

ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ імені ІВАНА ФРАНКА
Факультет прикладної математики та інформатики

**Комп'ютерні інформаційні мережі
ЛАБОРАТОРНА РОБОТА №8**

**Тема: «Аналіз TCP-сегментів та UDP-датаграм засобами
Wireshark».**

Виконала:
Ст. Пелєщак Вероніка
ПМІ-35с

Тема: Аналіз TCP-сегментів та UDP-датаграм засобами Wireshark.

Мета: Здобути практичні навички з інтерпретації протокольних блоків даних транспортного рівня стеку TCP/IP.

Хід роботи:

1. Використовуючи Wireshark, я захопила пакети під час активності в браузері, зокрема відвідувала **HTTP-сайти**. Також під час захоплення було завантажено три файли загальним розміром **18 МБ**.
2. Використала фільтр ***tcp || udp***.

No.	Time	Source	Destination	Protocol	Length	Info	Source Port	Host
754	19	192.168.0.163	142.251.38.164	QUIC	499	0-RTT, DCID=8b9a40e31abc269f	55417	
3403	32	192.168.0.181	192.168.0.255	UDP	77	32798 → 15600 Len=35	32798	
8394	74	192.168.0.181	192.168.0.255	UDP	77	35024 → 15600 Len=35	35024	
446	14	192.168.0.181	192.168.0.255	UDP	77	38097 → 15600 Len=35	38097	
6957	62	192.168.0.181	192.168.0.255	UDP	77	41845 → 15600 Len=35	41845	
2831	26	192.168.0.181	192.168.0.255	UDP	77	43045 → 15600 Len=35	43045	
3742	39	142.250.120.155	192.168.0.163	UDP	69	443 → 49846 Len=27	443	
3913	40	142.250.120.155	192.168.0.163	UDP	69	443 → 49846 Len=27	443	
4072	41	142.250.120.155	192.168.0.163	UDP	69	443 → 49846 Len=27	443	

3. Попри застосування цього фільтра, у вікні відображаються також інші протоколи, зокрема **HTTP, DNS, TLS, тощо**. Це відбувається тому, що такі протоколи належать до вищих рівнів мережової моделі (рівня застосунків) і передаються поверх **TCP або UDP**. Наприклад:

- HTTP використовує TCP для передачі веб-даних.
- DNS зазвичай працює поверх UDP (а інколи — TCP).

Тому, хоча фільтр обмежує відображення лише до транспортних протоколів, Wireshark все одно показує прикладні протоколи, оскільки вони передаються всередині **TCP або UDP-пакетів**.

4. Було вибрано пакет, що використовує протокол **UDP**.

У ньому зазначено:

- Порт відправника (Src Port): **32798**
- Порт одержувача (Dst Port): **15600**

capture8_.pcapng

tcp || udp

No.	Time	Source	Destination	Protocol	Length	Info	Source Port	Host
754	19	192.168.0.163	142.251.38.164	QUIC	499	0-RTT, DCID=8b9a40e31abc269f	55417	
3403	32	192.168.0.181	192.168.0.255	UDP	77	32798 → 15600 Len=35	32798	
8394	74	192.168.0.181	192.168.0.255	UDP	77	35024 → 15600 Len=35	35024	
446	14	192.168.0.181	192.168.0.255	UDP	77	38097 → 15600 Len=35	38097	
6957	62	192.168.0.181	192.168.0.255	UDP	77	41845 → 15600 Len=35	41845	
2831	26	192.168.0.181	192.168.0.255	UDP	77	43045 → 15600 Len=35	43045	
3742	39	142.250.120.155	192.168.0.163	UDP	69	443 → 49846 Len=27	443	
3913	40	142.250.120.155	192.168.0.163	UDP	69	443 → 49846 Len=27	443	
4072	41	142.250.120.155	192.168.0.163	UDP	69	443 → 49846 Len=27	443	
4207	42	142.250.120.155	192.168.0.163	UDP	69	443 → 49846 Len=27	443	
4220	42	142.250.120.155	192.168.0.163	UDP	69	443 → 49846 Len=27	443	
4236	42	142.250.120.155	192.168.0.163	UDP	69	443 → 49846 Len=27	443	

```
> Frame 3403: Packet, 77 bytes on wire (616 bits), 77 bytes captured (616 bits) on interface en0, id 0
> Ethernet II, Src: SamsungElect_0b:f8:05 (5c:c1:d7:0b:f8:05), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
> Internet Protocol Version 4, Src: 192.168.0.181, Dst: 192.168.0.255
> User Datagram Protocol, Src Port: 32798, Dst Port: 15600
    Source Port: 32798
    Destination Port: 15600
    Length: 43
    Checksum: 0x774d [unverified]
        [Checksum Status: Unverified]
        [Stream index: 44]
        [Stream Packet Number: 1]
    > [Timestamps]
        UDP payload (35 bytes)
```

Пояснення:

Порт 32798 є динамічним — його автоматично згенерувала операційна система відправника для встановлення з'єднання.

Порт 15600, навпаки, є закріпленим за певним сервісом або протоколом і використовується як порт одержувача для приймання даних.

5. Було вибрано пакет, що використовує протокол HTTP.

У ньому зазначено:

- Порт відправника (Src Port): **80**
- Порт одержувача (Dst Port): **61857**

capture8_.pcapng

tcp || udp

No.	Time	Source	Destination	Protocol	Length	Info	Source Port	Host
633	17	192.168.0.163	208.86.224.90	HTTP	528	GET /images/whoyou.gif HTTP/1.1	61854	textfiles.com
656	17	192.168.0.163	208.86.224.90	HTTP	529	GET /images/wmatter.gif HTTP/1.1	61857	textfiles.com
4592	44	192.168.0.163	90.130.70.73	HTTP	421	GET /js/bootstrap.min.js HTTP/1.1	61878	speedtest.tele2.net
4664	44	192.168.0.163	90.130.70.73	HTTP	416	GET /js/docs.min.js HTTP/1.1	61879	speedtest.tele2.net
2980	28	192.168.0.163	146.190.62.39	HTTP	408	GET /js/init.min.js HTTP/1.1	61869	httpforever.com
3312	29	146.190.62.39	192.168.0.163	HTTP/..	1393	HTTP/1.1 200 OK	80	
3314	29	146.190.62.39	192.168.0.163	HTTP/..	1343	HTTP/1.1 200 OK	80	
3323	29	146.190.62.39	192.168.0.163	HTTP/..	1349	HTTP/1.1 200 OK	80	
641	17	208.86.224.90	192.168.0.163	HTTP	1191	HTTP/1.1 200 OK (GIF89a)	80	
644	17	208.86.224.90	192.168.0.163	HTTP	1332	HTTP/1.1 200 OK (GIF89a)	80	
646	17	208.86.224.90	192.168.0.163	HTTP	1058	HTTP/1.1 200 OK (GIF89a)	80	
657	17	208.86.224.90	192.168.0.163	HTTP	1143	HTTP/1.1 200 OK (GIF89a)	80	
658	17	208.86.224.90	192.168.0.163	HTTP	1216	HTTP/1.1 200 OK (GIF89a)	80	
661	17	208.86.224.90	192.168.0.163	HTTP	1415	HTTP/1.1 200 OK (GIF89a)	80	
4873	44	90.130.70.73	192.168.0.163	HTTP	1069	HTTP/1.1 200 OK (JPEG JFIF image)	80	
3067	28	146.190.62.39	192.168.0.163	HTTP	464	HTTP/1.1 200 OK (application/javascript)	80	
4751	44	90.130.70.73	192.168.0.163	HTTP	559	HTTP/1.1 200 OK (application/javascript)	80	

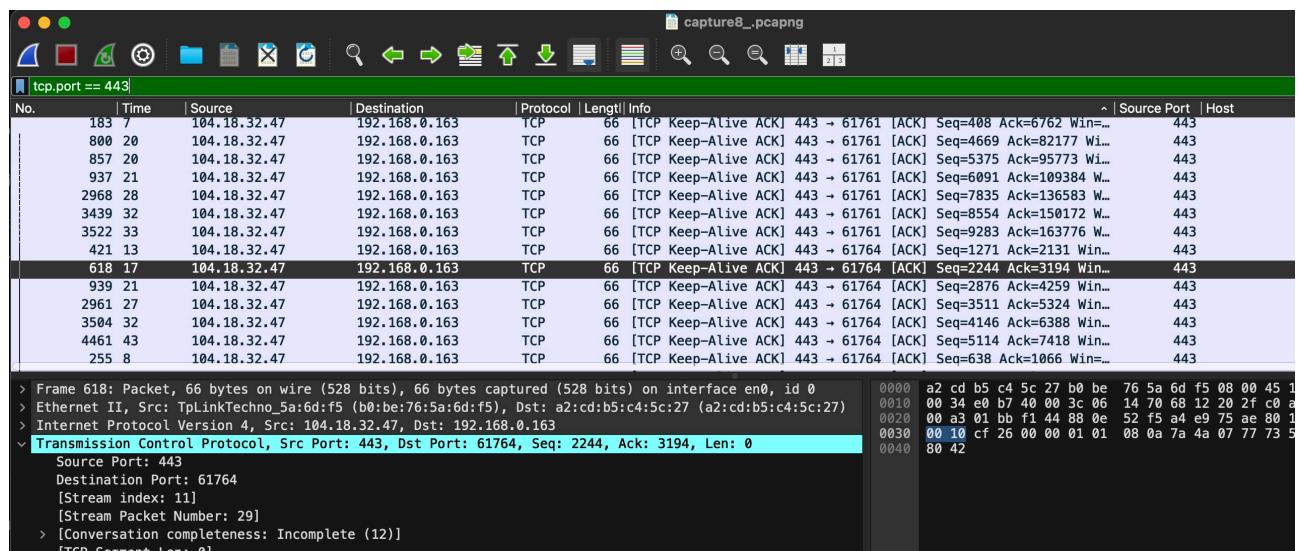
```
> Frame 661: Packet, 1415 bytes on wire (11320 bits), 1415 bytes captured (11320 bits) on interface en0, id 0
> Ethernet II, Src: TpLinkTechno_5a:6d:f5 (b0:be:76:5a:6d:f5), Dst: a2:cd:b5:c4:5c:27 (a2:cd:b5:c4:5c:27)
> Internet Protocol Version 4, Src: 208.86.224.90, Dst: 192.168.0.163
> Transmission Control Protocol, Src Port: 80, Dst Port: 61857, Seq: 1, Ack: 464, Len: 1349
    Source Port: 80
    Destination Port: 61857
    [Stream index: 22]
    [Stream Packet Number: 1]
```

Пояснення:

Порт 80 є стандартним (зарезервованим) портом, який закріплений за протоколом HTTP і використовується для передачі веб-сторінок.

Порт 61857 є динамічним — його автоматично згенерувала операційна система клієнта для встановлення з'єднання з вебсервером.

6. Для пошуку пакетів протоколу **HTTPS**, у Wireshark я застосувала фільтр:
tcp.port == 443.



The screenshot shows the Wireshark interface with a list of captured packets. A search filter is applied: `tcp.port == 443`. The table has columns: No., Time, Source, Destination, Protocol, Length, Info, and Source Port / Host. Most packets are from source 104.18.32.47 to destination 192.168.0.163, with lengths of 66 bytes and protocol TCP. The 'Info' column shows entries like "[TCP Keep-Alive ACK] 443 → 61761 [ACK] Seq=408 Ack=6762 Win=...". The last few rows show a single packet from 104.18.32.47 to 192.168.0.163, source port 443, destination port 61764, sequence number 2244, acknowledgement number 3194, and length 66 bytes. The 'Info' column for this row includes details about the frame structure and stream information. The bottom of the window shows the hex and ASCII representations of the selected packet.

No.	Time	Source	Destination	Protocol	Length	Info	Source Port	Host
183	7	104.18.32.47	192.168.0.163	TCP	66	[TCP Keep-Alive ACK] 443 → 61761 [ACK] Seq=408 Ack=6762 Win=...	443	
800	20	104.18.32.47	192.168.0.163	TCP	66	[TCP Keep-Alive ACK] 443 → 61761 [ACK] Seq=4669 Ack=82177 Win=...	443	
857	20	104.18.32.47	192.168.0.163	TCP	66	[TCP Keep-Alive ACK] 443 → 61761 [ACK] Seq=5375 Ack=95773 Win=...	443	
937	21	104.18.32.47	192.168.0.163	TCP	66	[TCP Keep-Alive ACK] 443 → 61761 [ACK] Seq=6091 Ack=109384 Win=...	443	
2968	28	104.18.32.47	192.168.0.163	TCP	66	[TCP Keep-Alive ACK] 443 → 61761 [ACK] Seq=7835 Ack=136583 Win=...	443	
3439	32	104.18.32.47	192.168.0.163	TCP	66	[TCP Keep-Alive ACK] 443 → 61761 [ACK] Seq=8554 Ack=150172 Win=...	443	
3522	33	104.18.32.47	192.168.0.163	TCP	66	[TCP Keep-Alive ACK] 443 → 61761 [ACK] Seq=9283 Ack=163776 Win=...	443	
421	13	104.18.32.47	192.168.0.163	TCP	66	[TCP Keep-Alive ACK] 443 → 61764 [ACK] Seq=1271 Ack=2131 Win=...	443	
618	17	104.18.32.47	192.168.0.163	TCP	66	[TCP Keep-Alive ACK] 443 → 61764 [ACK] Seq=2244 Ack=3194 Win=...	443	
939	21	104.18.32.47	192.168.0.163	TCP	66	[TCP Keep-Alive ACK] 443 → 61764 [ACK] Seq=2876 Ack=4259 Win=...	443	
2961	27	104.18.32.47	192.168.0.163	TCP	66	[TCP Keep-Alive ACK] 443 → 61764 [ACK] Seq=3511 Ack=5324 Win=...	443	
3504	32	104.18.32.47	192.168.0.163	TCP	66	[TCP Keep-Alive ACK] 443 → 61764 [ACK] Seq=4146 Ack=6388 Win=...	443	
4461	43	104.18.32.47	192.168.0.163	TCP	66	[TCP Keep-Alive ACK] 443 → 61764 [ACK] Seq=5114 Ack=7418 Win=...	443	
255	8	104.18.32.47	192.168.0.163	TCP	66	[TCP Keep-Alive ACK] 443 → 61764 [ACK] Seq=638 Ack=1064 Win=...	443	

Wireshark не відображає **HTTPS** у стовпці *Protocol*, бо дані цього протоколу зашифровані. Він не може розпізнати вміст, тому пакети показуються як **TCP (або TLS)**, тоді як HTTP передається відкритим текстом, тому його можна визначити точно.

7. Відшукала послідовність пакетів процедури «**потрійного рукостискання**».

114	16	192.168.0.163	188.184.67.127	TCP	78	65006 → 80 [SYN, ECE, CWR] Seq=0 Win=65535 Len=0 MSS=1460 WS...	65006
120	16	188.184.67.127	192.168.0.163	TCP	74	80 → 65006 [SYN, ACK, ECE] Seq=0 Ack=1 Win=65160 Len=0 MSS=1...	80
122	16	192.168.0.163	188.184.67.127	TCP	66	65006 → 80 [ACK] Seq=1 Ack=1 Win=131776 Len=0 TSval=28219708...	65006

Перший сегмент:

- **Src Port:** 65006 — порт, з якого надсилає клієнт.
- **Dst Port:** 80 — порт, на який надсилається запит на сервері (HTTP-порт).
- **Flags:** [SYN, ECE, CWR] — SYN означає ініціалізацію TCP-з'єднання; ECE і CWR сигналізують про підтримку контролю перевантаження (TCP congestion control).
- **Sequence Number:** 2000246939 — номер послідовності, який вказує на перший байт даних у цьому потоці (даних ще немає).
- **Ack Number:** 0 — оскільки це перший сегмент, клієнт ще не отримував даних від сервера, тому підтвердження відсутнє.

Клієнт ініціює з'єднання, повідомляючи серверу, що готовий почати обмін даними і підтримує механізми контролю перевантаження.

```
Transmission Control Protocol, Src Port: 65006, Dst Port: 80, Seq: 0, Len: 0
Source Port: 65006
Destination Port: 80
[Stream index: 5]
[Stream Packet Number: 1]
> [Conversation completeness: Complete, NO_DATA (55)]
[TCP Segment Len: 0]
Sequence Number: 0 (relative sequence number)
Sequence Number (raw): 2000246939
[Next Sequence Number: 1 (relative sequence number)]
Acknowledgment Number: 0
Acknowledgment number (raw): 0
1011 .... = Header Length: 44 bytes (11)
> Flags: 0x0c2 (SYN, ECE, CWR)
Window: 65535
```

Другий сегмент:

- **Src Port:** 80 — порт сервера, який відповідає на запит клієнта.
- **Dst Port:** 65006 — порт клієнта, на який сервер надсилає відповідь.

- **Flags:** [SYN, ACK, ECE] — SYN повідомляє про готовність сервера встановити з'єднання; ACK підтверджує отримання SYN від клієнта; ECE як і раніше відноситься до контролю перевантаження.
- **Sequence Number:** 3415540256 — власний номер послідовності сервера.
- **Ack Number:** 2000246940 — підтверджує отримання першого сегмента клієнта (наступний очікуваний байт).

Сервер відповідає на ініціацію з'єднання клієнта, підтверджує отримання SYN і готовий синхронізувати передачу даних.

```
> Internet Protocol Version 4, Src: 188.184.67.127, Dst: 192.168.0.163
< Transmission Control Protocol, Src Port: 80, Dst Port: 65006, Seq: 0, Ack: 1, Len: 0
  Source Port: 80
  Destination Port: 65006
  [Stream index: 5]
  [Stream Packet Number: 2]
  > [Conversation completeness: Complete, NO_DATA (55)]
    [TCP Segment Len: 0]
    Sequence Number: 0      (relative sequence number)
    Sequence Number (raw): 3415540256
    [Next Sequence Number: 1      (relative sequence number)]
    Acknowledgment Number: 1      (relative ack number)
    Acknowledgment number (raw): 2000246940
    1010 .... = Header Length: 40 bytes (10)
  > Flags: 0x052 (SYN, ACK, ECE)
```

Третій сегмент:

- **Src Port:** 65006 — порт клієнта.
- **Dst Port:** 80 — порт сервера.
- **Flags:** [ACK] — підтвердження отримання SYN-сегмента сервера.
- **Sequence Number:** 2000246940 — номер послідовності клієнта для наступного байта.
- **Ack Number:** 3415540257 — підтверджує отримання сегмента від сервера (наступний очікуваний байт).

```

> Internet Protocol Version 4, Src: 192.168.0.163, Dst: 188.184.67.127
└ Transmission Control Protocol, Src Port: 65006, Dst Port: 80, Seq: 1, Ack: 1, Len: 0
    Source Port: 65006
    Destination Port: 80
    [Stream index: 5]
    [Stream Packet Number: 3]
    > [Conversation completeness: Complete, NO_DATA (55)]
        [TCP Segment Len: 0]
        Sequence Number: 1 (relative sequence number)
        Sequence Number (raw): 2000246940
        [Next Sequence Number: 1 (relative sequence number)]
        Acknowledgment Number: 1 (relative ack number)
        Acknowledgment number (raw): 3415540257
        1000 .... = Header Length: 32 bytes (8)
    > Flags: 0x010 (ACK)

```

Клієнт підтверджує отримання сегмента від сервера, завершуючи трьохетапне рукостискання TCP і встановлюючи з'єднання для передачі даних через SCP.

8. Під час аналізу TCP-потоку у Wireshark перший сегмент від клієнта (*порт 65006 → 80*) показується з **відносним номером послідовності 0**, оскільки це перший байт у потоці — SYN займає один «байт» у нумерації. Його справжній номер послідовності (**ISN**) — **2000246939**. Сервер відповідає сегментом *SYN+ACK* (*порт 80 → 65006*), у якого теж **відносний номер 0**, бо для потоку від сервера до клієнта це перший байт; його справжній **ISN** — **3415540256**, а ACK клієнта підтверджує перший сегмент, тобто має значення **2000246940 (ISN клієнта + 1)**. Третій сегмент — ACK від клієнта, у Wireshark **відносний номер 1**, що означає «наступний байт після SYN», а справжнє значення — **2000246940**, ACK для сервера — **3415540257**. Таким чином, «0» і «1» у Wireshark відображають порядок байтів у потоці, а справжні числа використовуються TCP для синхронізації та підтвердження отриманих даних.

9. Отримала пакети криптографічного протоколу **TLS**, використовуючи фільтр *tls*. І відшукала пакети, які стосуються процедури **TLS-рукостискання**.

11827	26	192.168.0.163	104.18.16.5	TLSv1.3	215 Client Hello (SNI=ws6.qualified.com)	49691
11886	26	104.18.16.5	192.168.0.163	TLSv1.3	1514 Server Hello, Change Cipher Spec	443
11891	26	192.168.0.163	104.18.16.5	TLSv1.3	146 Change Cipher Spec, Application Data	49691

Перший сегмент:

Клієнт ініціює TLS-з'єднання, надсилаючи **Client Hello** (SNI=ws6.qualified.com) на сервер.

- **Src Port:** 49691 — порт клієнта, з якого надсилається запит.
- **Dst Port:** 443 — порт сервера (HTTPS), куди надсилається запит.
- **Sequence Number (relative):** 1389 — відносний номер послідовності для наочного порядку байтів у потоці.
- **Sequence Number (raw):** 3332809822 — справжній номер послідовності TCP (ISN клієнта).
- **Acknowledgment Number (raw):** 1035685341 — підтверджує отримання перших байтів від сервера.
- **Flags:** 0x018 (PSH, ACK) — PSH забезпечує негайну доставку даних, ACK сигналізує про підтвердження.

```
✓ Transmission Control Protocol, Src Port: 49691, Dst Port: 443, Seq: 1389, Ack: 1, Len: 149
  Source Port: 49691
  Destination Port: 443
  [Stream index: 12]
  [Stream Packet Number: 5]
  > [Conversation completeness: Incomplete, DATA (15)]
    [TCP Segment Len: 149]
    Sequence Number: 1389      (relative sequence number)
    Sequence Number (raw): 3332809822
    [Next Sequence Number: 1538      (relative sequence number)]
    Acknowledgment Number: 1      (relative ack number)
    Acknowledgment number (raw): 1035685341
    1000 .... = Header Length: 32 bytes (8)
  > Flags: 0x018 (PSH, ACK)
    Window: 2061
    [Calculated window size: 131904]
    [Window size scaling factor: 64]
    Checksum: 0xe42a [unverified]
    [Checksum Status: Unverified]
    Urgent Pointer: 0
```

Другий сегмент:

Сервер відповідає сегментом **Server Hello** та **Change Cipher Spec**, синхронізуючи параметри TLS і повідомляючи про готовність перейти на зашифровану передачу.

- **Src Port:** 443 — порт сервера, з якого надсилається відповідь.
- **Dst Port:** 49691 — порт клієнта, на який надсилається відповідь.
- **Sequence Number (relative):** 1 — перший байт у потоці сервера.
- **Sequence Number (raw):** 1035685341 — ISN сервера.
- **Acknowledgment Number (raw):** 3332809971 — підтверджує отримання сегмента Client Hello від клієнта.
- **Flags:** 0x010 (ACK) — сервер підтверджує отримання даних клієнта.

```

▼ Transmission Control Protocol, Src Port: 443, Dst Port: 49691, Seq: 1, Ack: 1538, Len: 1448
  Source Port: 443
  Destination Port: 49691
  [Stream index: 12]
  [Stream Packet Number: 7]
  > [Conversation completeness: Incomplete, DATA (15)]
    [TCP Segment Len: 1448]
    Sequence Number: 1      (relative sequence number)
    Sequence Number (raw): 1035685341
    [Next Sequence Number: 1449      (relative sequence number)]
    Acknowledgment Number: 1538      (relative ack number)
    Acknowledgment number (raw): 3332809971
    1000 .... = Header Length: 32 bytes (8)
  > Flags: 0x010 (ACK)
    Window: 16
    [Calculated window size: 131072]
    [Window size scaling factor: 8192]
    Checksum: 0xe019 [unverified]
    [Checksum Status: Unverified]
    Urgent Pointer: 0

```

Третій сегмент:

Клієнт надсилає **Change Cipher Spec** та початок **Application Data**, сигналізуючи про перехід на зашифровану передачу та початок захищеної передачі даних.

- **Src Port:** 49691 — порт клієнта, з якого надсилається зашифрований трафік.
- **Dst Port:** 443 — порт сервера, куди надсилається Application Data.

- **Sequence Number (relative):** 1538 — наступний байт після Client Hello у потоці клієнта.
- **Sequence Number (raw):** 3332809971 — справжній номер послідовності TCP.
- **Acknowledgment Number (raw):** 1035689558 — підтверджує отримання Server Hello та Change Cipher Spec від сервера.
- **Flags:** 0x018 (PSH, ACK) — PSH забезпечує негайну доставку даних, ACK підтверджує отримання серверних повідомлень.

```
Transmission Control Protocol, Src Port: 49691, Dst Port: 443, Seq: 1538, Ack: 4218, Len: 80
  Source Port: 49691
  Destination Port: 443
    [Stream index: 12]
    [Stream Packet Number: 12]
  > [Conversation completeness: Incomplete, DATA (15)]
    [TCP Segment Len: 80]
    Sequence Number: 1538      (relative sequence number)
    Sequence Number (raw): 3332809971
    [Next Sequence Number: 1618      (relative sequence number)]
    Acknowledgment Number: 4218      (relative ack number)
    Acknowledgment number (raw): 1035689558
    1000 .... = Header Length: 32 bytes (8)
  > Flags: 0x018 (PSH, ACK)
    Window: 2048
      [Calculated window size: 131072]
      [Window size scaling factor: 64]
      Checksum: 0xb8a7 [unverified]
      [Checksum Status: Unverified]
      Urgent Pointer: 0
```

10. Переконалась, що дані зашифровані.

```
> Internet Protocol Version 4, Src: 192.168.0.103, Dst: 104.18.10.5
  > Transmission Control Protocol, Src Port: 49691, Dst Port: 443, Seq: 1538, Ack: 4218, Len: 80
    < Transport Layer Security
      [Stream index: 11]
        > TLSv1.3 Record Layer: Change Cipher Spec Protocol: Change Cipher Spec
        < TLSv1.3 Record Layer: Application Data Protocol: Hypertext Transfer Protocol
          Opaque Type: Application Data (23)
          Version: TLS 1.2 (0x0303)
          Length: 69
          Encrypted Application Data: 8addaab4f06d7c92630b24b91d9ba20c12481150229238a92ef6190711b31d7dc3bc75965f7d8d8
            [Application Data Protocol: Hypertext Transfer Protocol]
```

Висновок: У ході лабораторної роботи було захоплено пакети за допомогою Wireshark під час відвідування HTTP-сайту та завантаження великого файлу. Було проаналізовано пакети TCP, UDP та TLS, встановлено фільтри, досліджено номери послідовності, порти та пропорці. Було відшукано процедури TCP-потрійного рукостискання і TLS-рукостискання, а також підтверджено, що дані TLS передаються зашифрованими.