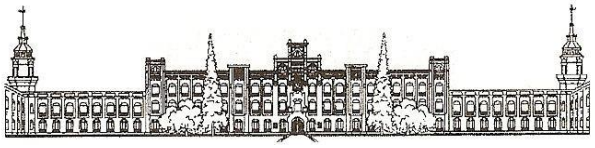

Міністерство освіти і науки України
Національний технічний університет України
«Київський політехнічний інститут»

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: КВ-11

ПІБ: Терентьєв 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 рукостискання" (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 рекомендачій викладача сформувати фільтр за MAC-адресою.

2. Розглянути результат інкапсуляції при передачі даних. В захоплених пакетах виділити службову інформацію (заголовки) всіх блоків даних, а також, за наявністю, 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 вміст її рядків;
- вивести IO Grafs, за допомогою якого визначити пікову швидкість передачі даних протягом інтервалу, що підлягає аналізу.

5. За результатами роботи зробити висновки.

Хід роботи

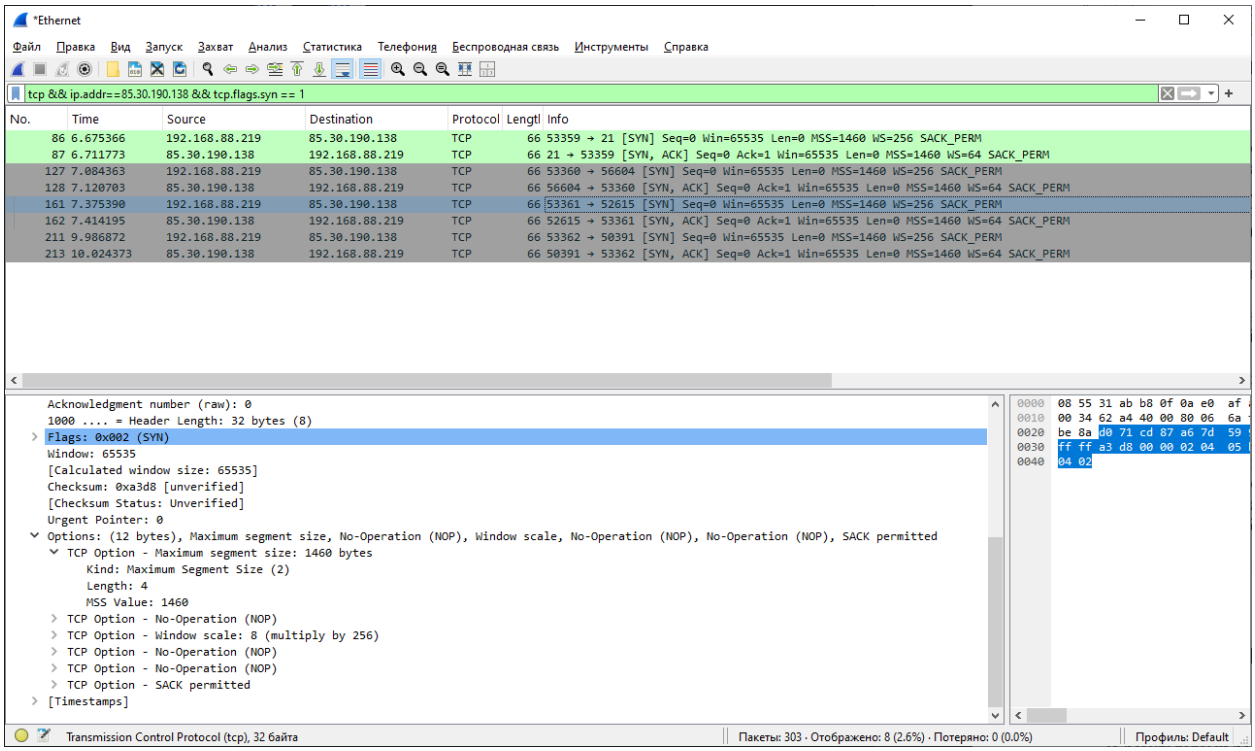

Рис.1 – Розгорнутий фрагмент TCP, встановлення зв’язку [SYN], [SYN, ACK], [SYN]

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.

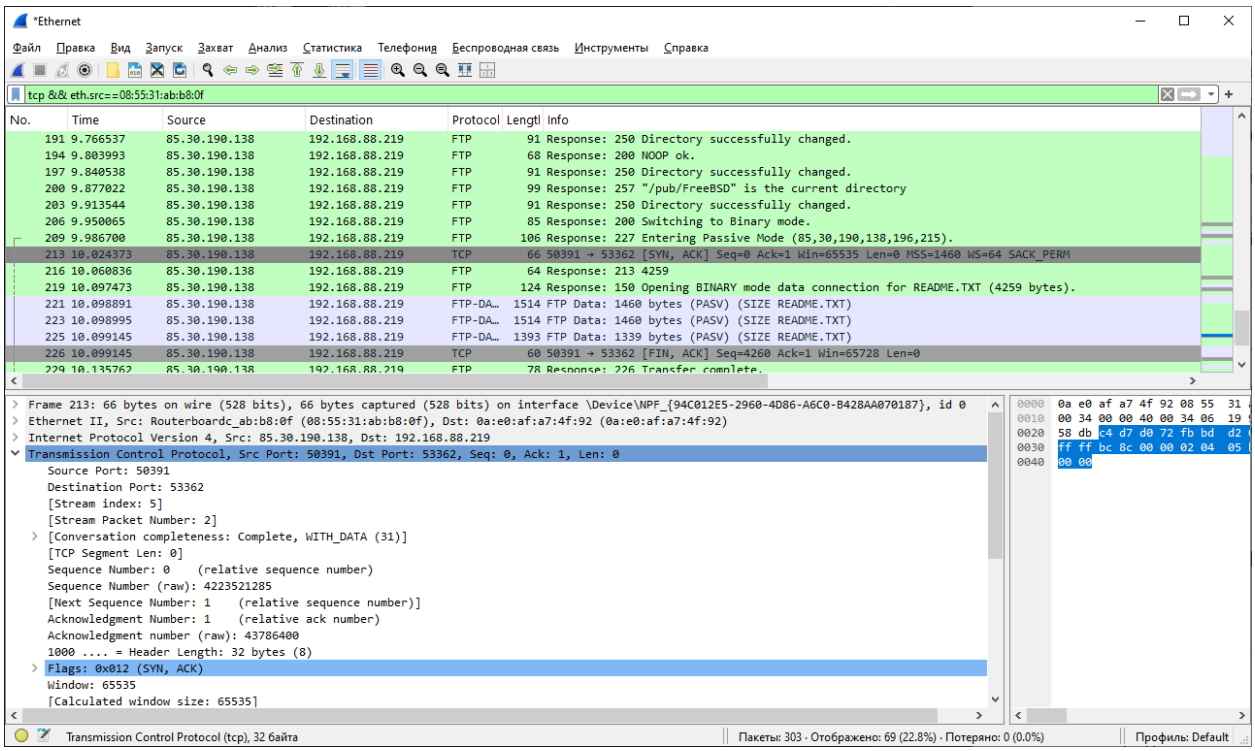

Рис.2 – Розгорнутий фрагмент TCP, пакети відфільтровані за MAC-адресою

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 NAT: 33449;

Порт одержувача: 80 (стандартний для FTP-з’єднань);

Порядковий номер: 0 (перший октет у сегменті TCP);

Номер підтвердження: 0 (також перший октет у сегменті TCP);

Прапорці: встановлено прапорець SYN (синхронізація);

Розмір вікна: 62240 байт;

Максимальний розмір сегменту (MSS): 1460 байт (MTU – 1484 байт, IP та TCP заголовки займають 12 байт, додаткових опцій немає, тому MSS = 1484 - 12 - 12 = 1460).

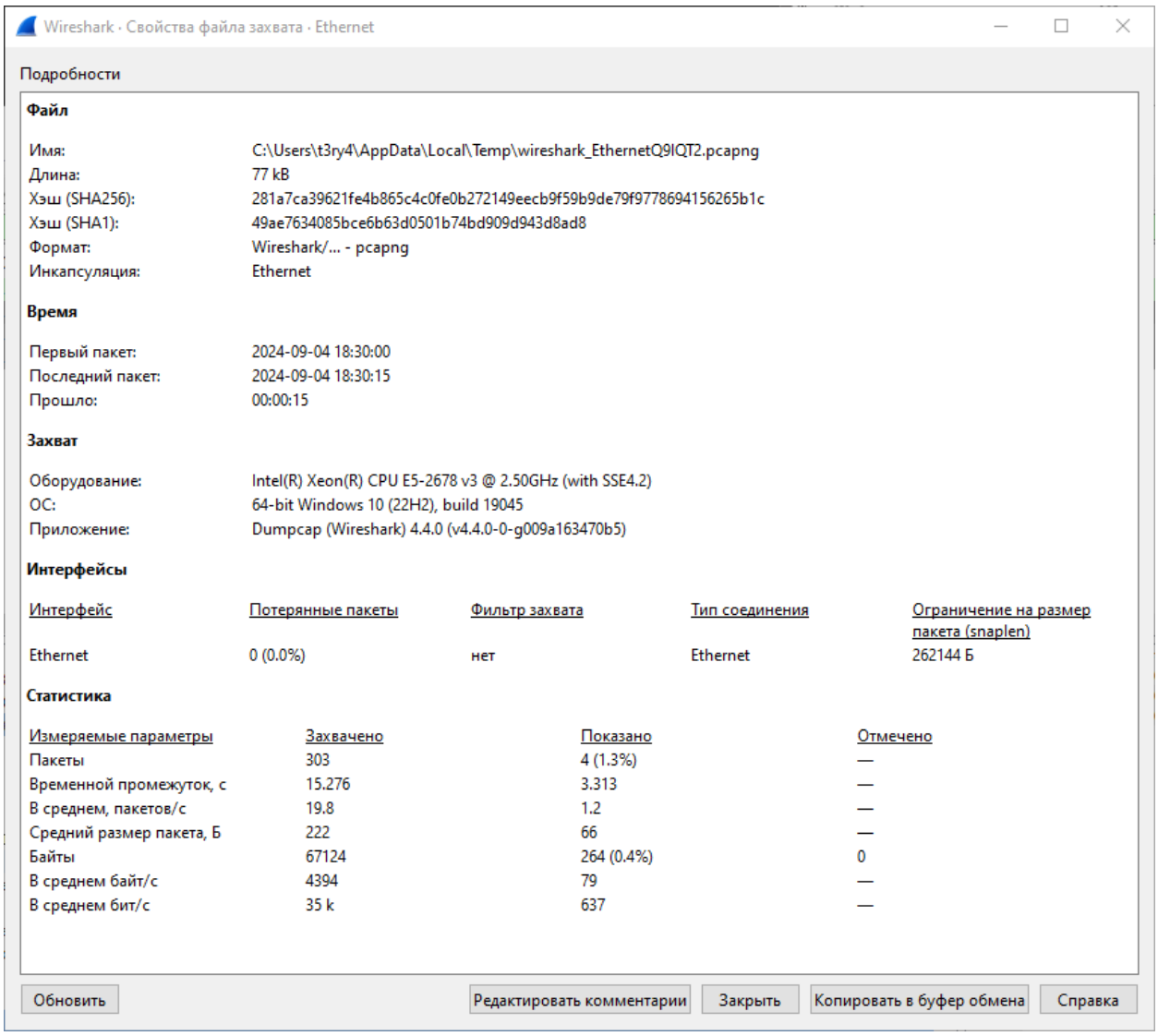

Рис. 3 – Статистика захоплення пакетів Wireshark

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

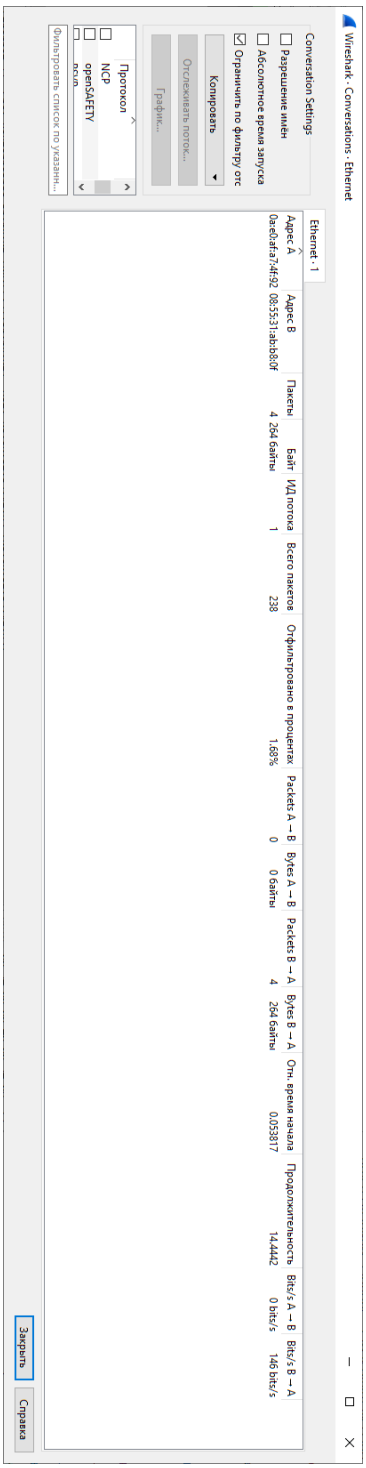

Рис. 4 – Таблиця Ethernet Conversations

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.

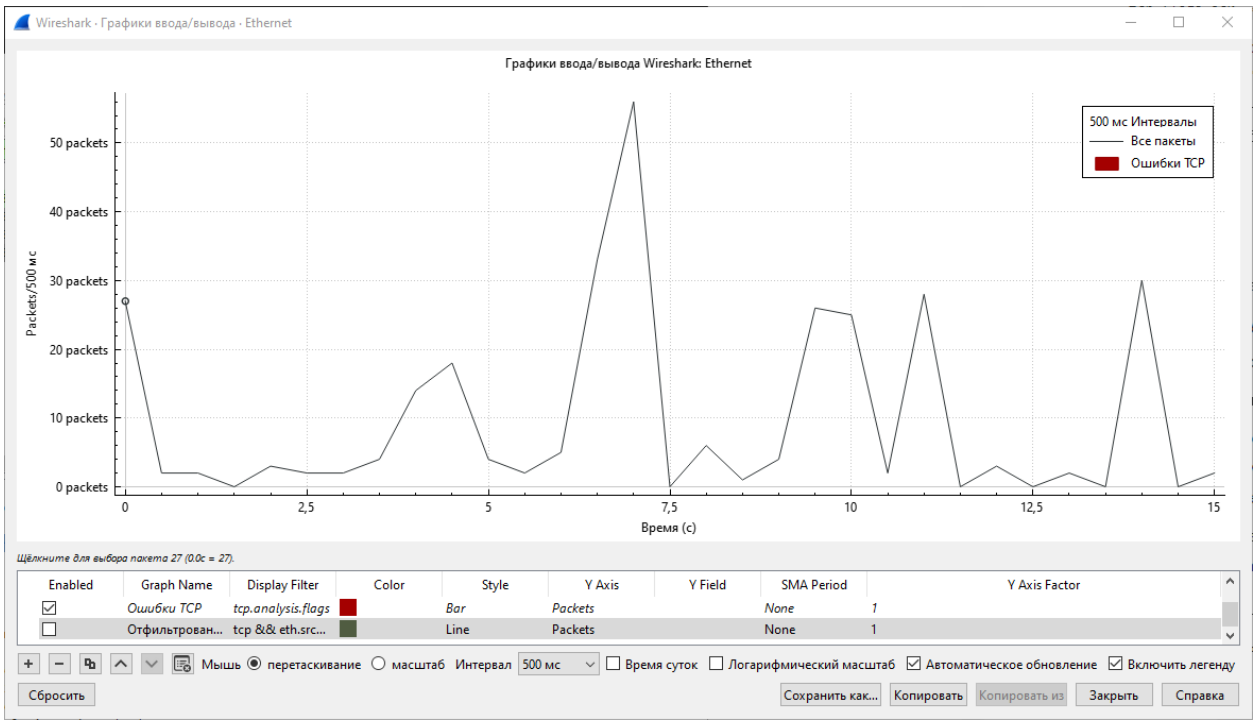

Рис. 5 – IO Graphs, графік вводу/виводу

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.