

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ № 03

дисциплина: Сетевые технологии (09.03.03)

Анализ трафика в Wireshark

Студент: Стелина Петрити

Группа: НПИбд-02-21

МОСКВА

2023 г.

Цели работы

Изучение посредством Wireshark кадров Ethernet, анализ PDU протоколов транспортного и прикладного уровней стека TCP/IP.

3.3. Задания для выполнения

3.3.1. MAC-адресация

3.3.1.1. Постановка задачи

1. Изучение возможностей команды ipconfig для ОС типа Windows (ifconfig для систем типа Linux).
2. Определение MAC-адреса устройства и его типа.

3.3.1.2. Порядок выполнения работы

1. С помощью команды ipconfig для ОС типа Windows или ifconfig для систем типа Linux выведите информацию о текущем сетевом соединении. Используйте разные опции команды. В отчёте поясните детально полученную в каждом случае при выводе информацию. Подтвердите свой ответ скриншотами.

Ipconfig/all

Команда "ipconfig /all" в командной строке (Windows) используется для отображения подробной информации о сетевых интерфейсах и конфигурации вашего компьютера. Эта команда предоставляет обширную информацию о ваших сетевых адаптерах, IP-адресах, настройках DNS и многом другом. Когда вы запустите эту команду, вы увидите список сведений, связанных с сетью, для каждого сетевого адаптера в вашей системе.

рис.1. информация о сетевых подключениях

```
C:\WINDOWS\system32>Ipconfig/all

Windows IP Configuration

Host Name . . . . . : DESKTOP-BKDNN1F
Primary Dns Suffix . . . . . :
Node Type . . . . . : Hybrid
IP Routing Enabled. . . . . : No
WINS Proxy Enabled. . . . . : No

Wireless LAN adapter Local Area Connection* 10:
Media State . . . . . : Media disconnected
Connection-specific DNS Suffix . . . . . :
Description . . . . . : Microsoft Wi-Fi Direct Virtual Adapter #3
Physical Address. . . . . : 76-4C-A1-DD-EE-8B
DHCP Enabled. . . . . : Yes
Autoconfiguration Enabled . . . . . : Yes

Wireless LAN adapter Local Area Connection* 12:
Media State . . . . . : Media disconnected
Connection-specific DNS Suffix . . . . . :
Description . . . . . : Microsoft Wi-Fi Direct Virtual Adapter #4
Physical Address. . . . . : 86-4C-A1-DD-EE-8B
DHCP Enabled. . . . . : Yes
Autoconfiguration Enabled . . . . . : Yes

Wireless LAN adapter Wi-Fi:
Connection-specific DNS Suffix . . . . . :
Description . . . . . : Qualcomm Atheros QCA61x4A Wireless Network Adapter
Physical Address. . . . . : 74-4C-A1-DD-EE-8B
DHCP Enabled. . . . . : Yes
Autoconfiguration Enabled . . . . . : Yes
Link-local IPv6 Address . . . . . : fe80::c3a3:4792%5a5c%56f4%7(PREFERRED)
```

2. Определите MAC-адреса сетевых интерфейсов на вашем компьютере. Подтвердите свой ответ скриншотом.

"Физический адрес" под каждым сетевым интерфейсом - это MAC-адрес.

Например:

```
Wireless LAN adapter Local Area Connection* 10:  
  
    Media State . . . . . : Media disconnected  
    Connection-specific DNS Suffix . :  
    Description . . . . . : Microsoft Wi-Fi Direct Virtual Adapter #3  
    Physical Address. . . . . : 76-4C-A1-DD-EE-8B  
    DHCP Enabled. . . . . : Yes  
    Autoconfiguration Enabled . . . . . : Yes
```

рис.2. Physical Address- MAC-адрес

3. Опишите структуру MAC-адресов вашего устройства. Какая часть адреса идентифицирует производителя? Какая часть адреса идентифицирует сетевой интерфейс? Определите, каким является адрес — индивидуальным или групповым, глобально администрируемым или локально администрируемым.

Поясните свой ответ. Используйте шестнадцатеричную запись MAC-адреса для пояснения.

Префикс производителя (OUI - Organizationally Unique Identifier): Первые 6 символов MAC-адреса идентифицируют производителя оборудования. Этот префикс уникален для каждого производителя и называется OUI.

Идентификатор устройства (NIC - Network Interface Controller): После OUI идет последовательность символов, которая уникально идентифицирует сетевой интерфейс устройства, произведенного этим производителем.

Мой MAC-адрес: 76-4C-A1-DD-EE-8B

76-4C-A1 идентифицирует производителя (OUI)

-DD-EE-8B идентифицирует конкретный сетевой интерфейс(NIC-specific identifier)

3.3.2. Анализ кадров канального уровня в Wireshark

3.3.2.1. Постановка задачи

1. Установить на домашнем устройстве Wireshark.

2. С помощью Wireshark захватить и проанализировать пакеты ARP и ICMP в части кадров канального уровня.

3.3.2. Анализ кадров канального уровня в Wireshark

3.3.2.1. Постановка задачи

1. Установить на домашнем устройстве Wireshark.

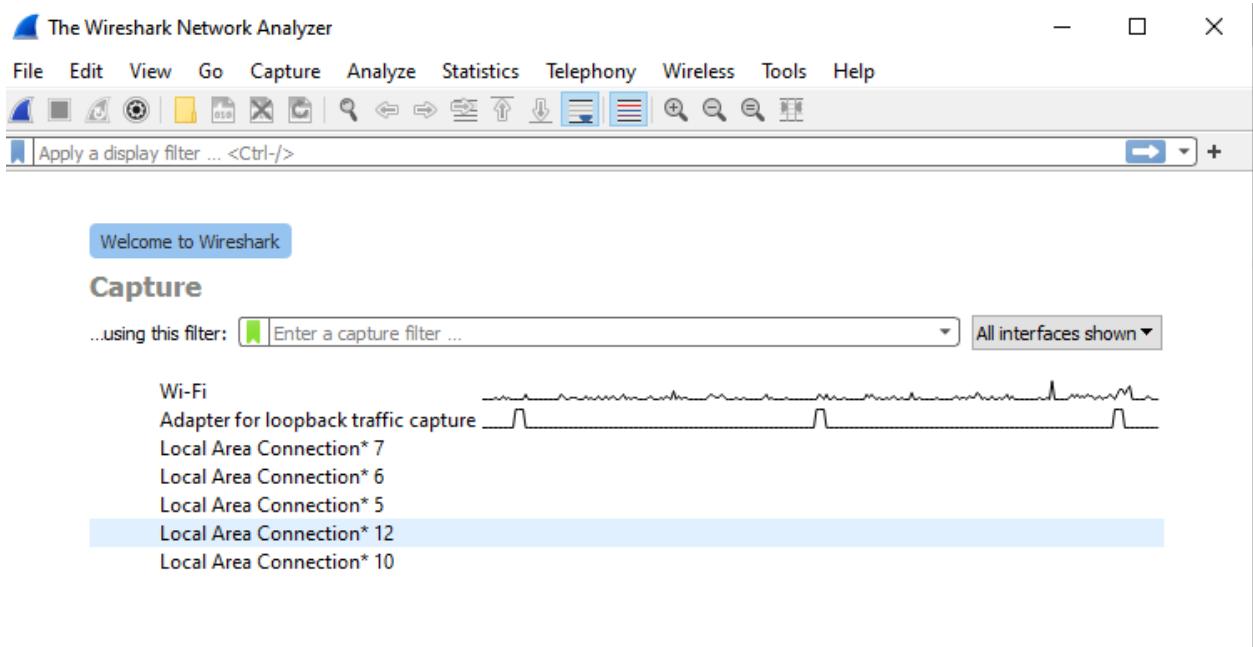


рис.3. Установите Wireshark на свое домашнее устройство.

2. С помощью Wireshark захватить и проанализировать пакеты ARP и ICMP в части кадров канального уровня. Убедитесь, что начался процесс захвата трафика.

No.	Time	Source	Destination	Protocol	Length	I
16073	475.252775	LiteonTe_dd:ee:8b	TaicangT_ba:b7:40	ARP	42	1
16093	480.301423	TaicangT_ba:b7:40	LiteonTe_dd:ee:8b	ARP	60	1
16094	480.301444	LiteonTe_dd:ee:8b	TaicangT_ba:b7:40	ARP	42	1
16102	485.352559	TaicangT_ba:b7:40	LiteonTe_dd:ee:8b	ARP	60	1
16103	485.352581	LiteonTe_dd:ee:8b	TaicangT_ba:b7:40	ARP	42	1
16142	490.402532	TaicangT_ba:b7:40	LiteonTe_dd:ee:8b	ARP	60	1
16143	490.402559	LiteonTe_dd:ee:8b	TaicangT_ba:b7:40	ARP	42	1
16177	495.451750	TaicangT_ba:b7:40	LiteonTe_dd:ee:8b	ARP	60	1
16178	495.451774	LiteonTe_dd:ee:8b	TaicangT_ba:b7:40	ARP	42	1
16280	500.501547	TaicangT_ba:b7:40	LiteonTe_dd:ee:8b	ARP	60	1
16281	500.501584	LiteonTe_dd:ee:8b	TaicangT_ba:b7:40	ARP	42	1

Frame 3: 60 bytes on wire (480 bits), 60 bytes captured
Ethernet II, Src: TaicangT_ba:b7:40 (e8:d0:b9:ba:b7:40), Dst: LiteonTe_dd:ee:8b (08:00:06:04:00:01)
Address Resolution Protocol (request)

0000 74 4c a1 dd ee 8b e1
0010 08 00 06 04 00 01 e1
0020 00 00 00 00 00 00 c1
0030 00 00 00 00 00 00 01

рис.4.пакет ARP

No.	Time	Source	Protocol	Length	Info
599	27.987610	34.84.0.87	ICMP	70	Destination unreachable (
600	27.988223	34.84.0.87	ICMP	70	Destination unreachable (
603	28.105612	34.84.0.87	ICMP	70	Destination unreachable (
604	28.306742	34.84.0.87	ICMP	70	Destination unreachable (
634	28.698497	34.84.0.87	ICMP	70	Destination unreachable (
642	29.462723	34.84.0.87	ICMP	70	Destination unreachable (
644	30.994594	34.84.0.87	ICMP	70	Destination unreachable (
725	31.988872	34.84.0.87	ICMP	70	Destination unreachable (
726	31.990115	34.84.0.87	ICMP	70	Destination unreachable (
727	31.990480	34.84.0.87	ICMP	70	Destination unreachable (
733	32.298995	34.84.0.87	ICMP	70	Destination unreachable (

> Frame 634: 70 bytes on wire (560 bits), 70 bytes captured (560 bits) on interface
 > Ethernet II, Src: TaicangT_ba:b7:40 (e8:d0:1f) [eth0]
 > Internet Protocol Version 4, Src: 34.84.0.87 [eth0]
 > Internet Control Message Protocol

рис.5.пакет ICMP

Чтобы увидеть, начался ли захват трафика, нужно просто понаблюдать, как новые строки информации о пакетах добавляются в ARP и ICMP окна Wireshark. Когда вы видите какое-либо действие, это означает, что процесс захвата пакетов работает.

3. На вашем устройстве в консоли определите с помощью команды ipconfig для ОС типа Windows или ifconfig для систем типа Linux IP-адрес вашего устройства и шлюз по умолчанию (default gateway).

рис.5. IP-адрес и шлюз по умолчанию (default gateway)

```
C:\WINDOWS\system32>ipconfig

Windows IP Configuration

Wireless LAN adapter Local Area Connection* 10:
  Media State . . . . . : Media disconnected
  Connection-specific DNS Suffix . . :

Wireless LAN adapter Local Area Connection* 12:
  Media State . . . . . : Media disconnected
  Connection-specific DNS Suffix . . :

Wireless LAN adapter Wi-Fi:
  Connection-specific DNS Suffix . . .
  Link-local IPv6 Address . . . . . : fe80::c3a3:4792:5a5c:56f4%7
  IPv4 Address . . . . . :
  Subnet Mask . . . . . :
  Default Gateway . . . . . :
```

4. На вашем устройстве в консоли с помощью команды ping адрес_шлюза пропингуйте шлюз по умолчанию. Для остановки процесса используйте комбинацию клавиш Ctrl + c или изначально при помощи параметров команды ping задайте число сообщений эхо-запроса.

```
Pinging 192.168.1.1 with 32 bytes of data:  
Reply from 192.168.1.1: bytes=32 time=2ms TTL=64  
Reply from 192.168.1.1: bytes=32 time=2ms TTL=64  
Reply from 192.168.1.1: bytes=32 time=3ms TTL=64  
Reply from 192.168.1.1: bytes=32 time=3ms TTL=64  
  
Ping statistics for 192.168.1.1:  
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss)  
    Approximate round trip times in milli-seconds:  
        Minimum = 2ms, Maximum = 3ms, Average = 2ms  
  
C:\WINDOWS\system32>
```

рис.6. ping адрес_шлюза пропингуйте шлюз по умолчанию

```
C:\WINDOWS\system32>  
  
C:\WINDOWS\system32>ipconfig  
  
Windows IP Configuration  
  
Wireless LAN adapter Local Area Connection* 10:  
    Media State . . . . . : Media disconnected  
    Connection-specific DNS Suffix . :  
  
Wireless LAN adapter Local Area Connection* 12:  
    Media State . . . . . : Media disconnected  
    Connection-specific DNS Suffix . :  
  
Wireless LAN adapter Wi-Fi:  
    Connection-specific DNS Suffix . :  
    Link-local IPv6 Address . . . . . :  
    IPv4 Address . . . . . : 192.168.1.1  
    Subnet Mask . . . . . : 255.255.255.0  
    Default Gateway . . . . . :  
  
C:\WINDOWS\system32>
```

рис.7. ping отображает соответствующее имя хоста, поэтому ctrl + c сработало

5. В Wireshark остановите захват трафика. В строке фильтра пропишите фильтр arp or icmp. Убедитесь, что в списке пакетов отобразятся только пакеты ARP или ICMP, в частности пакеты, которые были сгенерированы с помощью команды ping, отправленной с вашего устройства на шлюз по умолчанию.

No.	Time	Source	Destination	Protocol	Length	Info
113	28.200866	LiteonTe_dd:ee:8b	TaicangT_ba:b7:40	ARP	42	1 is at 7
143	25.249941	TaicangT_ba:b7:40	LiteonTe_dd:ee:8b	ARP	60	
144	25.249979	LiteonTe_dd:ee:8b	TaicangT_ba:b7:40	ARP	42	: is at 7
177	30.301523	TaicangT_ba:b7:40	LiteonTe_dd:ee:8b	ARP	60	Who has
178	30.301544	LiteonTe_dd:ee:8b	TaicangT_ba:b7:40	ARP	42	
212	35.349968	TaicangT_ba:b7:40	LiteonTe_dd:ee:8b	ARP	60	Who has
213	35.350006	LiteonTe_dd:ee:8b	TaicangT_ba:b7:40	ARP	42	
236	40.399909	TaicangT_ba:b7:40	LiteonTe_dd:ee:8b	ARP	60	Who has
237	40.399930	LiteonTe_dd:ee:8b	TaicangT_ba:b7:40	ARP	42	
346	45.452784	TaicangT_ba:b7:40	LiteonTe_dd:ee:8b	ARP	60	Who has
347	45.452808	LiteonTe_dd:ee:8b	TaicangT_ba:b7:40	ARP	42	

> Frame 1: 60 bytes on wire (480 bits), 60 bytes captured
> Ethernet II, Src: TaicangT_ba:b7:40 (e8:d0:b9:ba:b7:40)
> Address Resolution Protocol (request)

0000	74 4c a1 dd ee 8b e8 d0 b9 ba b7 40 00 00 00 00
0010	08 00 06 04 00 01 e8 d0 b9 ba b7 40 c0 a8 01
0020	00 00 00 00 00 00 c0 a8 01 6a 00 00 00 00 00
0030	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

рис.7.остановка перехвата трафика ARP

No.	Time	Source	Destination	Protocol	Length	Info
114	21.096359			ICMP	74	Echo (ping) request id=0x0
115	21.180155			ICMP	74	Echo (ping) reply id=0x0
116	22.180232			ICMP	74	Echo (ping) request id=0x0
117	22.118999			ICMP	74	Echo (ping) reply id=0x0
129	23.127318	1		ICMP	74	Echo (ping) request id=0x0
130	23.130574			ICMP	74	Echo (ping) reply id=0x0
137	24.145432			ICMP	74	Echo (ping) request id=0x0
138	24.149664	1		ICMP	74	Echo (ping) reply id=0x0

> Frame 114: 74 bytes on wire (592 bits), 74 bytes captured
> Ethernet II, Src: LiteonTe_dd:ee:8b (74:4c:a1:dd:ee:8b)
> Internet Protocol Version 4, Src: 192.168.1.1
> Internet Control Message Protocol

0000	c8 d9 b9 ba b7 48 74 4c a1 dd ee 8b 00 00 45
0010	08 3c 45 82 00 00 00 81 71 06 c0 a8 01 6a ca
0020	82 fe 88 00 4d 52 00 81 00 09 31 82 63 04 05
0030	57 68 69 66 00 00 00 8e 0f 70 71 72 73 74 75
0040	77 41 42 43 64 05 66 07 68 69

рис.8.остановка перехвата трафика ICMP

6. Изучите эхо-запрос и эхо-ответ ICMP в программе Wireshark:

– На панели списка пакетов (верхний раздел) выберите первый указанный кадр ICMP — эхо-запрос. Изучите информацию на панели сведений о пакете в средней части экрана. В отчёте укажите длину кадра, к какому типу Ethernet относится кадр, определите MAC-адреса источника и шлюза, определите тип MAC-адресов.

- длина кадра: 74 bytes (592 bits)
- тип Ethernet : Ethernet II
- MAC-адрес источника :74:4c:a1:dd:ee:8b
- MAC-адрес шлюза: e8:d0:b9:ba:b7:40
- тип MAC-адресов:IPv4 (0.0800) рис.10

рис.9.первый указанный кадр ICMP — эхо-запрос Frame:397

No.	Time	Source	Destination	Protocol	Length	Info
397	70.699622			ICMP	74	Echo (ping) requ
401	71.715646			ICMP	74	Echo (ping) requ
405	72.731306			ICMP	74	Echo (ping) requ
407	73.742590			ICMP	74	Echo (ping) requ
398	70.702382			ICMP	74	Echo (ping) repl
402	71.718357			ICMP	74	Echo (ping) repl
406	72.737171		6	ICMP	74	Echo (ping) repl
408	73.745353			ICMP	74	Echo (ping) repl

```
<
> Frame 397: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface \Device\NP
> Ethernet II. Src: LiteonTe dd:ee:8b (74:4c:a1:dd:ee:8b). Dst: TaicangT ba:b7:40 (e8:d0:b9:ba:b7:40)

```

рис.10.mun MAC-адресов

```
▼ Ethernet II, Src: LiteonTe_dd:ee:8b (74:4c:a1:dd:ee:8b), Dst: TaicangT_ba:b7:40 (e8:d0:b9:ba:b7:40)
  > Destination: TaicangT_ba:b7:40 (e8:d0:b9:ba:b7:40)
  > Source: LiteonTe_dd:ee:8b (74:4c:a1:dd:ee:8b)
  Type: IPv4 (0x0800)
```

– На панели списка пакетов (верхний раздел) выберите второй указанный кадр ICMP — эхо-ответ. Изучите информацию на панели сведений о пакете в средней части экрана. В отчёте укажите длину кадра, к какому типу Ethernet относится кадр, определите MAC-адреса источника и шлюза, определите тип MAC-адресов.

408 73.745353	4	36	ICMP	74 Echo (ping) reply id=0
17824 1113.307321			ICMP	70 Destination unreachable (
17825 1113.307321			ICMP	70 Destination unreachable (
17826 1113.375403			ICMP	70 Destination unreachable (
<				>
> Frame 408: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface \Device\NPF_{2EB515}				
▼ Ethernet II, Src: TaicangT_ba:b7:40 (e8:d0:b9:ba:b7:40), Dst: LiteonTe_dd:ee:8b (74:4c:a1:dd:ee:8b)				
> Destination: LiteonTe_dd:ee:8b (74:4c:a1:dd:ee:8b)				
> Source: TaicangT_ba:b7:40 (e8:d0:b9:ba:b7:40)				
Type: IPv4 (0x0800)				
> Internet Protocol Version 4, Src: [REDACTED]4, Dst: [REDACTED]6				
> Internet Control Message Protocol				

рис.11.второй указанный кадр ICMP — эхо-ответ Frame 408

- длина кадра: 74 bytes (592 bits)
- тип Ethernet : Ethernet II
- MAC-адрес источника :e8:d0:b9:ba:b7:40
- MAC-адрес шлюза: 74:4c:a1:dd:ee:8b
- тип MAC-адресов:IPv4 (0.0800)

7. Изучите кадры данных протокола ARP. Изучите данные в полях заголовка Ethernet II.

рис.12.кадры данных протокола ARP

.	Time	Source	Destination	Protocol	Length	Info
177	-2065.883520	LiteonTe_dd:ee:8b	TaicangT_ba:b7:40	ARP	42	192
179	-2060.830618	LiteonTe_dd:ee:8b	TaicangT_ba:b7:40	ARP	42	192
203	-2055.783318	LiteonTe_dd:ee:8b	TaicangT_ba:b7:40	ARP	42	192
256	-2050.733299	LiteonTe_dd:ee:8b	TaicangT_ba:b7:40	ARP	42	192
260	-2045.683230	LiteonTe_dd:ee:8b	TaicangT_ba:b7:40	ARP	42	192
268	-2040.632412	LiteonTe_dd:ee:8b	TaicangT_ba:b7:40	ARP	42	192
319	-2035.583355	LiteonTe_dd:ee:8b	TaicangT_ba:b7:40	ARP	42	192
375	-2030.533300	LiteonTe_dd:ee:8b	TaicangT_ba:b7:40	ARP	42	192
390	-2025.480378	LiteonTe_dd:ee:8b	TaicangT_ba:b7:40	ARP	42	192
394	-2020.432895	LiteonTe_dd:ee:8b	TaicangT_ba:b7:40	ARP	42	192
404	-2015.383047	LiteonTe_dd:ee:8b	TaicangT_ba:b7:40	ARP	42	192

Frame 177: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface Ethernet II, Src: LiteonTe_dd:ee:8b (74:4c:a1:dd:ee:8b), Dst: TaicangT_ba:b7:40
> Destination: TaicangT_ba:b7:40 (e8:d0:b9:ba:b7:40)
> Source: LiteonTe_dd:ee:8b (74:4c:a1:dd:ee:8b)
Type: ARP (0x0806)
Address Resolution Protocol (reply)

Тип Ethernet II

- MAC-адрес источника:(74:4c:a1:dd:ee:8b) Поле MAC-адреса источника представляет адрес управления доступом к мультимедиа (MAC) устройства, отправившего пакет ARP. Он идентифицирует аппаратный адрес отправителя (MAC-адрес) в локальной сети.
- MAC-адрес шлюза:(e8:d0:b9:ba:b7:40)Поле MAC-адрес назначения представляет собой MAC-адрес устройства-получателя или широковещательный адрес.
- Тип:(ARP (0x0806))Поле Type указывает тип фрейма Ethernet. В ARP-пакетах значение типа равно 0x0806, что является шестнадцатеричным представлением. Это значение указывает, что кадр содержит данные протокола ARP, позволяя принимающему устройству понять, что пакет является кадром ARP.

8. Начните новый процесс захвата трафика в Wireshark. На вашем устройстве в консоли пропингуйте по имени какой-нибудь известный вам адрес, например ping ruds.ru.

```
C:\WINDOWS\system32>ping google.com

Pinging google.com [192.168.1.1] with 32 bytes of data:
Reply from 192.168.1.1: bytes=32 time=12ms TTL=118
Reply from 192.168.1.1: bytes=32 time=12ms TTL=118
Reply from 192.168.1.1: bytes=32 time=15ms TTL=118
Reply from 192.168.1.1: bytes=32 time=12ms TTL=118

Ping statistics for 192.168.1.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 12ms, Maximum = 15ms, Average = 12ms

C:\WINDOWS\system32>
```

рис.13. ping google.com

Сначала ICMP:

Table showing captured ICMP traffic:

No.	Time	Source	Destination	Protocol	Length	Info
470	30.908321	ICMP	74	Echo (ping) request id=0..
471	30.922073	ICMP	74	Echo (ping) reply id=0..
486	31.926167	ICMP	74	Echo (ping) request id=0..
487	31.940930	ICMP	74	Echo (ping) reply id=0..
489	32.944333	ICMP	74	Echo (ping) request id=0..
490	32.958606	ICMP	74	Echo (ping) reply id=0..
496	33.969521	ICMP	74	Echo (ping) request id=0..
497	33.982662	ICMP	74	Echo (ping) reply id=0..

```
> Frame 470: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface \Device\NPF_{2EB515
└Ethernet II, Src: LiteonTe_dd:ee:8b (74:4c:a1:dd:ee:8b), Dst: TaicangT_ba:b7:40 (e8:d0:b9:ba:b7:40)
  > Destination: TaicangT_ba:b7:40 (e8:d0:b9:ba:b7:40)
  > Source: LiteonTe_dd:ee:8b (74:4c:a1:dd:ee:8b)
  Type: IPv4 (0x0800)
> Internet Protocol Version 4, Src: ..., Dst: ...
> Internet Control Message Protocol
```

рис.14. ICMP — эхо-запрос

- длина кадра: 74 bytes (592 bits)
- тип Ethernet : Ethernet II
- MAC-адрес источника :74:4c:a1:dd:ee:8b
- MAC-адрес шлюза: e8:d0:b9:ba:b7:40
- тип MAC-адресов:IPv4 (0.0800)

Table showing captured ICMP frames:

No.	Time	Source	Destination	Protocol	Length	Info
470	30.908321		4	ICMP	74	Echo (ping) request id=0..
471	30.922073	4	6	ICMP	74	Echo (ping) reply id=0..
486	31.926167	5		ICMP	74	Echo (ping) request id=0..
487	31.940930	8		ICMP	74	Echo (ping) reply id=0..
489	32.944333		1	ICMP	74	Echo (ping) request id=0..
490	32.958606			ICMP	74	Echo (ping) reply id=0..
496	33.969521	1		ICMP	74	Echo (ping) request id=0..
497	33.982662	2		ICMP	74	Echo (ping) reply id=0..

Detailed analysis of frame 471:

- > Frame 471: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface \Device\NPF_{2EB515}
- > Ethernet II, Src: TaicangT_ba:b7:40 (e8:d0:b9:ba:b7:40), Dst: LiteonTe_dd:ee:8b (74:4c:a1:dd:ee:8b)
 - > Destination: LiteonTe_dd:ee:8b (74:4c:a1:dd:ee:8b)
 - > Source: TaicangT_ba:b7:40 (e8:d0:b9:ba:b7:40)
 - Type: IPv4 (0x0800)
- > Internet Protocol Version 4, Src: 192.168.1.4, Dst: 192.168.1.6
- > Internet Control Message Protocol

рис.15. ICMP — эхо-ответ

- длина кадра: 74 bytes (592 bits)
- тип Ethernet : Ethernet II
- MAC-адрес источника :e8:d0:b9:ba:b7:40
- MAC-адрес шлюза: 74:4c:a1:dd:ee:8b
- тип MAC-адресов:IPv4 (0.0800)

ARP:

Table showing captured ARP frames:

No.	Time	Source	Destination	Protocol	Length	Info
9	2.983097	LiteonTe_dd:ee:8b	TaicangT_ba:b7:40	ARP	42	1 is at...
10	3.071924	TaicangT_ba:b7:40	Broadcast	ARP	42	Who has ...
11	4.108974	TaicangT_ba:b7:40	Broadcast	ARP	42	Who has ...
16	8.273225	TaicangT_ba:b7:40	LiteonTe_dd:ee:8b	ARP	60	Who has ... 1..
17	8.273246	LiteonTe_dd:ee:8b	TaicangT_ba:b7:40	ARP	42	... is at...
84	12.503632	TaicangT_ba:b7:40	LiteonTe_dd:ee:8b	ARP	60	Who has ... 1..
85	12.503655	LiteonTe_dd:ee:8b	TaicangT_ba:b7:40	ARP	42	... is at...
86	17.520968	TaicangT_ba:b7:40	LiteonTe_dd:ee:8b	ARP	60	Who has ...
87	17.521004	LiteonTe_dd:ee:8b	TaicangT_ba:b7:40	ARP	42	! is at...
136	22.538169	TaicangT_ba:b7:40	LiteonTe_dd:ee:8b	ARP	60	Who has 1 ..
137	22.538207	LiteonTe_dd:ee:8b	TaicangT_ba:b7:40	ARP	42	... is at...

Detailed analysis of frame 87:

- > Frame 87: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface \Device\NPF_{2EB5150}
- > Ethernet II, Src: LiteonTe_dd:ee:8b (74:4c:a1:dd:ee:8b), Dst: TaicangT_ba:b7:40 (e8:d0:b9:ba:b7:40)
 - > Destination: TaicangT_ba:b7:40 (e8:d0:b9:ba:b7:40)
 - > Source: LiteonTe_dd:ee:8b (74:4c:a1:dd:ee:8b)
 - Type: ARP (0x0806)
- > Address Resolution Protocol (reply)

рис.16. ARP — эхо-запрос

Тип Ethernet II

- MAC-адрес источника:74:4c:a1:dd:ee:8b
- MAC-адрес шлюза:e8:d0:b9:ba:b7:40
- Тип:ARP (0x0806)

No.	Time	Source	Destination	Protocol	Length	Info
9	2.983097	LiteonTe_dd:ee:8b	TaicangT_ba:b7:40	ARP	42	is at...
10	3.071924	TaicangT_ba:b7:40	Broadcast	ARP	42	Who has ... is at...
11	4.108974	TaicangT_ba:b7:40	Broadcast	ARP	42	Who has 19...
16	8.273225	TaicangT_ba:b7:40	LiteonTe_dd:ee:8b	ARP	60	Who has 1...
17	8.273246	LiteonTe_dd:ee:8b	TaicangT_ba:b7:40	ARP	42	1... 6 is at...
84	12.503632	TaicangT_ba:b7:40	LiteonTe_dd:ee:8b	ARP	60	Who has ... is at...
85	12.503655	LiteonTe_dd:ee:8b	TaicangT_ba:b7:40	ARP	42	1... 5 is at...
86	17.520968	TaicangT_ba:b7:40	LiteonTe_dd:ee:8b	ARP	60	Who has 1 1...
87	17.521004	LiteonTe_dd:ee:8b	TaicangT_ba:b7:40	ARP	42	1... 1... is at...
136	22.538169	TaicangT_ba:b7:40	LiteonTe_dd:ee:8b	ARP	60	Who has 1...
137	22.538207	LiteonTe_dd:ee:8b	TaicangT_ba:b7:40	ARP	42	1... 1... is at...

```

> Frame 136: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface \Device\NPF_{2EB515
> Ethernet II, Src: TaicangT_ba:b7:40 (e8:d0:b9:ba:b7:40), Dst: LiteonTe_dd:ee:8b (74:4c:a1:dd:ee:8b)
  > Destination: LiteonTe_dd:ee:8b (74:4c:a1:dd:ee:8b)
  > Source: TaicangT_ba:b7:40 (e8:d0:b9:ba:b7:40)
    Type: ARP (0x0806)
      Padding: 00000000000000000000000000000000
  > Address Resolution Protocol (request)

```

рис.17. ICMP — эхо-ответ

Тип Ethernet II

- MAC-адрес источника:e8:d0:b9:ba:b7:40
- MAC-адрес шлюза: 74:4c:a1:dd:ee:8b
- Тип:ARP (0x0806)

9. В Wireshark остановите захваттрафика. Изучите запросы и ответы протоколов ARP и ICMP. Определите MAC-адреса источника и получателя, определите тип MAC-адресов.

Сначала ICMP:

No.	Time	Source	Destination	Protocol	Length	Info
470	30.908321	ICMP	74	Echo (ping) request id=0..
471	30.922073	ICMP	74	Echo (ping) reply id=0..
486	31.926167	ICMP	74	Echo (ping) request id=0..
487	31.948930	ICMP	74	Echo (ping) reply id=0..
489	32.944333	ICMP	74	Echo (ping) request id=0..
490	32.958606	ICMP	74	Echo (ping) reply id=0..
496	33.969521	ICMP	74	Echo (ping) request id=0..
497	33.982662	ICMP	74	Echo (ping) reply id=0..

```

> Frame 470: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface \Device\NPF_{2EB515
> Ethernet II, Src: LiteonTe_dd:ee:8b (74:4c:a1:dd:ee:8b), Dst: TaicangT_ba:b7:40 (e8:d0:b9:ba:b7:40)
  > Destination: TaicangT_ba:b7:40 (e8:d0:b9:ba:b7:40)
  > Source: LiteonTe_dd:ee:8b (74:4c:a1:dd:ee:8b)
    Type: IPv4 (0x0800)
  > Internet Protocol Version 4, Src: 1... 6, Dst: 1... 14
  > Internet Control Message Protocol

```

рис.14. ICMP — эхо-запрос

- длина кадра: 74 bytes (592 bits)
- тип Ethernet : Ethernet II

- MAC-адрес источника :74:4c:a1:dd:ee:8b
- MAC-адрес шлюза: e8:d0:b9:ba:b7:40
- тип MAC-адресов:IPv4 (0.0800)

Wireshark screenshot showing ICMP traffic. The list pane shows frames 470 through 497. The details pane shows a selected ICMP frame (Frame 471) with the following structure:

```

> Frame 471: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface \Device\NPF_{2EB515}
  ✓ Ethernet II, Src: TaicangT_ba:b7:40 (e8:d0:b9:ba:b7:40), Dst: LiteonTe_dd:ee:8b (74:4c:a1:dd:ee:8b)
    > Destination: LiteonTe_dd:ee:8b (74:4c:a1:dd:ee:8b)
    > Source: TaicangT_ba:b7:40 (e8:d0:b9:ba:b7:40)
      Type: IPv4 (0x0800)
  > Internet Protocol Version 4, Src: ..., Dst: ...
  > Internet Control Message Protocol
  
```

рис.15. ICMP — эхо-ответ

- длина кадра: 74 bytes (592 bits)
- тип Ethernet : Ethernet II
- MAC-адрес источника :e8:d0:b9:ba:b7:40
- MAC-адрес шлюза: 74:4c:a1:dd:ee:8b
- тип MAC-адресов:IPv4 (0.0800)

ARP:

Wireshark screenshot showing ARP traffic. The list pane shows frames 9 through 137. The details pane shows a selected ARP frame (Frame 87) with the following structure:

```

> Frame 87: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface \Device\NPF_{2EB5150}
  ✓ Ethernet II, Src: LiteonTe_dd:ee:8b (74:4c:a1:dd:ee:8b), Dst: TaicangT_ba:b7:40 (e8:d0:b9:ba:b7:40)
    > Destination: TaicangT_ba:b7:40 (e8:d0:b9:ba:b7:40)
    > Source: LiteonTe_dd:ee:8b (74:4c:a1:dd:ee:8b)
      Type: ARP (0x0806)
  > Address Resolution Protocol (reply)
  
```

рис.16. ARP — эхо-запрос

Тип Ethernet II

- MAC-адрес источника:74:4c:a1:dd:ee:8b
- MAC-адрес шлюза:e8:d0:b9:ba:b7:40
- Тип:ARP (0x0806)

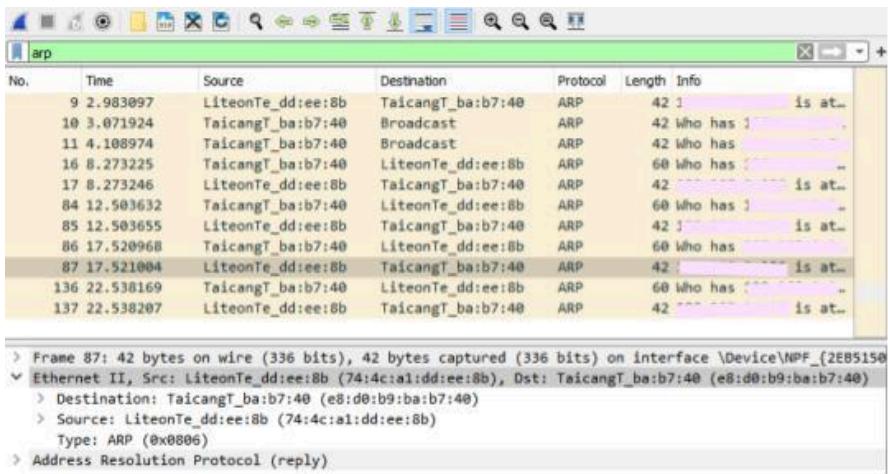


рис.17. ICMP — эхо-ответ

Тип Ethernet II

- MAC-адрес источника:e8:d0:b9:ba:b7:40
- MAC-адрес шлюза: 74:4c:a1:dd:ee:8b
- Тип:ARP (0x0806)

3.3.3. Анализ протоколов транспортного уровня в Wireshark

3.3.3.1. Постановка задачи

С помощью Wireshark захватить и проанализировать пакеты HTTP,DNS в части заголовков и информации протоколов TCP, UDP, QUIC.

3.3.3.2. Порядок выполнения работы

1. Запустите Wireshark. Выберите активный на вашем устройстве сетевой интерфейс. Убедитесь, что начался процесс захвата трафика.

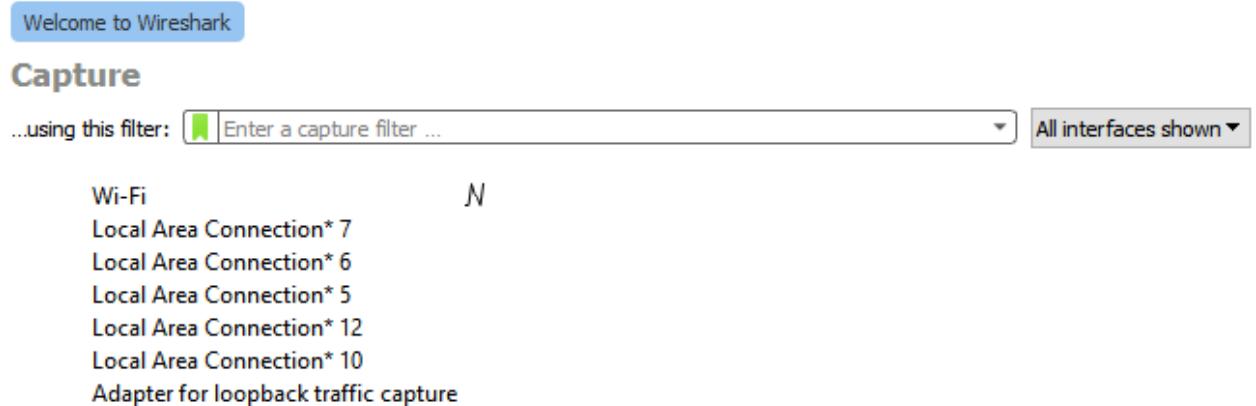


рис.18.активный устройстве сетевой интерфейс wi-fi

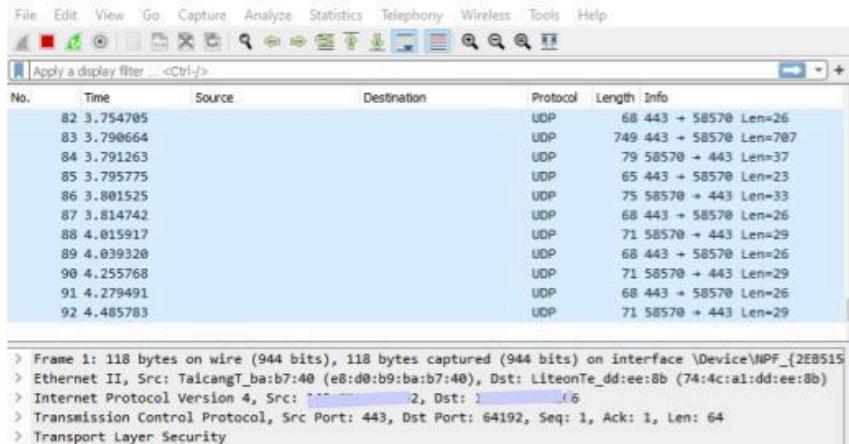
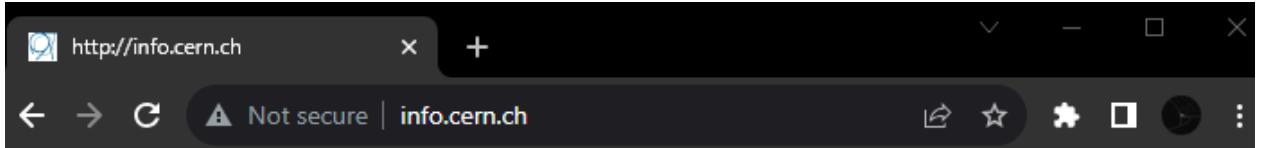


рис.19.процесс начал захватывать трафик.

2. На вашем устройстве в браузере перейдите на сайт, работающий по протоколу HTTP (например, на сайт CERN <http://info.cern.ch/>). При необходимости получения большей информации для Wireshark поперемещайтесь по ссылкам или разделам сайта в браузере.



http://info.cern.ch - home of the first website

From here you can:

- [Browse the first website](#)
- [Browse the first website using the line-mode browser simulator](#)
- [Learn about the birth of the web](#)
- [Learn about CERN, the physics laboratory where the web was born](#)

рис.20.сайт CERN <http://info.cern.ch/>

3. В Wireshark в строке фильтра укажите http и проанализируйте информацию по протоколу TCP в случае запросов и ответов. В отчёте приведите пояснение информации, захваченной в Wireshark.

No.	Time	Source	Destination	Prot
-	249 28.448377	96	32	HT
-	264 28.499463	---	---	HT
	288 29.621108		2	HT
	299 29.668851			HT
	308 33.155354			HT
	382 33.184728			OC
>	Frame 249: 511 bytes on wire (4088 bits), 511 bytes captured (4088 bits)			
>	Ethernet II, Src: LiteonTe_dd:ee:Bb (74:4c:a1:dd:ee:Bb), Dst: Ta			
>	Internet Protocol Version 4, Src: 1.6.2.1, Dst: 1.2.3.1			
>▼	Transmission Control Protocol, Src Port: 64399, Dst Port: 80, Seq: 1, Ack: 1			
	Source Port: 64399			
	Destination Port: 80			
	[Stream index: 5]			
	[Conversation completeness: Complete, WITH_DATA (31)]			
	[TCP Segment Len: 457]			
	Sequence Number: 1 (relative sequence number)			
	Sequence Number (raw): 1310621896			
	[Next Sequence Number: 458 (relative sequence number)]			
	Acknowledgment Number: 1 (relative ack number)			
	Acknowledgment number (raw): 1875559922			
	0x81 - Header Length: 20 bytes (5)			
>	Flags: 0x010 (PSH, ACK)			
	Window: 512			
	[Calculated window size: 131072]			
	[Window size scaling factor: 256]			
	Checksum: 0x30dc [unverified]			
	[Checksum Status: Unverified]			
	Urgent Pointer: 0			
>	[Timestamps]			
>	[SEQ/ACK analysis]			
	TCP payload (457 bytes)			
>▼	Expect Transfer Protocol			

рис.21. HTTP— эхо-запрос

No.	Time	Source	Destination	Protocol	Len
-	249 28.448377	96	32	HTTP	5
-	264 28.499463	32	96	HTTP	5
	288 29.621108			HTTP	5
>	Frame 264: 932 bytes on wire (7456 bits), 932 bytes captured (7456 bits)				
>	Ethernet II, Src: TaicangT_bach7:40 (e8:d0:b0:ba:b7:40), Dst: LiteonTe_dd				
>	Internet Protocol Version 4, Src: 1.6.2.1, Dst: 1.2.3.1				
>▼	Transmission Control Protocol, Src Port: 80, Dst Port: 64399, Seq: 1, Ack: 1				
	Source Port: 80				
	Destination Port: 64399				
	[Stream index: 5]				
	[Conversation completeness: Complete, WITH_DATA (31)]				
	[TCP Segment Len: 878]				
	Sequence Number: 1 (relative sequence number)				
	Sequence Number (raw): 1875559922				
	[Next Sequence Number: 879 (relative sequence number)]				
	Acknowledgment Number: 458 (relative ack number)				
	Acknowledgment number (raw): 1310622353				
	0x81 - Header Length: 20 bytes (5)				
>	Flags: 0x010 (PSH, ACK)				
	Window: 237				
	[Calculated window size: 30396]				
	[Window size scaling factor: 128]				
	Checksum: 0x579d [unverified]				
	[Checksum Status: Unverified]				
	Urgent Pointer: 0				
>	[Timestamps]				
>▼	[SEQ/ACK analysis]				
	[RTT: 0.055070000 seconds]				
	[Bytes in flight: 878]				
	[Bytes sent since last PSH flag: 878]				
	TCP payload (878 bytes)				
>▼	Expect Transfer Protocol				
>	Line-based text data: text/html (13 lines)				

рис.22. HTTP— эхо-ответ

- Source and Destination Ports:
 - TCP использует номера портов для идентификации исходного и конечного приложений или служб.
 - В HTTP-запросе исходным портом обычно является случайно присвоенный эфемерный номер порта на вашем компьютере.
 - Порт назначения для HTTP обычно является портом 80 для обычного HTTP или портом 443 для HTTPS (безопасный HTTP).
- Sequence and Acknowledgment Numbers:

- TCP использует порядковые номера для отслеживания порядка следования пакетов данных.
- Порядковый номер указывает позицию первого байта в сегменте данных.
- Номер подтверждения подтверждает получение данных вплоть до этого порядкового номера.
- Эти цифры помогают обеспечить надежную доставку данных.

- Flags:

TCP-пакеты включают в себя различные флаги для управления передачей данных.
Наши флаги - это:

- ACK (подтверждение): Указывает, что пакет подтверждает полученные данные.
- PSH (Push): подает сигнал получателю о немедленной доставке данных в приложение.

- Window:

- Размер окна - это важное значение, которое указывает на размер принимающего буфера.
- Это помогает регулировать поток данных и обеспечивает эффективную передачу данных.
- Большой размер окна может обеспечить более быструю передачу данных.

- Acknowledgment of Data:

- Поскольку данные отправляются от клиента к серверу в HTTP-запросе, вы увидите подтверждение данных в соответствующих пакетах подтверждения TCP (ACK).
- Номер подтверждения в пакете ACK указывает следующий ожидаемый порядковый номер от сервера.

Изучая эти сведения о протоколе TCP в Wireshark, вы можете получить представление об установлении TCP-соединения, потоке данных и подтверждении данных между клиентом и веб-сервером во время HTTP-запроса. Эта информация необходима для понимания надежности и эффективности передачи данных по сети.

4. Wireshark в строке фильтра укажите dns и проанализируйте информацию по протоколу UDP в случае запросов и ответов. В отчёте приведите пояснение по информации, захваченной в Wireshark.

239 28.386308	6	4	DNS	95 Standard query 0x4912 HTTPS optimizationguide-pa.googleapis.com
243 28.395235			DNS	351 Standard query response 0xd143 A optimizationguide-pa.googleapis.com
244 28.395329			DNS	155 Standard query response 0x4912 HTTPS optimizationguide-pa.googleapis.com
> Frame 239: 95 bytes on wire (760 bits), 95 bytes captured (760 bits) on interface '\Device\NPF_{2EB51507-B802-4C69-BCA7-B8B308867E0B}', id 0				
> Ethernet II, Src: LiteonTe_dd:ee:8b (74:4c:a1:dd:ee:8b), Dst: TaicangT_ba:b7:40 (e8:d0:b9:ba:b7:40)				
> Internet Protocol Version 4, Src: 192.168.1.36, Dst: 192.168.1.4				
User Datagram Protocol, Src Port: 57711, Dst Port: 53				
Source Port: 57711				
Destination Port: 53				
Length: 61				
Checksum: 0x92b2 [unverified]				
[Checksum Status: Unverified]				
[Stream index: 16]				
> [Timestamps]				
UDP payload (53 bytes)				
> Domain Name System (query)				

рис.23. dns— эхо-запрос

239 28.386308	1	4	DNS	95 Standard query 0x4912 HTTPS optimizationguide-pa.googleapis.com
243 28.395235			DNS	351 Standard query response 0xd143 A optimizationguide-pa.googleapis.com A 142.251.1.1
244 28.395329			DNS	155 Standard query response 0x4912 HTTPS optimizationguide-pa.googleapis.com SOA ns1
> Frame 244: 155 bytes on wire (1240 bits), 155 bytes captured (1240 bits) on interface '\Device\NPF_{2EB51507-B802-4C69-BCA7-B8B308867E0B}', id 0				
> Ethernet II, Src: TaicangT_ba:b7:40 (e8:d0:b9:ba:b7:40), Dst: LiteonTe_dd:ee:8b (74:4c:a1:dd:ee:8b)				
> Internet Protocol Version 4, Src: 192.168.1.4, Dst: 192.168.1.6				
User Datagram Protocol, Src Port: 53, Dst Port: 57711				
Source Port: 53				
Destination Port: 57711				
Length: 121				
Checksum: 0xa3c7 [unverified]				
[Checksum Status: Unverified]				
[Stream index: 16]				
> [Timestamps]				
UDP payload (113 bytes)				
> Domain Name System (response)				

рис.24. dns— эхо-ответ

Запрос DNS:

- User Datagram Protocol (UDP): Это указывает на то, что пакет использует транспортный протокол UDP.
- Src Port (Source Port): номер порта источника, с которого отправляется DNS-запрос. В данном случае это порт 57711.
- Dst Port (Destination Port): Номер порта назначения, на который отправляется DNS-запрос. Порт 53 является стандартным портом для DNS.
- Length: общая длина UDP-пакета, включая UDP-заголовок и полезную нагрузку. В данном случае это 61 байт.
- Checksum: Поле контрольной суммы используется для проверки ошибок. Он в шестнадцатеричном формате (0x92b2) и помечен как "непроверенный". Контрольные суммы UDP необязательны, и в данном случае они не проверяются.
- Stream index: Этот индекс помогает Wireshark организовывать и отображать связанные пакеты. В данном случае он помечен как stream index 16.
- Timestamps: Это указывает на то, что в пакет включены временные метки, которые могут использоваться для различных целей, включая измерение времени в пути туда и обратно.
- UDP payload (53 bytes): Фактические данные, переносимые пакетом UDP. В DNS-запросах эта полезная нагрузка обычно включает в себя DNS-запрос, включая запрашиваемое доменное имя.

Ответ DNS:

User Datagram Protocol (UDP):: Как и в случае с запросом, это указывает на использование транспортного протокола UDP.

- Src Port (Source Port): номер порта источника для ответа DNS. В данном случае

это стандартный DNS-порт, 53.

Dst Port (Destination Port): Номер порта назначения, на который отправляется ответ DNS. Порт 57711 совпадает с исходным портом исходного запроса.

- Length: общая длина UDP-пакета, включая UDP-заголовок и полезную нагрузку. В данном случае это 121 байт.

- Checksum: Поле контрольной суммы, снова помеченное как "непроверенное". Контрольные суммы UDP являются необязательными.

- Stream index: тот же индекс потока, что и у запроса DNS (индекс потока 16), указывающий, что этот ответ связан с запросом.

- Timestamps: Временные метки также включены в пакет.

- UDP payload (53 bytes): Фактические данные, переносимые пакетом UDP. В ответах DNS это обычно включает разрешенный IP-адрес (или другие записи DNS), соответствующий доменному имени из исходного запроса.

Таким образом, эти сведения о захвате Wireshark показывают обмен пакетами запросов DNS и ответов. Запрос DNS инициируется с исходного порта (57711) на порт DNS-сервера (53). Ответ DNS возвращается с сервера на исходный порт (57711) с разрешенной информацией. Для этого обмена используется протокол UDP, и хотя контрольные суммы присутствуют, в данном случае они непроверены. Временные метки предоставляют дополнительный контекст для измерения информации, связанной со временем, такой как время в пути туда и обратно.

5. Wireshark в строке фильтра укажите `quic` и проанализируйте информацию по протоколу `quic` в случае запросов и ответов. В отчёте приведите пояснение по информации, захваченной в Wireshark.

1353 27.234546	16	16	QUIC	1292 Initial, DCID=fe63954939
1354 27.251207			QUIC	1292 Initial, SCID=fe63954939
1355 27.257457			QUIC	1292 Initial, DCID=fe63954939
1356 27.284436			QUIC	1292 Handshake, SCID=fe63954939
1357 27.284869			QUIC	1292 Handshake, SCID=fe63954939
1358 27.284869			QUIC	1292 Handshake, SCID=fe63954939
1359 27.285039			QUIC	81 Handshake, DCID=fe63954939
1360 27.285226			QUIC	884 Protected Payload (KP0)

> Frame 1353: 1292 bytes on wire (10336 bits), 1292 bytes captured (10336 bits) on interface \Device\

> Ethernet II, Src: LiteonTe_dd:ee:8b (74:4c:a1:dd:ee:8b), Dst: TaicangT_ba:b7:40 (e8:d0:b9:ba:b7:40)

> Internet Protocol Version 4, Src: 192.168.1.100, Dst: 192.168.1.101

> User Datagram Protocol, Src Port: 62540, Dst Port: 443

Source Port: 62540
Destination Port: 443
Length: 1258
Checksum: 0x7b88 [unverified]
[Checksum Status: Unverified]
[Stream index: 64]
> [Timestamps]
UDP payload (1250 bytes)
> QUIC IETF

рис.25. `quic`— эхо-запрос

1980	38.645032	6	..106	QUIC	66 Protected Payload (KP0)
2003	41.088125			QUIC	144 Protected Payload (KP0), DCID=
2004	41.134847			QUIC	69 Protected Payload (KP0)
2007	41.161963		'	QUIC	74 Protected Payload (KP0), DCID=
2008	41.176116			QUIC	181 Protected Payload (KP0)
2009	41.176292		3	QUIC	64 Protected Payload (KP0)
2010	41.176907		7	QUIC	77 Protected Payload (KP0), DCID=
2011	41.202857			QUIC	74 Protected Payload (KP0), DCID=
2012	41.246928			QUIC	66 Protected Payload (KP0)
2024	44.115753			QUIC	1292 Initial, DCID=4015a07849d47a71
> Frame 1980: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface '\Device\NPF_{2E851507-B					
> Ethernet II, Src: TaicangT_ba:b7:40 (e8:d8:b9:ba:b7:40), Dst: LiteonTe_dd:ee:8b (74:4c:a1:dd:ee:8b)					
> Internet Protocol Version 4, Src: ..., Dst: ..., 6, Dst: ..., 86					
User Datagram Protocol, Src Port: 443, Dst Port: 62540					
Source Port: 443					
Destination Port: 62540					
Length: 32					
Checksum: 0x12fd [unverified]					
[Checksum Status: Unverified]					
[Stream index: 64]					
[Timestamps]					
UDP payload (24 bytes)					
> QUIC IETF					

рис.26. quic— эхо-ответ

Давайте разберем их подробно:

Запрос (Request):

- Протокол: User Datagram Protocol (UDP)
- Исходный порт (Src Port): 62540
- Порт назначения (Dst Port): 443
- Длина (Length): 1258 байт
- Контрольная сумма (Checksum): 0x7b88 [непроверенная]
- Статус контрольной суммы: Непроверенная
- Индекс потока: 64
- Метка времени (Timestamps)
- Полезная нагрузка UDP (UDP payload): 1250 байт

Ответ (Response):

- Протокол: User Datagram Protocol (UDP)
- Исходный порт (Src Port): 443
- Порт назначения (Dst Port): 62540
- Длина (Length): 32 байта
- Контрольная сумма (Checksum): 0x12fd [непроверенная]
- Статус контрольной суммы: Непроверенная
- Индекс потока: 64
- Метка времени (Timestamps)
- Полезная нагрузка UDP (UDP payload): 24 байта

Теперь давайте объясним, что означает каждое из этих полей:

- Протокол (Protocol): Указывает на то, что это UDP-пакет.

- Исходный порт (Source Port): Порт отправителя (вашего компьютера) для идентификации отправителя.
- Порт назначения (Destination Port): Порт получателя (какой-либо удаленный сервер или служба), куда направлен запрос или ответ.
- Длина (Length): Общая длина UDP-пакета в байтах, включая заголовок и данные.
- Контрольная сумма (Checksum): Значение контрольной суммы, которое помогает обнаруживать ошибки в данных (но в данном случае она отмечена как непроверенная).
- Статус контрольной суммы (Checksum Status): Показывает, что контрольная сумма пакета не была проверена.
- Индекс потока (Stream Index): Индекс потока, связанный с этим пакетом.
- Метка времени (Timestamps): Метка времени может использоваться для отслеживания времени отправки и приема пакетов.
- Полезная нагрузка UDP (UDP payload): Это собственно данные, передаваемые внутри UDP-пакета. Длина полезной нагрузки указана в байтах.

Запрос был отправлен с порта 62540 на порт 443, а ответ был получен с порта 443 на порт 62540. Ответ имеет гораздо меньшую длину по сравнению с запросом.

6. Остановите захват трафика в Wireshark

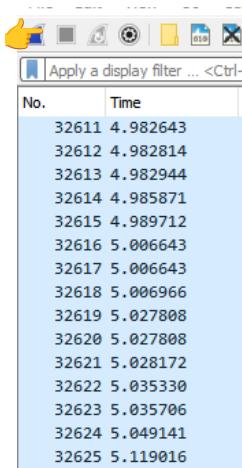


рис.26. Остановка захвата трафика в Wireshark(нет красный свет (его же знак)

3.3.4. Анализ handshake протокола TCP в Wireshark

3.3.4.1. Постановка задачи

С помощью Wireshark проанализировать handshake протокола TCP.

3.3.4.2. Порядок выполнения работы

1. Запустите Wireshark. Выберите активный на вашем устройстве сетевой интерфейс. Убедитесь, что начался процесс захвата трафика.

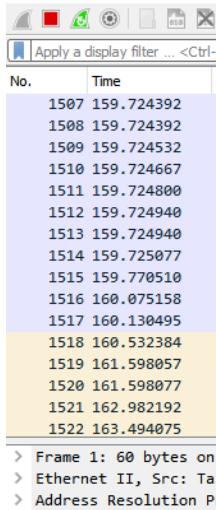


рис.27.Начался процесс перехвата трафика

2. На вашем устройстве или используйте подсоединение по telnet или ssh к вашему маршрутизатору (например с помощью PUTTY или соответствующих команд в консоли), или соединение по HTTP с каким-то сайтом для захвата в Wireshark пакетов TCP.

```
'telnet' is not recognized as an internal or external command,  
operable program or batch file.
```

3. В Wireshark проанализируйте handshake протокола TCP, в отчёте приведите пример с пояснениями изменения значений соответствующих сообщений при установлении соединения по TCP.

```
▼ Transmission Control Protocol, Src Port: 49767, Dst Port: 5228, Seq: 1, Ack: 1, Len: 1  
  Source Port: 49767  
  Destination Port: 5228  
  [Stream index: 0]  
  [Conversation completeness: Incomplete (12)]  
  [TCP Segment Len: 1]  
  Sequence Number: 1 (relative sequence number)  
  Sequence Number (raw): 2704850857  
  [Next Sequence Number: 2 (relative sequence number)]  
  Acknowledgment Number: 1 (relative ack number)  
  Acknowledgment number (raw): 2587352805  
  0101 .... = Header Length: 20 bytes (5)  
  > Flags: 0x010 (ACK)  
  Window: 508  
  [Calculated window size: 508]  
  [Window size scaling factor: -1 (unknown)]  
  Checksum: 0x8c3a (unverified)
```

рис.28. tcp-эхо-запрос

```
* Transmission Control Protocol, Src Port: 5228, Dst Port: 49767, Seq: 27, Ack: 28, Len: 0
  Source Port: 5228
  Destination Port: 49767
  [Stream index: 0]
  [Conversation completeness: Incomplete (12)]
  [TCP Segment Len: 0]
  Sequence Number: 27      (relative sequence number)
  Sequence Number (raw): 2587352831
  [Next Sequence Number: 27      (relative sequence number)]
  Acknowledgment Number: 28      (relative ack number)
  Acknowledgment number (raw): 2704850884
  1000 .... = Header Length: 32 bytes (8)
  > Flags: 0x010 (ACK)
  Window: 265
```

рис.29. tcp-эхо-ответ

При анализе подтверждения по протоколу TCP в Wireshark вы обычно обращаете внимание на процесс трехстороннего подтверждения, который происходит при установлении TCP-соединения. Тремя этапами рукопожатия являются:

SYN (синхронизировать) - Инициирующее устройство отправляет TCP-пакет с установленным флагом SYN на сервер для запроса соединения.

SYN-ACK (Синхронизировать-подтверждение) - Сервер отвечает TCP-пакетом, содержащим флаги SYN и ACK, установленные для подтверждения запроса и инициирования соединения.

ACK (подтверждение) - Инициирующее устройство отправляет окончательный TCP-пакет с установленным флагом ACK для подтверждения ответа сервера, завершая настройку соединения.

Подтверждение (от клиента к серверу):

Порт источника: тот же порт источника, что и раньше.

Порт назначения: Тот же порт назначения, что и раньше.

Порядковый номер: увеличивается на 1 по сравнению с предыдущим порядковым номером клиента.

Номер подтверждения: увеличивается на 1 по сравнению с предыдущим номером подтверждения сервера.

Флаги: Флаг SYN не установлен, флаг подтверждения установлен.

4. В Wireshark в меню «Статистика» выберете «График Потока». В отчёте приведите пояснения по изменениям значений соответствующих сообщений при установлении соединения по TCP.

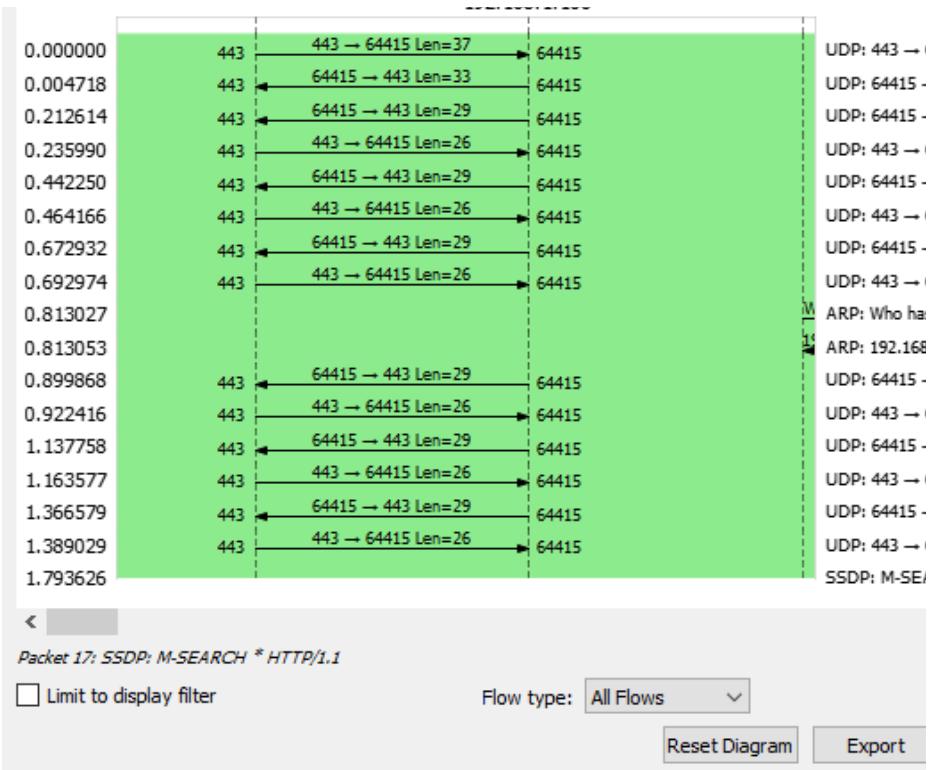


рис.30.График Потока

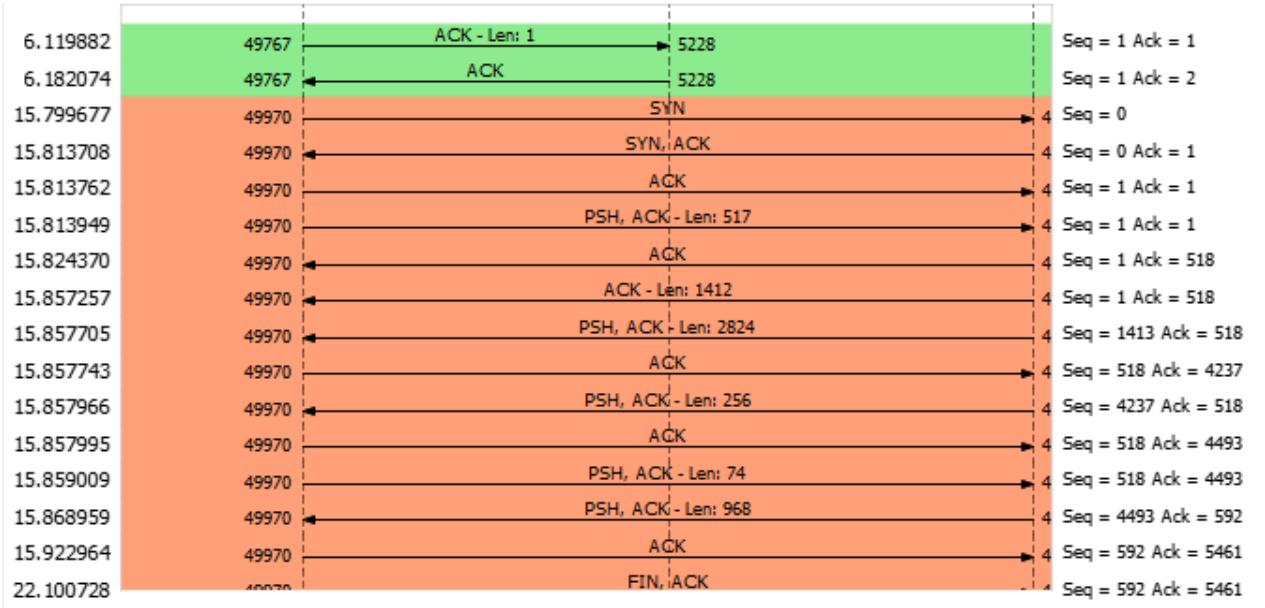


рис.31.TCP flows

Трехстороннее рукопожатие TCP включает в себя три основных пакета: SYN, SYN-ACK и ACK-подтверждение. Давайте разберем изменения в их значениях:

Пакет SYN (синхронизировать) (клиент-сервер):

Порт источника: Случайно выбранный временный порт на стороне клиента.

Порт назначения: Хорошо известный порт, связанный со службой на сервере (например, 80 для HTTP, 22 для SSH).

Порядковый номер: Случайное число, выбранное клиентом, часто называемое

начальным порядковым номером (ISN).

Номер подтверждения: 0 (поскольку это начальный пакет и подтверждать пока нечего).

Флаги: установлен флаг SYN (указывающий на начало соединения), флаг подтверждения не установлен (поскольку данные еще не получены).

SYN-ACK (Синхронизация-подтверждение) Пакет (от сервера к клиенту):

Порт источника: Хорошо известный порт, связанный со службой на сервере.

Порт назначения: исходный порт из пакета SYN (временный порт на стороне клиента).

Порядковый номер: случайное число, выбранное сервером (ISN сервера).

Номер подтверждения: Номеру подтверждения присваивается порядковый номер клиента + 1. Это подтверждает получение SYN-пакета от клиента.

Флаги: установлен флаг SYN (указывает на продолжение инициализации соединения), установлен флаг ACK (подтверждает SYN от клиента).

ACK (Подтверждение) Пакет (от клиента к серверу):

Порт источника: тот же самый эфемерный порт на стороне клиента.

Порт назначения: Тот же самый хорошо известный порт на стороне сервера.

Порядковый номер: увеличивается на 1 по сравнению с предыдущим порядковым номером клиента. Это подтверждает получение пакета SYN-ACK от сервера.

Номер подтверждения: увеличивается на 1 по сравнению с предыдущим номером подтверждения сервера. Это подтверждает получение SYN-ACK от сервера.

Флаги: Флаг SYN не установлен (поскольку настройка соединения завершена), установлен флаг ACK (подтверждение SYN-ACK с сервера).

Таким образом, ключевые изменения во время трехстороннего подтверждения связи TCP включают в себя последовательность пакетов и установку флагов:

Пакет SYN инициирует соединение и содержит начальный порядковый номер клиента.

Пакет SYN-ACK подтверждает SYN клиента, содержит начальный порядковый номер сервера и подтверждает порядковый номер клиента.

Пакет ACK подтверждает SYN-ACK от сервера, увеличивая порядковые номера как клиента, так и сервера на 1.

Эти шаги гарантируют, что обе стороны соединения синхронизированы и знают начальные порядковые номера друг друга, что обеспечивает надежную передачу данных.

5. Остановите захват трафика в Wireshark.



рис.32.захват трафика в Wireshark

3.3.4. Анализ handshake протокола TCP в Wireshark

3.3.4.1. Постановка задачи

С помощью Wireshark проанализировать handshake протокола TCP.

Анализ "handshake" протокола TCP с использованием Wireshark - это отличный способ изучить как работает установление соединения между двумя устройствами. Протокол TCP используется для надежной передачи данных в сети. Handshake TCP состоит из трех этапов: установление соединения, передача данных и завершение соединения. Давайте разберем каждый этап на примере с помощью Wireshark:

1. Установление соединения (TCP Three-Way Handshake):

- Когда клиент хочет установить соединение с сервером, он отправляет TCP пакет с флагом SYN (Synchronize) и начальным номером последовательности (ISN - Initial Sequence Number).
 - Когда сервер получает этот пакет, он отправляет ответный пакет с флагами SYN и ACK (Acknowledgment). В этом пакете сервер устанавливает свой ISN.
 - Клиент, получив ответ от сервера, отправляет третий пакет с флагом ACK. Этот пакет подтверждает установление соединения.

В Wireshark вы можете найти эти пакеты, используя фильтр `tcp.flags.syn == 1 && tcp.flags.ack == 0` для поиска пакета SYN от клиента и `tcp.flags.syn == 1 && tcp.flags.ack == 1` для поиска пакета SYN-ACK от сервера.

2. Передача данных:

- После успешного установления соединения клиент и сервер могут обмениваться данными, отправляя TCP пакеты с флагом ACK.

Для отображения данных, передаваемых между клиентом и сервером, вы можете просто просматривать TCP пакеты в Wireshark и анализировать их содержимое.

3. Завершение соединения (TCP Four-Way Handshake):

- Когда клиент или сервер хочет завершить соединение, они отправляют TCP пакет с флагом FIN (Finish).
 - Завершение соединения состоит из двух этапов: сначала одна сторона отправляет FIN, а затем другая подтверждает завершение соединения.

Для анализа завершения соединения, вы можете использовать фильтры `tcp.flags.fin == 1 && tcp.flags.ack == 0` и `tcp.flags.fin == 1 && tcp.flags.ack == 1` для поиска соответствующих пакетов.

Обратите внимание, что в Wireshark вы можете также просматривать дополнительные сведения о каждом пакете, такие как порты, IP-адреса, длина данных и т. д., что помогает в более подробном анализе handshake и обмена данными в TCP соединении.

Выводы

В этой лабораторной работе я изучил фреймы Ethernet с помощью Wireshark и научился анализировать протоколы PDU транспортного и прикладного уровней стека TCP/IP.