



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

Лабораторна робота №1
з дисципліни «Комп'ютерні мережі»

**«Стеки мережевих протоколів.
Аналізатор мережевого трафіку Wireshark»**

Виконав студент групи: КВ-11

ПІБ: Терентьєв Іван Дмитрович

Перевірив: _____

Київ 2024

Мета роботи

Засвоєння функцій модулів різних рівнів еталонної моделі OSI, процедури інкапсуляції та формування повідомлень для передачі в мережу; ознайомлення та вивчення аналізатора мережевого трафіку Wireshark.

План виконання лабораторної роботи

1. Ознайомитися та засвоїти теоретичні відомості про еталонну модель взаємодії відкритих систем OSI та стек мережевих протоколів TCP/IP.
2. Ознайомитися з можливостями аналізатора мережевого трафіку Wireshark.
3. За допомогою аналізатора Wireshark виконати захоплення та провести аналіз мережевих пакетів.

Завдання

1. При виконанні роботи використовується програмне забезпечення для аналізу протоколів комп'ютерних мереж Wireshark. Запустити відповідну програму.
2. Вибрати інтерфейс для захоплення трафіку (меню Capture/Interface) та активізувати режим захоплення.
3. Скопіювати через мережу файл розміром кілька десятків Мбайт.
4. Завершити захоплення трафіку та перейти до режиму аналізу. В захопленому фрагменті виберіть кадр, який містить пакет TCP. Виділіть складові частини кадру. Знайдіть в кадрі транспортні, логічні та фізичні адреси відправника та отримувача.

Теоретичні відомості

Модель OSI

Модель OSI (Open System Interconnect) є еталонною моделлю взаємодії комп'ютерних систем через мережу. Вона розділяє процес передачі даних на 7 рівнів, кожен з яких виконує специфічні функції: від взаємодії з користувачем (прикладний рівень) до передачі сигналів (фізичний рівень). Кожен рівень взаємодіє лише з сусідніми рівнями, забезпечуючи абстрагування складних процесів обміну даними.

TCP/IP

На відміну від OSI, модель TCP/IP (Department of Defense) складається з 4 рівнів і більше орієнтована на реальні потреби передачі даних. Вона використовується у мережах Інтернету і охоплює транспортний рівень, який забезпечує надійність передачі даних, та мережевий рівень, який відповідає за маршрутизацію.

Інкапсуляція

Процедура інкапсуляції — це процес додавання службової інформації до даних кожного рівня під час їхнього переміщення від верхніх рівнів до фізичного. Вона дозволяє мережевим пристроям коректно передавати і приймати дані.

Аналізатор Wireshark

Wireshark — це популярний інструмент для перехоплення та аналізу мережевого трафіку. Він дозволяє досліджувати пакети на різних рівнях, фільтрувати їх за протоколами або умовами, що дозволяє знаходити проблеми в мережі або аналізувати шкідливий трафік.

Хід роботи

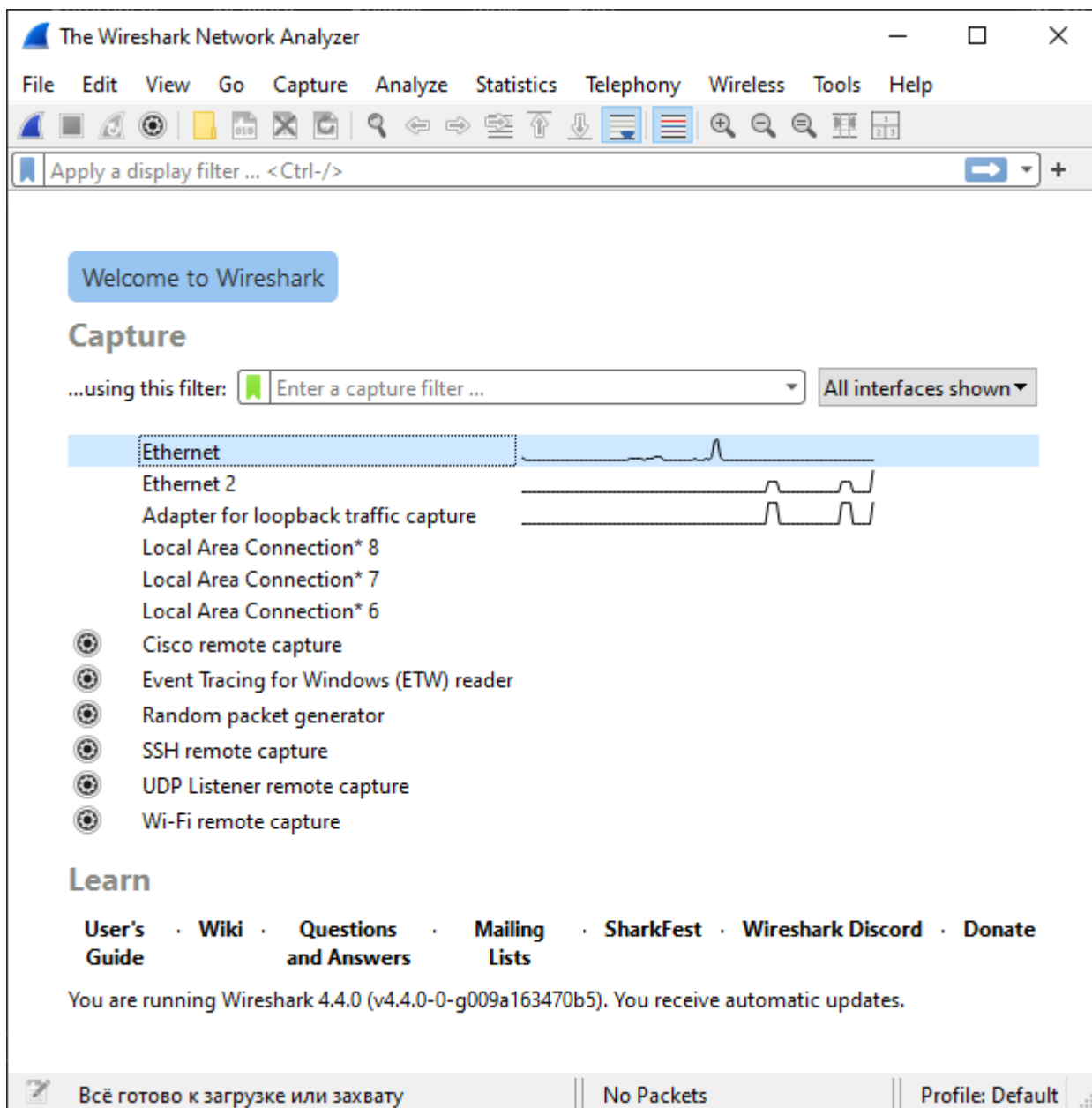


Рис. 1 – Головне вікно програми Wireshark

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.88.219	104.81.96.166	TCP	54	57227 → 443 [ACK] Seq=1 Ack=1 Win=510 Len=0
2	0.241425	142.250.203.138	192.168.88.219	UDP	119	443 → 51328 Len=77
3	0.263453	192.168.88.219	142.250.203.138	UDP	75	51328 → 443 Len=33
4	0.451482	192.168.88.219	142.250.203.138	UDP	71	51328 → 443 Len=29
5	0.492866	142.250.203.138	192.168.88.219	UDP	67	443 → 51328 Len=25
6	0.575098	Routerboardc_ab:b8:...	Spanning-tree-(for...	STP	60	RST. Root = 32768/0/08:55:31:ab:b8:0f Cost = 0 Port = 0x8003
7	0.621942	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0x433018b7
8	0.632352	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0x359f2db9
9	0.698184	192.168.88.219	142.250.203.138	UDP	71	51328 → 443 Len=29
10	0.739610	142.250.203.138	192.168.88.219	UDP	67	443 → 51328 Len=25
11	0.945346	192.168.88.219	142.250.203.138	UDP	71	51328 → 443 Len=29
12	0.986238	142.250.203.138	192.168.88.219	UDP	67	443 → 51328 Len=25
13	1.197571	192.168.88.219	142.250.203.138	UDP	71	51328 → 443 Len=29
14	1.238657	142.250.203.138	192.168.88.219	UDP	67	443 → 51328 Len=25
15	1.450519	192.168.88.219	142.250.203.138	UDP	71	51328 → 443 Len=29
16	1.491768	142.250.203.138	192.168.88.219	UDP	67	443 → 51328 Len=25
17	1.623673	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0xcb14351d
18	1.634054	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0xc91d7a79
19	1.699090	192.168.88.219	142.250.203.138	UDP	71	51328 → 443 Len=29
20	1.740278	142.250.203.138	192.168.88.219	UDP	67	443 → 51328 Len=25
21	2.144722	192.168.88.219	142.250.203.138	UDP	71	51328 → 443 Len=29
22	2.185654	142.250.203.138	192.168.88.219	UDP	67	443 → 51328 Len=25

> Frame 1: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface \Device...
 > Ethernet II, Src: 0a:e0:af:a7:4f:92 (0a:e0:af:a7:4f:92), Dst: Routerboardc_ab:b8:0f (0...
 > Internet Protocol Version 4, Src: 192.168.88.219, Dst: 104.81.96.166
 > Transmission Control Protocol, Src Port: 57227, Dst Port: 443, Seq: 1, Ack: 1, Len: 0

Рис. 2 – Результат сканування інтерфейсу Ethernet

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.88.219	149.154.167.99	TCP	66	57648 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
2	0.042875	149.154.167.99	192.168.88.219	TCP	66	443 → 57648 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM WS=1024
3	0.042928	192.168.88.219	149.154.167.99	TCP	54	57648 → 443 [ACK] Seq=1 Ack=1 Win=131328 Len=0
4	0.043293	192.168.88.219	149.154.167.99	TCP	1514	57648 → 443 [ACK] Seq=1 Ack=1 Win=131328 Len=1460 [TCP PDU reassembled in 5]
5	0.043293	192.168.88.219	149.154.167.99	TLSv1.3	380	Client Hello (SNI=telegram.org)
6	0.086270	149.154.167.99	192.168.88.219	TCP	60	443 → 57648 [ACK] Seq=1 Ack=1787 Win=33792 Len=0
7	0.088544	149.154.167.99	192.168.88.219	TLSv1.3	1294	Server Hello, Change Cipher Spec, Application Data
8	0.088633	149.154.167.99	192.168.88.219	TCP	1294	443 → 57648 [ACK] Seq=1241 Ack=1787 Win=33792 Len=1240 [TCP PDU reassembled in 13]
9	0.088649	192.168.88.219	149.154.167.99	TCP	54	57648 → 443 [ACK] Seq=1787 Ack=2481 Win=131328 Len=0
10	0.088738	149.154.167.99	192.168.88.219	TCP	1294	443 → 57648 [ACK] Seq=2481 Ack=1787 Win=33792 Len=1240 [TCP PDU reassembled in 13]
11	0.088775	149.154.167.99	192.168.88.219	TCP	430	443 → 57648 [PSH, ACK] Seq=3721 Ack=1787 Win=33792 Len=376 [TCP PDU reassembled in 13]
12	0.088786	192.168.88.219	149.154.167.99	TCP	54	57648 → 443 [ACK] Seq=1787 Ack=4097 Win=131328 Len=0
13	0.089848	149.154.167.99	192.168.88.219	TLSv1.3	1294	Application Data
14	0.089876	149.154.167.99	192.168.88.219	TLSv1.3	405	Application Data, Application Data
15	0.089885	192.168.88.219	149.154.167.99	TCP	54	57648 → 443 [ACK] Seq=1787 Ack=5688 Win=131328 Len=0
16	0.090108	192.168.88.219	149.154.167.99	TLSv1.3	134	Change Cipher Spec, Application Data
17	0.090231	192.168.88.219	149.154.167.99	TLSv1.3	146	Application Data
18	0.090365	192.168.88.219	149.154.167.99	TLSv1.3	646	Application Data
19	0.133021	149.154.167.99	192.168.88.219	TLSv1.3	341	Application Data
20	0.133071	149.154.167.99	192.168.88.219	TLSv1.3	341	Application Data
21	0.133093	192.168.88.219	149.154.167.99	TCP	54	57648 → 443 [ACK] Seq=2551 Ack=6262 Win=130816 Len=0
22	0.133202	149.154.167.99	192.168.88.219	TLSv1.3	125	Application Data
23	0.133277	192.168.88.219	149.154.167.99	TLSv1.3	85	Application Data
24	0.140881	149.154.167.99	192.168.88.219	TLSv1.3	318	Application Data
25	0.169205	192.168.88.219	149.154.167.99	TLSv1.3	141	Application Data
26	0.211822	149.154.167.99	192.168.88.219	TCP	60	443 → 57648 [ACK] Seq=6597 Ack=2669 Win=35840 Len=0
27	0.212158	149.154.167.99	192.168.88.219	TCP	1294	443 → 57648 [ACK] Seq=6597 Ack=2669 Win=35840 Len=1240 [TCP PDU reassembled in 46]
28	0.212264	149.154.167.99	192.168.88.219	TCP	1294	443 → 57648 [ACK] Seq=7837 Ack=2669 Win=35840 Len=1240 [TCP PDU reassembled in 46]
29	0.212275	192.168.88.219	149.154.167.99	TCP	54	57648 → 443 [ACK] Seq=2669 Ack=9077 Win=131328 Len=0
30	0.212369	149.154.167.99	192.168.88.219	TCP	1294	443 → 57648 [ACK] Seq=9077 Ack=2669 Win=35840 Len=1240 [TCP PDU reassembled in 46]
31	0.212475	149.154.167.99	192.168.88.219	TCP	1294	443 → 57648 [ACK] Seq=10317 Ack=2669 Win=35840 Len=1240 [TCP PDU reassembled in 46]
32	0.212482	192.168.88.219	149.154.167.99	TCP	54	57648 → 443 [ACK] Seq=2669 Ack=11557 Win=131328 Len=0
33	0.212580	149.154.167.99	192.168.88.219	TCP	1294	443 → 57648 [ACK] Seq=11557 Ack=2669 Win=35840 Len=1240 [TCP PDU reassembled in 46]
34	0.212686	149.154.167.99	192.168.88.219	TCP	1294	443 → 57648 [ACK] Seq=12797 Ack=2669 Win=35840 Len=1240 [TCP PDU reassembled in 46]
35	0.212693	192.168.88.219	149.154.167.99	TCP	54	57648 → 443 [ACK] Seq=2669 Ack=14037 Win=131328 Len=0
36	0.212791	149.154.167.99	192.168.88.219	TCP	1294	443 → 57648 [ACK] Seq=14037 Ack=2669 Win=35840 Len=1240 [TCP PDU reassembled in 46]
37	0.212896	149.154.167.99	192.168.88.219	TCP	1294	443 → 57648 [ACK] Seq=15277 Ack=2669 Win=35840 Len=1240 [TCP PDU reassembled in 46]
38	0.212909	192.168.88.219	149.154.167.99	TCP	54	57648 → 443 [ACK] Seq=2669 Ack=16517 Win=131328 Len=0
39	0.213002	149.154.167.99	192.168.88.219	TCP	1294	443 → 57648 [ACK] Seq=16517 Ack=2669 Win=35840 Len=1240 [TCP PDU reassembled in 46]
40	0.213109	149.154.167.99	192.168.88.219	TCP	1294	443 → 57648 [ACK] Seq=17757 Ack=2669 Win=35840 Len=1240 [TCP PDU reassembled in 46]
41	0.213122	192.168.88.219	149.154.167.99	TCP	54	57648 → 443 [ACK] Seq=2669 Ack=18997 Win=131328 Len=0
42	0.213215	149.154.167.99	192.168.88.219	TCP	1294	443 → 57648 [ACK] Seq=18997 Ack=2669 Win=35840 Len=1240 [TCP PDU reassembled in 46]

Рис. 3 – Результат сканування інтерфейсу Ethernet під час завантаження файлу

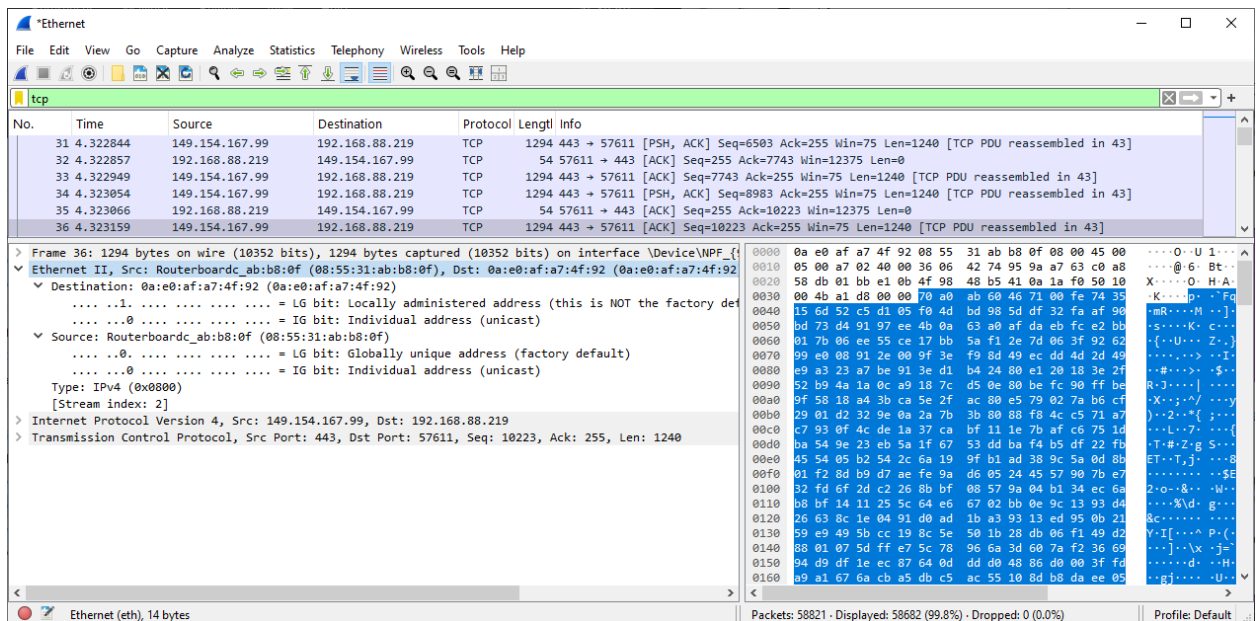


Рис. 4 – Ethernet заголовок пакета

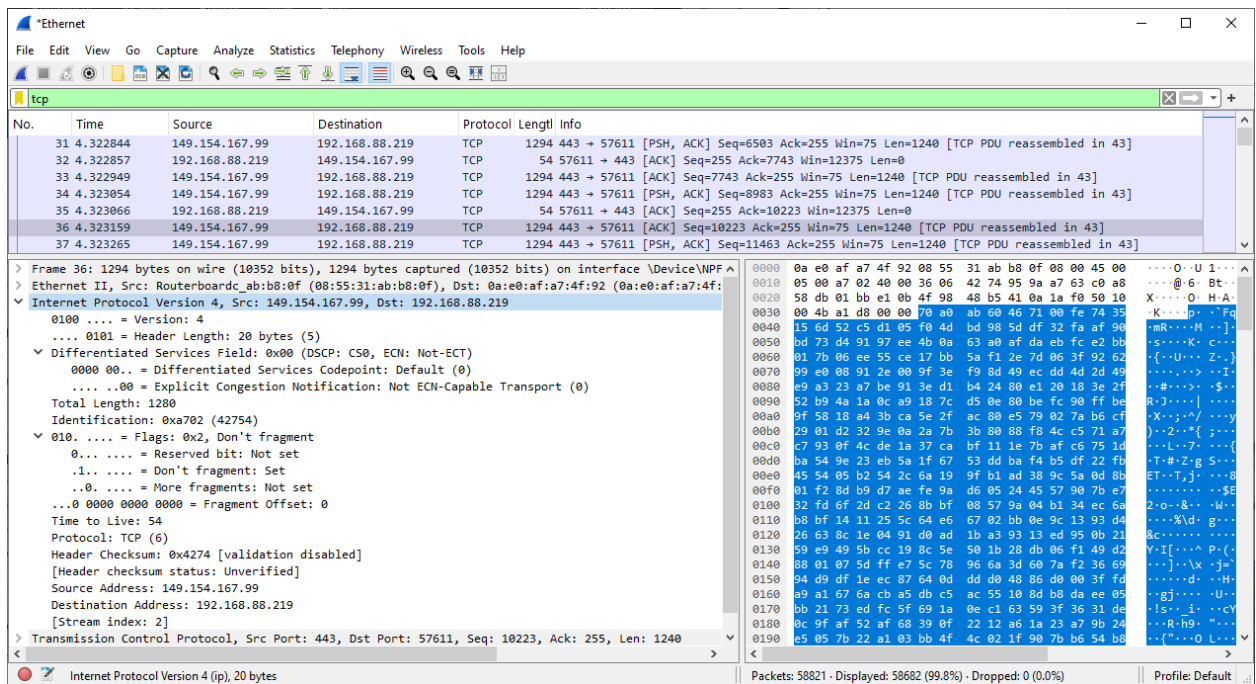


Рис. 5 – IP заголовок пакета

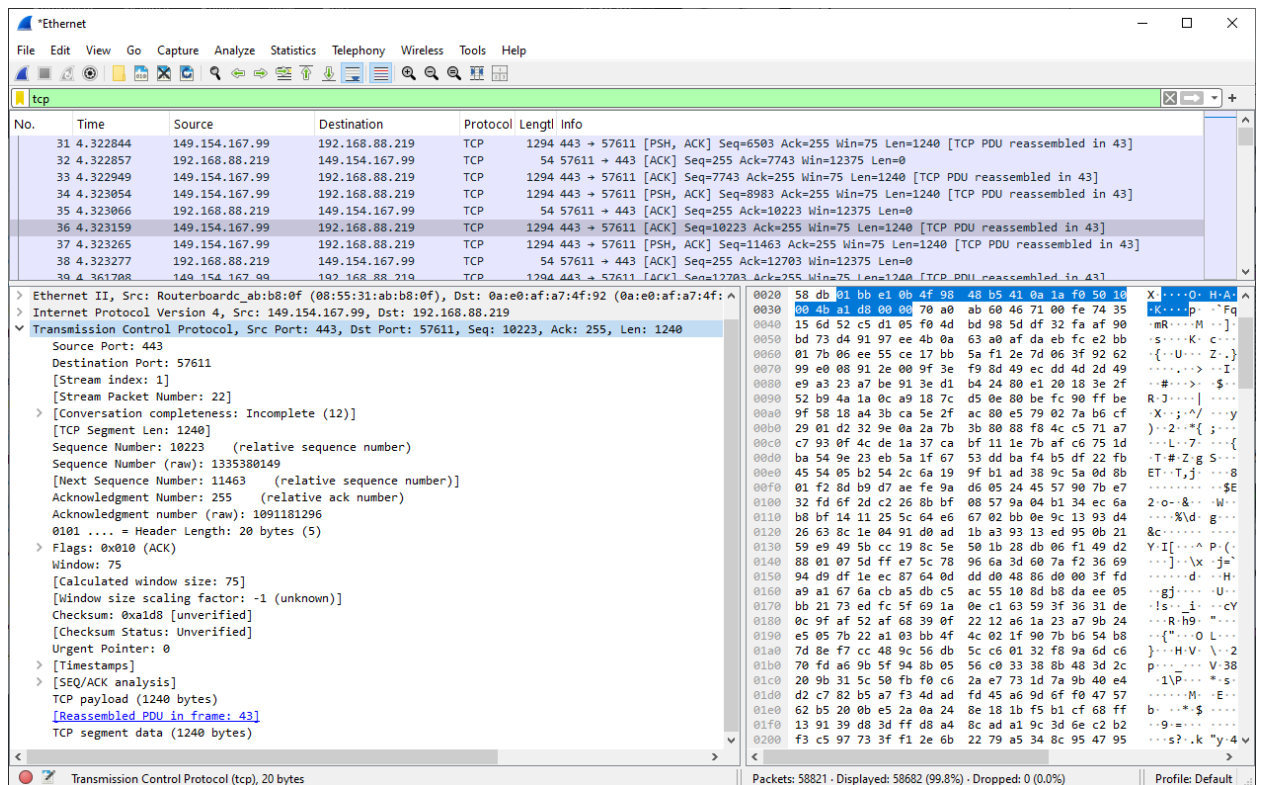


Рис. 6 – TCP заголовок пакету

На рис. 4, рис. 5, рис. 6 можна побачити транспортну, фізичну та логічну адреси відправника та отримувача, а саме:

- MAC адреса маршрутизатора: 08:55:31:ab:b8:0f
- MAC адреса локального пристрою: 0a:e0:af:a7:4f:92
- IP адреса серверу: 149.154.167.99
- IP адреса локального пристрою: 192.168.88.219(локальна адреса)
- Порт серверу: 443
- Порт отримувача(назначений NAT): 57611

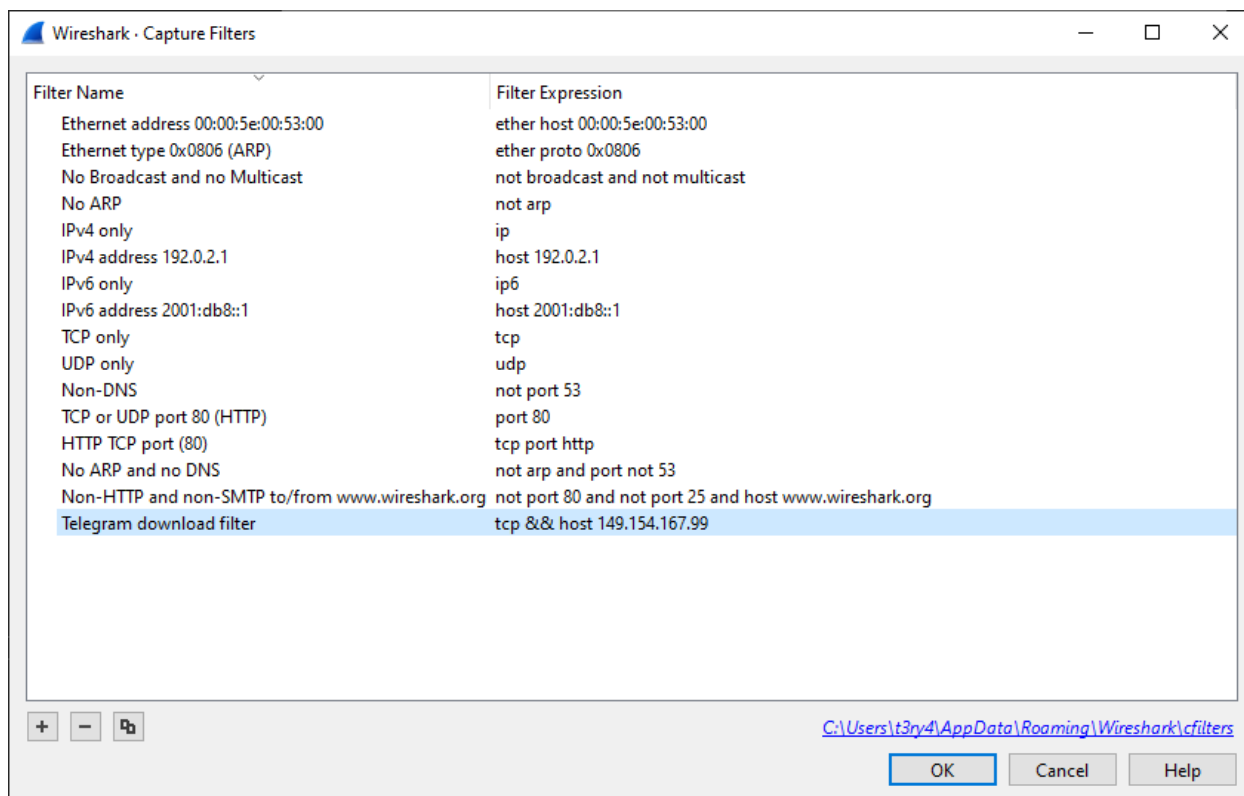


Рис. 7 – Створення фільтру захоплення пакетів TCP з IP адреси 149.154.167.99

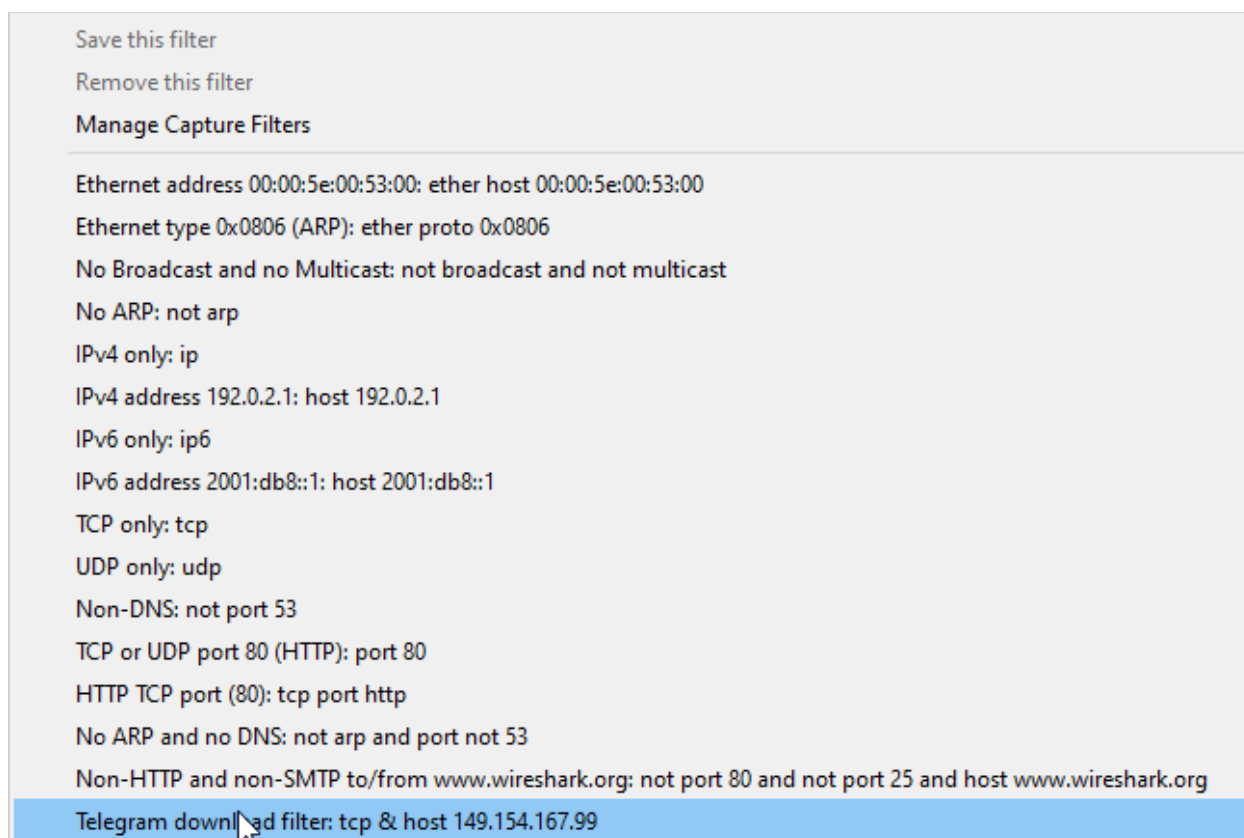


Рис. 8 – Застосування фільтру

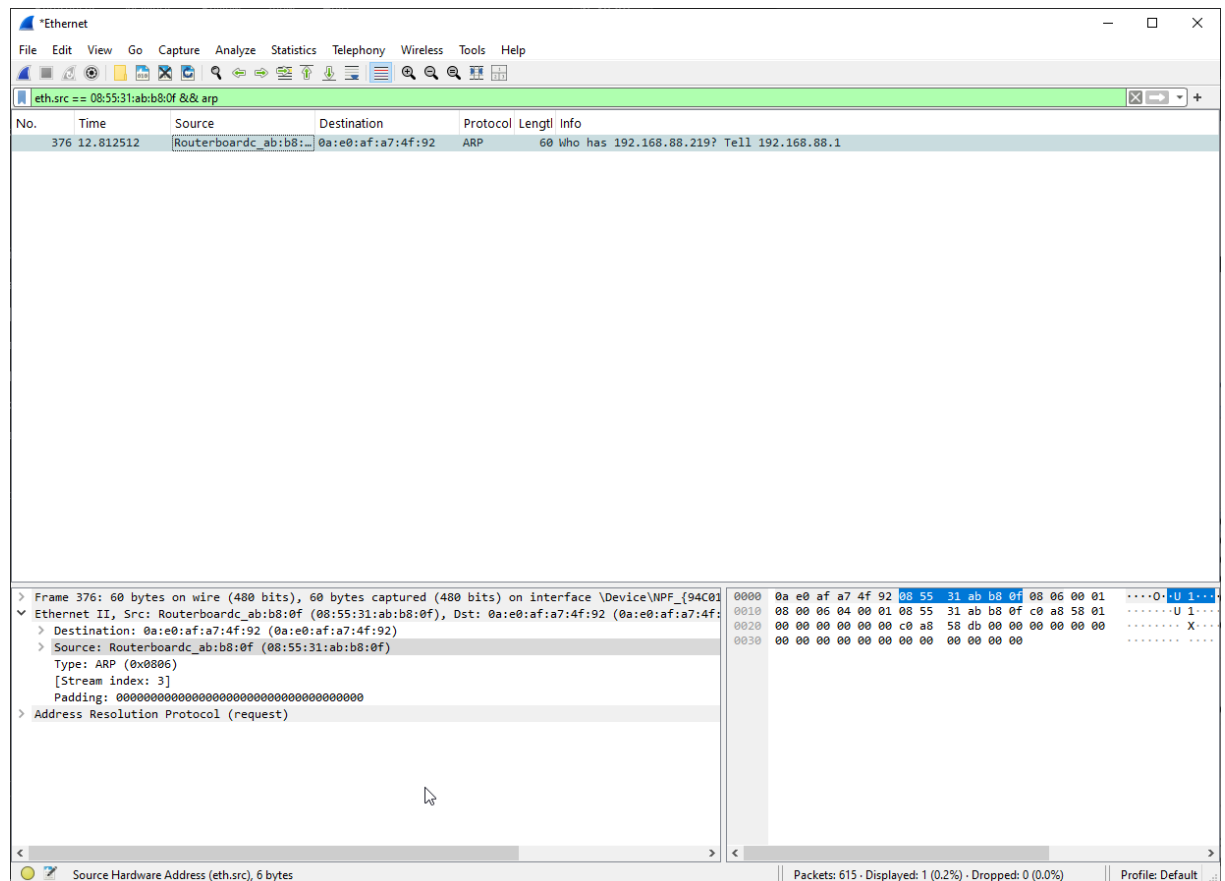


Рис. 11 – Захоплені пакети зі застосуванням фільтру та доданою фільтрацією arp

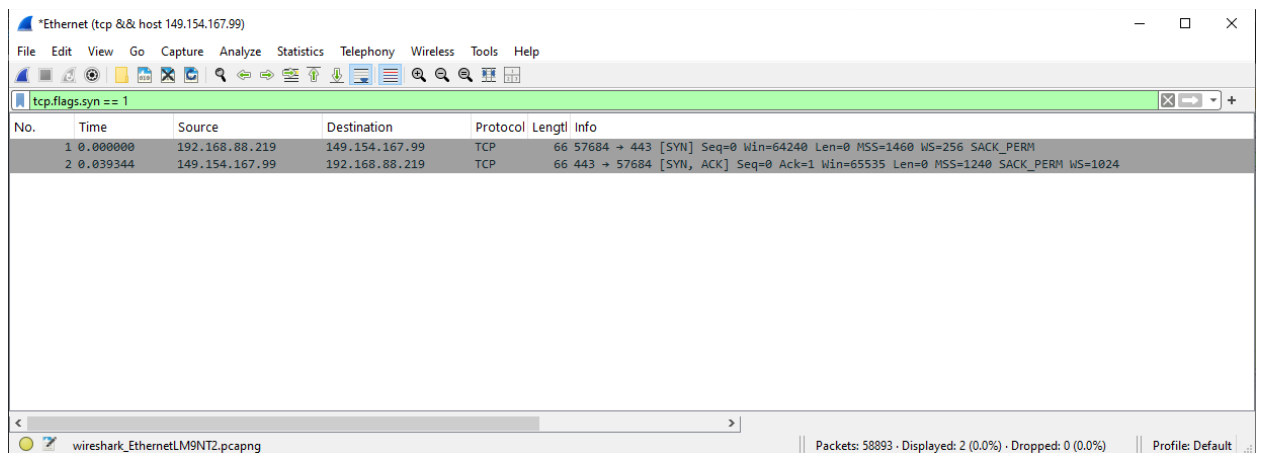


Рис. 12 – Захоплені пакети з прапорцем SYN під час завантаження файлу та доданим раніше фільтром захоплення

Висновки

Ознайомились з функціональними можливостями Wireshark для аналізу мережевого трафіку, зокрема з процесом застосування фільтрів. Використання фільтрів дозволило ефективно відсіяти зайвий трафік та зосередитися на аналізі конкретних протоколів, таких як TCP та IP, або на пакетах за певними критеріями. Це значно полегшило процес аналізу великого обсягу даних. Метою роботи було навчитися використовувати фільтри для точного визначення необхідних пакетів за заданими параметрами, такими як IP-адреси або порти. Було проведено аналіз інкапсуляції даних на різних рівнях мережевої моделі OSI та вивчено взаємодію між транспортним і мережевим рівнями.