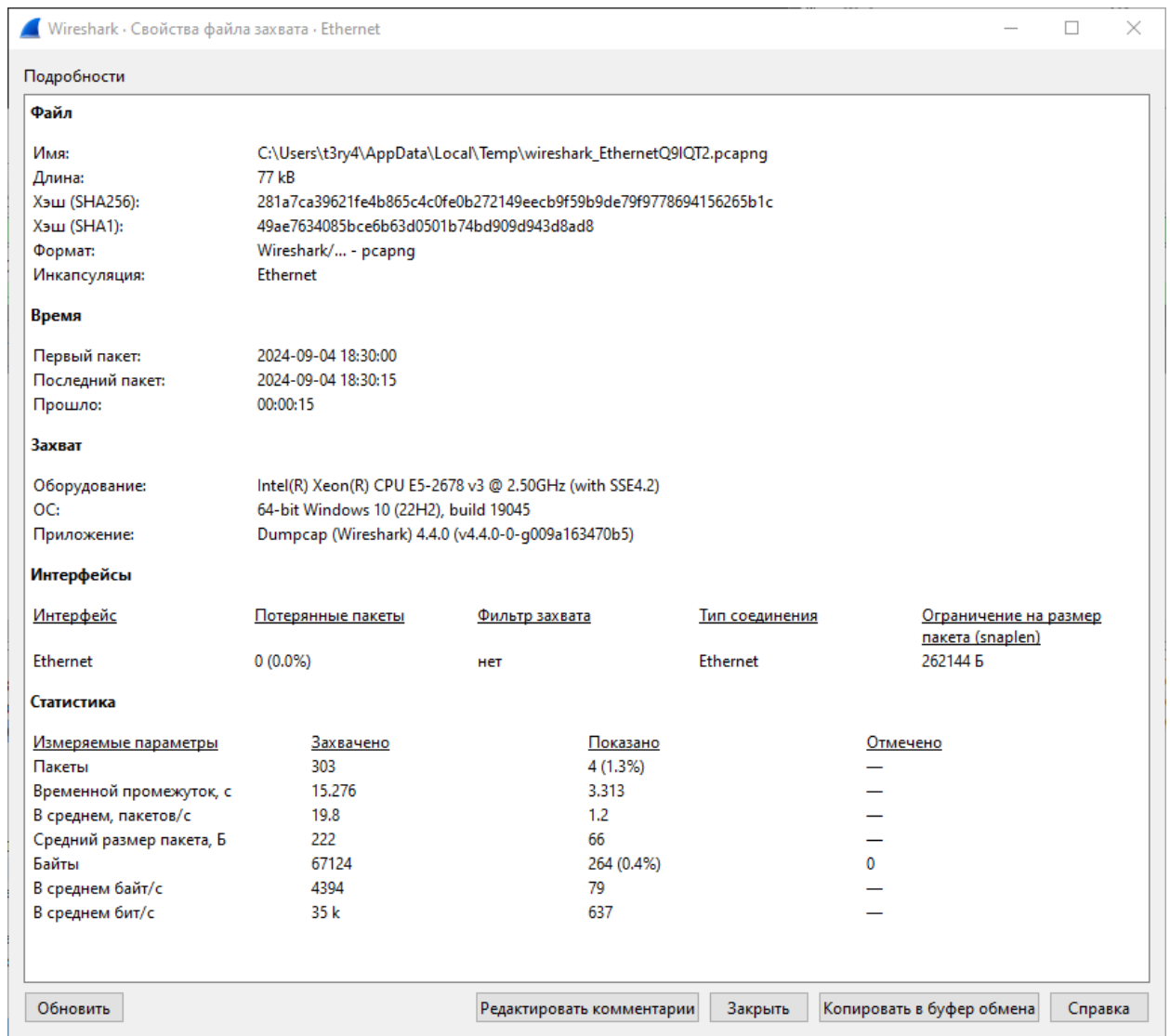


Рис. 3 – Статистика захоплення пакетів Wireshark

З рис. 3 можна отримати основну статистику.

Кількість захоплених пакетів та байтів: 303 пакети та 67124 байти

Середня швидкість передачі даних(біти/секунда): 19.8 бітів за секунду

Середній розмір пакету: 222 байти

Час, протягом якого здійснювалось захоплення трафіку: 15.276 секунд

Wireshark - Conversations - Ethernet

Conversation Settings

☐ Разрешение имен

☐ Абсолютное время запуска

☒ Ограничить по фильтру: et

Копировать

Отслеживать поток...

График...

Протокол

☐ NCP

☐ openssl

☐ other

Фильтровать список по указанн...

Эthernet 1

Адрес А	Адрес В	Пакеты	Байт	ID потока	Всего пакетов	Оформировано в процентах	Packets A -> B	Bytes A -> B	Packets B -> A	Bytes B -> A	Отн. время начала	Продолжительность	Bits/s A -> B	Bits/s B -> A
08:00:27:44:92	08:55:31:ab:b6:0f	4	264 байт	1	238	1.68%	0	0 байт	4	264 байт	0.053817	14.44с	0 bits/s	146 bits/s

Закреть

Справка

Рис. 4 – Таблица Ethernet Conversations

На рис. 4 зображено таблицю Ethernet Conversations. Кожен рядок у цьому списку містить дані про:

- MAC-адреси відправника і отримувача – унікальні апаратні адреси пристроїв, які беруть участь у передачі даних.
- Кількість пакетів – скільки пакетів було надіслано з одного пристрою на інший і назад (якщо є двосторонній зв'язок).
- Загальний обсяг даних – скільки байтів було передано між пристроями.
- Тривалість – час, протягом якого відбувалася передача даних між цими двома пристроями.

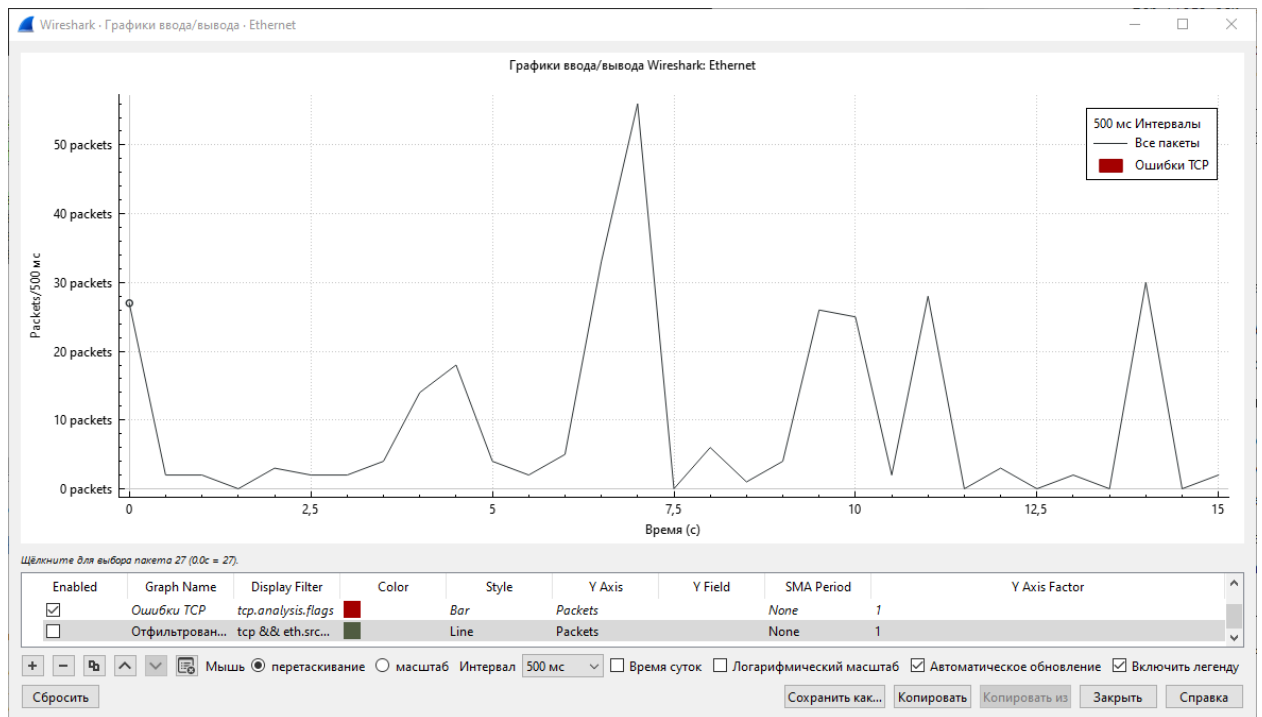


Рис. 5 – IO Graphs, графік вводу/виводу

На рис. 5 зображений графік зміни обсягу переданих даних(пакетів) у часі, можна побачити, що на 7-ій секунді було пікове навантаження, а саме 56 пакетів.

Висновок

Під час лабораторної роботи було проведено дослідження передачі даних за стеком протоколів TCP/IP за допомогою Wireshark. Дослідження було зосереджено на ролях транспортного та мережевого рівнів, проаналізовано заголовки протоколів TCP і UDP і проведено фільтрацію даних на основі MAC-адрес.

Аналіз також включав сегменти TCP, зокрема ті, що містять прапори SYN та ACK, під час визначення параметрів з'єднання MSS. Крім того, дослідження таблиці розмов Ethernet і графіків вводу/виводу показало пікові навантаження, що виникають під час передачі даних.

Загалом покращили розуміння принципів мережевого трафіку, а також удосконалили вміння використовувати Wireshark для аналізу мережевих з'єднань.