

Отчет по лабораторной работе №3

Дисциплина: Сетевые технологии

Иванов Сергей Владимирович

Содержание

1	Цель работы	4
2	Выполнение лабораторной работы	5
2.1	MAC-адресация	5
2.2	Анализ кадров канального уровня в Wireshark	10
2.3	Анализ протоколов транспортного уровня в Wireshark	17
2.4	Анализ handshake протокола TCP в Wireshark	24
3	Выводы	28

Список иллюстраций

2.1	Команда <code>ipconfig</code>	7
2.2	Команда <code>ipconfig /all</code>	9
2.3	Команда <code>ipconfig /all</code>	9
2.4	MAC-адрес	10
2.5	Захват трафика	11
2.6	Определение IP и шлюза	11
2.7	<code>ping</code> шлюза	11
2.8	фильтр <code>arp or icmp</code>	12
2.9	Эхо запрос	12
2.10	Эхо ответ	13
2.11	Протокол ARP	14
2.12	Пинг <code>ya.ru</code>	14
2.13	анализ ARP протокола	15
2.14	ICMP эхо-запрос	16
2.15	ICMP эхо-ответ	17
2.16	Захват трафика	17
2.17	Переход на сайт <code>cern.ch</code>	18
2.18	Анализ TCP (запрос)	19
2.19	Анализ TCP (ответ)	20
2.20	Анализ UDP (запрос)	21
2.21	Анализ UDP (ответ)	22
2.22	Анализ QUIC (запрос)	23
2.23	Анализ QUIC (ответ)	24
2.24	Захват трафика	24
2.25	TCP handshake шаг 1	25
2.26	TCP handshake шаг 2	26
2.27	TCP handshake шаг 3	27
2.28	График потока	27

1 Цель работы

Целью этой лабораторной работы является изучение посредством Wireshark кадров Ethernet, анализ PDU протоколов транспортного и прикладного уровней стека TCP/IP.

2 Выполнение лабораторной работы

2.1 MAC-адресация

С помощью команды `ipconfig` для ОС типа Windows выведем информацию о текущем сетевом соединении. Вывод команды показывает информацию о сетевых адаптерах. Некоторые из них активны, другие - отключены.

1. Адаптер Ethernet Ethernet 3. Это Ethernet-адаптер (проводное соединение). Это виртуальный адаптер, созданный для локальной сети, в виртуальной машине (VirtualBox). Он активен и имеет конфигурацию.

DNS-суффикс подключения: отсутствует. DNS-суффикс — это доменное имя, которое автоматически добавляется к неполным именам хостов при разрешении DNS. Если поле пустое, суффикс не задан, и система будет использовать только полные доменные имена или другие глобальные DNS-настройки.

Локальный IPv6-адрес канала: `fe80::926a:5553:d565:4c13%8`. Это link-local IPv6-адрес (адрес локальной связи). IPv6-адреса типа `fe80::` используются для коммуникации только внутри локального сегмента сети (без маршрутизации через роутеры). Суффикс “%8” — это идентификатор интерфейса, указывающий, к какому адаптеру привязан адрес. Этот адрес генерируется автоматически на основе MAC-адреса устройства и не требует DHCP-сервера.

IPv4-адрес: `192.168.56.1`. Это основной IPv4-адрес адаптера. `192.168.56.1` — приватный IP-адрес из диапазона `192.168.0.0/16`, предназначенный для локальных сетей.

Маска подсети: 255.255.255.0. Маска подсети определяет размер сети и разделяет IP-адрес на сетевую и хостовую части. 255.255.255.0 соответствует префиксу /24, то есть в сети может быть до 254 хостов.

Основной шлюз: (пусто). Шлюз по умолчанию — это IP-адрес роутера, через который трафик направляется за пределы локальной сети. Пустое поле означает, что шлюз не настроен, поэтому этот адаптер может общаться только с устройствами в той же подсети. Доступ к интернету через этот адаптер невозможен.

2. Адаптер беспроводной локальной сети Подключение по локальной сети*

9. Это беспроводной (Wi-Fi) адаптер. Звёздочка указывает на виртуальный или вспомогательный адаптер, созданный системой.

Состояние среды: Среда передачи недоступна. Этот адаптер неактивен.

3. Адаптер беспроводной локальной сети Подключение по локальной сети*

10. Аналогично предыдущему — ещё один беспроводной адаптер. Этот адаптер тоже неактивен.

4. Адаптер Ethernet outline-tap0. Это Ethernet-адаптер с названием “outline-tap0”. Название говорит, что это виртуальный TAP-адаптер, созданный программой VPN (Outline VPN).

Состояние среды: Среда передачи недоступна. Адаптер отключён, так как VPN выключен. (рис. 1).

```
C:\Windows\System32>
C:\Windows\System32>ipconfig

Настройка протокола IP для Windows

Адаптер Ethernet Ethernet 3:

    DNS-суффикс подключения . . . . . :
    Локальный IPv6-адрес канала . . . . : fe80::926a:5553:d565:4c13%8
    IPv4-адрес. . . . . : 192.168.56.1
    Маска подсети . . . . . : 255.255.255.0
    Основной шлюз. . . . . :

Адаптер беспроводной локальной сети Подключение по локальной сети* 9:

    Состояние среды. . . . . : Среда передачи недоступна.
    DNS-суффикс подключения . . . . . :

Адаптер беспроводной локальной сети Подключение по локальной сети* 10:

    Состояние среды. . . . . : Среда передачи недоступна.
    DNS-суффикс подключения . . . . . :

Адаптер Ethernet outline-tap0:

    Состояние среды. . . . . : Среда передачи недоступна.
    DNS-суффикс подключения . . . . . :
```

Рис. 2.1: Команда ipconfig

Теперь используем опцию /all команды ipconfig. Команда предоставляет расширенную информацию о конфигурации сетевых адаптеров. Разберу подробно Беспроводное соединение.

DNS-суффикс подключения: (пусто). DNS-суффикс — это доменное имя, добавляемое автоматически к запросам DNS для разрешения неполных имён. Отсутствие суффикса указывает, что система не настроена на использование определённого домена по умолчанию для этого подключения. Это может быть связано с динамической настройкой через DHCP.

Описание: MediaTek Wi-Fi 6 MT7921 Wireless LAN Card. Это беспроводной адаптер, поддерживающий Wi-Fi 6.

Физический адрес: 90-E8-68-FA-55-E3. MAC-адрес — уникальный идентификатор сетевого интерфейса, назначенный производителем

DHCP включён: Да. DHCP включён, IP-адрес и другие параметры сети автоматически выдаются сервером DHCP (например, роутером).

Автонастройка IPv4-адрес канала: Да. Включена автоконфигурация IPv4, что позволяет адаптеру автоматически назначать себе link-local адрес, если DHCP-сервер недоступен. Это резервный механизм.

Локальный IPv6-адрес канала: fe80::2fef:1cdc:5e8:217a%14. Link-local IPv6-

адрес используется для коммуникации внутри локального сегмента сети без маршрутизации.

IPv4-адрес: 172.16.4.1. Это основной IPv4-адрес, назначенный адаптеру. Адрес 172.16.4.1 находится в частном диапазоне 172.16.0.0/12, используемом для локальных сетей. Это может быть адрес, выданный DHCP-сервером роутера или точки доступа.

Маска подсети: 255.255.255.0. Маска подсети /24 определяет, что подсеть содержит до 254 хостов.

Аренда получена: 29 сентября 2025 г. 10:49:21. Дата и время, когда адаптер получил текущую конфигурацию от DHCP-сервера.

Срок аренды истекает: 29 сентября 2025 г. 15:49:10. Время истечения аренды IP-адреса.

Основной шлюз: 172.16.4.59. Шлюз по умолчанию — IP-адрес роутера или точки доступа, через который трафик направляется за пределы локальной сети. Адрес 172.16.4.59 находится в той же подсети, что и адаптер, что логично для внутренней маршрутизации.

DHCP-сервер: 192.168.80.59. IP-адрес сервера DHCP, который выдал конфигурацию.

IAID DHCPv6: 49603594. Идентификатор клиента для DHCPv6, используемый при конфигурации IPv6.

DUID клиента DHCPv6: 00-01-00-01-2D-00-EA-E7-58-11-22-88-F5-69. Уникальный идентификатор клиента DHCPv6, сгенерированный системой для работы с IPv6.

DNS-серверы: 37.18.92.5 193.232.218.194. Два DNS-сервера. Первый может быть публичным сервером, второй — возможно, сервер организации или резервный.

NetBIOS через TCP/IP: Включён. NetBIOS включён, что позволяет использовать старые протоколы именования в локальной сети. (рис. 2, 3).


```

Адаптер беспроводной локальной сети Беспроводная сеть:

DNS-суффикс подключения . . . . . :
Описание. . . . . : MediaTek Wi-Fi 6 MT7921 Wireless LAN Card
Физический адрес. . . . . : 90-E8-68-FA-55-E3
DHCP включен. . . . . : Да
Автонастройка включена. . . . . : Да
Локальный IPv6-адрес канала . . . : fe80::2fef:1cd:c5e8:217a%14(Основной)
IPv4-адрес. . . . . : 172.16.94.216(Основной)
Маска подсети . . . . . : 255.255.254.0
Аренда получена. . . . . : 29 сентября 2025 г. 10:49:21
Срок аренды истекает. . . . . : 29 сентября 2025 г. 15:49:10
Основной шлюз. . . . . : 172.16.94.1
DHCP-сервер. . . . . : 192.168.80.59
IAID DHCPv6 . . . . . : 496035944
DUID клиента DHCPv6 . . . . . : 00-01-00-01-2D-00-EA-E7-58-11-22-88-F5-69
DNS-серверы. . . . . : 37.18.92.5
                        193.232.218.194
NetBios через TCP/IP. . . . . : Включен

Адаптер Ethernet Сетевое подключение Bluetooth:

Состояние среды. . . . . : Среда передачи недоступна.
DNS-суффикс подключения . . . . . :
Описание. . . . . : Bluetooth Device (Personal Area Network)
Физический адрес. . . . . : 90-E8-68-FA-55-E2
DHCP включен. . . . . : Да
Автонастройка включена. . . . . : Да

```

Рис. 2.2: Команда ipconfig /all

```

C:\Windows\System32>ipconfig /all

Настройка протокола IP для Windows

Имя компьютера . . . . . : Rtiop
Основной DNS-суффикс . . . . . :
Тип узла. . . . . : Гибридный
IP-маршрутизация включена . . . . : Нет
WINS-прокси включен . . . . . : Нет

Адаптер Ethernet Ethernet 3:

DNS-суффикс подключения . . . . . :
Описание. . . . . : VirtualBox Host-Only Ethernet Adapter
Физический адрес. . . . . : 0A-00-27-00-00-08
DHCP включен. . . . . : Нет
Автонастройка включена. . . . . : Да
Локальный IPv6-адрес канала . . . : fe80::926a:5553:d565:4c13%8(Основной)
IPv4-адрес. . . . . : 192.168.56.1(Основной)
Маска подсети . . . . . : 255.255.255.0
Основной шлюз. . . . . :
IAID DHCPv6 . . . . . : 722075687
DUID клиента DHCPv6 . . . . . : 00-01-00-01-2D-00-EA-E7-58-11-22-88-F5-69
NetBios через TCP/IP. . . . . : Включен

Адаптер беспроводной локальной сети Подключение по локальной сети* 9:

Состояние среды. . . . . : Среда передачи недоступна.
DNS-суффикс подключения . . . . . :
Описание. . . . . : Microsoft Wi-Fi Direct Virtual Adapter
Физический адрес. . . . . : 92-E8-68-FA-55-A3
DHCP включен. . . . . : Да
Автонастройка включена. . . . . : Да

```

Рис. 2.3: Команда ipconfig /all

Определим MAC-адрес интерфейса Беспроводное соединение (Wi-Fi). Он находится в поле физический адрес (90-E8-68-FA-55-E3). Имеет длину 6 байт и записывается в 16-ти ричном формате. Первые 3 байта: OUI (Organizationally Unique Identifier) — идентификатор организации, который определяет производителя оборудования. Последние 3 байта: NIC Specific — серийный номер, назначаемый производителем.

Согласно базе данных IEEE, OUI 90-E8-68 принадлежит компании AzureTechnolody (MediaTek Inc.). Это совпадает с описанием устройства в выводе команды: “MediaTek Mi-Fi 6 MT7921 Wireless LAN Card”. Проанализируем адрес 90-E8-68-FA-55-E3. Для этого нужно посмотреть на значение первого первого байта адреса в двоичном формате.

Первый байт: 90 (в шестнадцатеричной системе) = 1001 0000 (в двоичной системе).

В этом байте нас интересуют два младших бита:

b0 (самый младший бит): Управляет типом адресации. 0 = Индивидуальный (Unicast) Адрес предназначен для одного конкретного сетевого интерфейса. 1 = Групповой (Multicast) Адрес предназначен для группы интерфейсов.

b1 (второй бит): Управляет способом администрирования. 0 = Глобально уникальный (UAA, Universally Administered Address). Постоянный адрес, прошитый на заводе. 1 = Локально управляемый (LAA, Locally Administered Address). Адрес, переопределенный пользователем или программным обеспечением.

Значит, наш адрес индивидуальный и глобально администрируемый. (рис. 4)

```
Адаптер беспроводной локальной сети Беспроводная сеть:
DNS-суффикс подключения . . . . . :
Описание. . . . . : MediaTek Wi-Fi 6 MT7921 Wireless LAN Card
Физический адрес. . . . . : 90-E8-68-FA-55-E3
```

Рис. 2.4: MAC-адрес

2.2 Анализ кадров канального уровня в Wireshark

Запустим Wireshark. Выберем активный на устройстве сетевой интерфейс и убедимся, что начался процесс захвата трафика. (рис. 5)

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	172.16.94.159	224.0.0.251	DNS	81	Standard query 0x0000 PTR _lyra-mdns_udp.local, "Q" question
2	0.104233	172.16.94.159	224.0.0.251	DNS	119	Standard query 0x03db PTR _googlecast_tcp.local, "Q" question PTR_67440243_sub_googlecast_tcp.local,
3	0.104487	172.16.94.253	224.0.0.251	DNS	119	Standard query 0x03db PTR _googlecast_tcp.local, "Q" question PTR_67440243_sub_googlecast_tcp.local,
4	0.203520	172.16.94.247	224.0.0.251	DNS	119	Standard query 0x0035 PTR _googlecast_tcp.local, "Q" question PTR_67440243_sub_googlecast_tcp.local,
5	0.306526	192.168.0.1	239.255.255.250	SSDP	459	NOTIFY * HTTP/1.1
6	0.307229	192.168.0.1	239.255.255.250	SSDP	468	NOTIFY * HTTP/1.1
7	0.308404	192.168.0.1	239.255.255.250	SSDP	521	NOTIFY * HTTP/1.1
8	0.309420	192.168.0.1	239.255.255.250	SSDP	523	NOTIFY * HTTP/1.1
9	0.312810	192.168.0.1	239.255.255.250	SSDP	468	NOTIFY * HTTP/1.1
10	0.314293	192.168.0.1	239.255.255.250	SSDP	507	NOTIFY * HTTP/1.1
11	0.314735	192.168.0.1	239.255.255.250	SSDP	530	NOTIFY * HTTP/1.1
12	0.320165	192.168.0.1	239.255.255.250	SSDP	468	NOTIFY * HTTP/1.1
13	0.320250	192.168.0.1	239.255.255.250	SSDP	527	NOTIFY * HTTP/1.1
14	0.320699	192.168.0.1	239.255.255.250	SSDP	523	NOTIFY * HTTP/1.1
15	0.408560	192.168.0.1	239.255.255.250	SSDP	459	NOTIFY * HTTP/1.1
16	0.409416	192.168.0.1	239.255.255.250	SSDP	468	NOTIFY * HTTP/1.1

Рис. 2.5: Захват трафика

На устройстве в консоли определим с помощью команды `ipconfig` IP-адрес устройства и шлюз по умолчанию. IP: 172.16.94.216, Шлюз: 172.16.94.1 (рис. 6)

```
Адаптер беспроводной локальной сети Беспроводная сеть:

DNS-суффикс подключения . . . . . :
Локальный IPv6-адрес канала . . . : fe80::2fef:1cd:c5e8:217a%14
IPv4-адрес . . . . . : 172.16.94.216
Маска подсети . . . . . : 255.255.254.0
Основной шлюз . . . . . : 172.16.94.1
```

Рис. 2.6: Определение IP и шлюза

В консоли с помощью команды `ping` адрес_шлюза пропингуем шлюз по умолчанию. (рис. 7)

```
C:\Users\lserg>ping 172.16.94.1

Обмен пакетами с 172.16.94.1 по 32 байтами данных:
Ответ от 172.16.94.1: число байт=32 время=3мс TTL=254
Ответ от 172.16.94.1: число байт=32 время=2мс TTL=254
Ответ от 172.16.94.1: число байт=32 время=4мс TTL=254
Ответ от 172.16.94.1: число байт=32 время=4мс TTL=254

Статистика Ping для 172.16.94.1:
    Пакетов: отправлено = 4, получено = 4, потеряно = 0
    (0% потерь)
Приблизительное время приема-передачи в мс:
    Минимальное = 2мсек, Максимальное = 4 мсек, Среднее = 3 мсек

C:\Users\lserg>
```

Рис. 2.7: ping шлюза

В Wireshark остановим захват трафика. В строке фильтра пропишем фильтр `arp or icmp`. Убедимся, что в списке пакетов отобразятся только пакеты ARP или ICMP, в частности пакеты, которые были сгенерированы с помощью команды `ping`. (рис. 8)

Беспроводная сеть						
Файл Правка Вид Запуск Анализ Статистика Телефония Беспроводная связь Инструменты Справка						
arp or icmp						
Список пакетов		Фильтр отображения		Введите фильтр отображения ...		
Опции: Обычные и многобайтовые		Чувствительность к регистру		Назад Множественные случаи		
No.	Time	Source	Destination	Protocol	Length	Info
1767	121.559542	c2:85:6e:43:31:17	Broadcast	ARP	60	Who has 172.16.94.1? Tell 172.16.94.204
2287	127.896711	c6:0d:74:01:b0:67	Broadcast	ARP	60	Who has 172.16.94.85? Tell 172.16.94.101
2288	127.897691	Apple_d3:3d:43	Broadcast	ARP	60	Who has 172.16.94.85? Tell 172.16.94.74
2296	138.443066	e2:a7:23:73:08:00	Broadcast	ARP	60	Who has 172.16.94.1? Tell 172.16.94.87
2307	140.184151	6e:b9:26:3a:d0:69	Broadcast	ARP	60	ARP Announcement for 172.16.94.111
2308	140.286357	6e:b9:26:3a:d0:69	Broadcast	ARP	60	Who has 172.16.94.1? Tell 172.16.94.111
2323	141.720039	6e:b9:26:3a:d0:69	Broadcast	ARP	60	ARP Announcement for 172.16.94.111
2324	141.720069	6e:b9:26:3a:d0:69	Broadcast	ARP	60	Who has 172.16.94.1? Tell 172.16.94.111
2325	142.027253	6e:b9:26:3a:d0:69	Broadcast	ARP	60	Who has 172.16.94.1? Tell 172.16.94.111
2364	146.547978	172.16.94.216	172.16.94.1	ICMP	74	Echo (ping) request id=0x0001, seq=1/256, ttl=128 (reply in 2365)
2365	146.551747	172.16.94.1	172.16.94.216	ICMP	74	Echo (ping) reply id=0x0001, seq=1/256, ttl=254 (request in 2364)
2367	147.557766	172.16.94.216	172.16.94.1	ICMP	74	Echo (ping) request id=0x0001, seq=2/512, ttl=128 (reply in 2368)
2368	147.559846	172.16.94.1	172.16.94.216	ICMP	74	Echo (ping) reply id=0x0001, seq=2/512, ttl=254 (request in 2367)
2369	148.570093	172.16.94.1	172.16.94.216	ICMP	74	Echo (ping) request id=0x0001, seq=3/768, ttl=128 (reply in 2370)
2370	148.574770	172.16.94.1	172.16.94.216	ICMP	74	Echo (ping) reply id=0x0001, seq=3/768, ttl=254 (request in 2369)
2371	149.586784	172.16.94.216	172.16.94.1	ICMP	74	Echo (ping) request id=0x0001, seq=4/1024, ttl=128 (reply in 2372)
2372	149.591655	172.16.94.1	172.16.94.216	ICMP	74	Echo (ping) reply id=0x0001, seq=4/1024, ttl=254 (request in 2371)
2373	149.912138	92:1b:50:cf:b9:e3	Broadcast	ARP	60	Who has 172.16.94.1? Tell 172.16.94.125
2424	161.073515	Cisco_60:9c:e6	Broadcast	ARP	60	Who has 172.16.94.42? Tell 172.16.94.1

Рис. 2.8: фильтр arp or icmp

Изучим эхо-запрос и эхо-ответ ICMP в программе Wireshark:

Эхо-запрос: длина кадра - 74 байта, относится к типу Ethernet (1), MAC-адрес источника - 90:e8:68:fa:55:e3 (тип индивидуальный, глобально администрируемый, как мы уже определяли), MAC-адрес шлюза - 70:18:a7:60:9c:e6 (тип индивидуальный, глобально администрируемый, т.к 70 в 16 системе = 11100000 в 2). (рис. 9)

→	2364	146.547978	172.16.94.21	172.16.94.1	ICMP	74	Echo (ping) request	id=0x0001, seq=1/256, ttl=128 (reply in 2365)
←	2365	146.551747	172.16.94.1	172.16.94.216	ICMP	74	Echo (ping) reply	id=0x0001, seq=1/256, ttl=254 (request in 2364)
	2367	147.557766	172.16.94.216	172.16.94.1	ICMP	74	Echo (ping) request	id=0x0001, seq=2/512, ttl=128 (reply in 2368)
	2368	147.559846	172.16.94.1	172.16.94.216	ICMP	74	Echo (ping) reply	id=0x0001, seq=2/512, ttl=254 (request in 2367)
	2369	148.570093	172.16.94.216	172.16.94.1	ICMP	74	Echo (ping) request	id=0x0001, seq=3/768, ttl=128 (reply in 2370)
	2370	148.574770	172.16.94.1	172.16.94.216	ICMP	74	Echo (ping) reply	id=0x0001, seq=3/768, ttl=254 (request in 2369)
	2371	149.586784	172.16.94.216	172.16.94.1	ICMP	74	Echo (ping) request	id=0x0001, seq=4/1024, ttl=128 (reply in 2372)
	2372	149.591655	172.16.94.1	172.16.94.216	ICMP	74	Echo (ping) reply	id=0x0001, seq=4/1024, ttl=254 (request in 2371)
	2373	149.912138	92:1b:50:cf:b9:e3	Broadcast	ARP	60	Who has 172.16.94.1? Tell 172.16.94.125	
	2424	161.073515	Cisco_60:9c:e6	Broadcast	ARP	60	Who has 172.16.94.42? Tell 172.16.94.1	

[Time shift for this packet: 0.000000000 seconds]

[Time delta from previous captured frame: 0.932068000 seconds]

[Time delta from previous displayed frame: 4.520725000 seconds]

[Time since reference or first frame: 146.547978000 seconds]

Frame Number: 2364

Frame Length: 74 bytes (592 bits)

Capture Length: 74 bytes (592 bits)

[Frame is marked: False]

[Frame is ignored: False]

[Protocols in frame: eth:ethertype:ip:icmp:data]

[Coloring Rule Name: ICMP]

[Coloring Rule String: icmp || icmpv6]

▼ Ethernet II, Src: AzureWaveTec_fa:55:e3 (90:e8:68:fa:55:e3), Dst: Cisco_60:9c:e6 (70:18:a7:60:9c:e6)

Destination: Cisco_60:9c:e6 (70:18:a7:60:9c:e6)

Source: AzureWaveTec_fa:55:e3 (90:e8:68:fa:55:e3)

Type: IPv4 (0x0800)

[Stream index: 35]

Рис. 2.9: Эхо запрос

Эхо-ответ: длина кадра - 74 байта, относится к типу Ethernet (1), MAC-адрес источника - 70:18:a7:60:9c:e6 (тип индивидуальный, глобально администрируемый), MAC-адрес шлюза - 90:e8:68:fa:55:e3 (тип индивидуальный, глобально администрируемый) (рис. 10)

→	2364	146.547978	172.16.94.216	172.16.94.1	ICMP	74 Echo (ping) request	id=0x0001, seq=1/256, ttl=128 (reply in 2365)
→	2365	146.551747	172.16.94.1	172.16.94.216	ICMP	74 Echo (ping) reply	id=0x0001, seq=1/256, ttl=254 (request in 2364)
→	2367	147.557766	172.16.94.216	172.16.94.1	ICMP	74 Echo (ping) request	id=0x0001, seq=2/512, ttl=128 (reply in 2368)
	2368	147.559846	172.16.94.1	172.16.94.216	ICMP	74 Echo (ping) reply	id=0x0001, seq=2/512, ttl=254 (request in 2367)
	2369	148.570893	172.16.94.216	172.16.94.1	ICMP	74 Echo (ping) request	id=0x0001, seq=3/768, ttl=128 (reply in 2370)
	2370	148.574770	172.16.94.1	172.16.94.216	ICMP	74 Echo (ping) reply	id=0x0001, seq=3/768, ttl=254 (request in 2369)
	2371	149.586784	172.16.94.216	172.16.94.1	ICMP	74 Echo (ping) request	id=0x0001, seq=4/1024, ttl=128 (reply in 2372)
	2372	149.591655	172.16.94.1	172.16.94.216	ICMP	74 Echo (ping) reply	id=0x0001, seq=4/1024, ttl=254 (request in 2371)
	2373	149.912138	92:1b:50:cf:b9:e3	Broadcast	ARP	60 Who has 172.16.94.1? Tell 172.16.94.125	
	2424	161.073515	Cisco_60:9c:e6	Broadcast	ARP	60 Who has 172.16.94.42? Tell 172.16.94.1	

[Time shift for this packet: 0.000000000 seconds]	0000	90 e8 68 fa 55 e3 70 18 a7 60
[Time delta from previous captured frame: 0.003769000 seconds]	0010	00 3c 97 cb 00 00 fe 01 0f fb
[Time delta from previous displayed frame: 0.003769000 seconds]	0020	5e d8 00 00 55 5a 00 01 00 01
[Time since reference or first frame: 146.551747000 seconds]	0030	67 68 69 6a 6b 6c 6d 6e 6f 70
Frame Number: 2365	0040	77 61 62 63 64 65 66 67 68 69


```

[Frame is marked: False]
[Frame is ignored: False]
[Protocols in frame: ethertype:ip:icmp:data]
[Coloring Rule Name: ICMP]
[Coloring Rule String: icmp || icmpv6]
▼ Ethernet II, Src: Cisco_60:9c:e6 (70:18:a7:60:9c:e6), Dst: AzureWaveTec_fa:55:e3 (90:e8:68:fa:55:e3)
  > Destination: AzureWaveTec_fa:55:e3 (90:e8:68:fa:55:e3)
  > Source: Cisco_60:9c:e6 (70:18:a7:60:9c:e6)
  Type: IPv4 (0x0800)
  [Stream index: 35]

```

Рис. 2.10: Эхо ответ

Изучим кадры данных протокола ARP. Изучим данные в полях заголовка Ethernet II.

MAC-адрес назначения (Destination): ff:ff:ff:ff:ff:ff

Тип адреса: Широковещательный (Broadcast). ARP-запрос отправляется всем узлам в сети, так как отправителю неизвестен MAC-адрес получателя.

MAC-адрес источника (Source): 70:18:a7:60:9c:e6

Тип адреса: Индивидуальный, глобально администрируемый. Производитель (OUI): 70:18:a7 — Cisco Systems.

Тип (Type): 0x0806. Назначение: Идентифицирует инкапсулированный протокол. Указывает, что содержимым кадра является пакет протокола ARP (Address Resolution Protocol). (рис. 11)

968	57.752156	Cisco_60:9c:e6	Broadcast	ARP	60 Who has 172.16.94.29? Tell 172.16.94.1	0000	ff ff
1021	65.739480	Cisco_60:9c:e6	Broadcast	ARP	60 Who has 172.16.94.29? Tell 172.16.94.1	0010	00 00
1025	65.740822	TplinkTechno_59:85:1c	Broadcast	ARP	60 Who has 172.16.94.204? Tell 172.16.94.121	0020	00 00
1026	65.741263	TplinkTechno_59:9d:1c	Broadcast	ARP	60 Who has 172.16.94.204? Tell 172.16.94.103	0030	00 00
1035	67.275318	Cisco_60:9c:e6	Broadcast	ARP	60 Who has 172.16.94.75? Tell 172.16.94.1		
1049	70.245344	Cisco_60:9c:e6	Broadcast	ARP	60 Who has 172.16.94.75? Tell 172.16.94.1		
1057	72.087877	Intel_07:c6:99	Broadcast	ARP	60 Who has 172.16.94.110? Tell 172.16.95.24		
1058	72.087917	AzureWaveTec_73:80:1c	Broadcast	ARP	60 Who has 172.16.94.110? Tell 172.16.94.27		


```

> Frame 968: 60 bytes on wire (4800 bits) on interface \Device\NPF{7C57B6A3-1A75-4d
> Ethernet II, Src: Cisco_60:9c:e6 (70:18:a7:60:9c:e6), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
  > Destination: Broadcast (ff:ff:ff:ff:ff:ff)
    .....1. .... = LG bit: Locally administered address (this is NOT the factory default)
    .....1. .... = IG bit: Group address (multicast/broadcast)
  > Source: Cisco_60:9c:e6 (70:18:a7:60:9c:e6)
    .....0. .... = LG bit: Globally unique address (factory default)
    .....0. .... = IG bit: Individual address (unicast)

    Type: ARP (0x0806)
    [Stream index: 71]
    Padding: 0000000000000000000000000000000000000000000000000000000000000000

  > Address Resolution Protocol (request)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: request (1)
    Sender MAC address: Cisco_60:9c:e6 (70:18:a7:60:9c:e6)
    Sender IP address: 172.16.94.1
    Target MAC address: 00:00:00:00:00:00 (00:00:00:00:00:00)
  
```

Начнем новый процесс захвата трафика в Wireshark. На устройстве в консоли пропингуем по имени какой-нибудь известный адрес, я возьму ya.ru (рис. 12)

Рис. 2.12: Пинг ya.ru

В Wireshark остановим захваттрафика. Изучим запросы и ответы протоколов ARP и ICMP. Определим MAC-адреса источника и получателя, определим тип MAC-адресов.

Источник данных:

MAC-адрес: 38:fc:98:ea:cf:68

MAC-адрес: 70:18:a7:60:9c:e6

Тип MAC-адреса: Индивидуальный, Глобально администрируемый.

Мой компьютер (172.16.94.216) хочет отправить пакет на внешний IP 77.88.44.242. По правилам маршрутизации, он отправляет пакет на свой основной шлюз (маршрутизатор). Он уже знает MAC-адрес шлюза (вероятно, из предыдущего ARP-запроса), поэтому кадр Ethernet адресован напрямую на MAC-адрес маршрутизатора Cisco. (рис. 14)

873	16.024582	172.16.94.216	77.88.44.242	ICMP	74 Echo (ping) request	id=0x0001, seq=5/1280, ttl=128 (reply in 874)
874	16.056519	77.88.44.242	172.16.94.216	ICMP	74 Echo (ping) reply	id=0x0001, seq=5/1280, ttl=57 (request in 873)
910	17.031803	172.16.94.216	77.88.44.242	ICMP	74 Echo (ping) request	id=0x0001, seq=6/1536, ttl=128 (reply in 920)
920	17.070511	77.88.44.242	172.16.94.216	ICMP	74 Echo (ping) reply	id=0x0001, seq=6/1536, ttl=57 (request in 910)
931	17.915556	Intel_ea:cf:68	Broadcast	ARP	60 Who has 172.16.94.234? Tell 172.16.94.39	
932	17.920240	Intel_ea:cf:68	Broadcast	ARP	60 Who has 172.16.94.180? Tell 172.16.94.39	
933	17.920246	Intel_ea:cf:68	Broadcast	ARP	60 Who has 172.16.94.246? Tell 172.16.94.39	
934	17.920551	Intel_ea:cf:68	Broadcast	ARP	60 Who has 172.16.94.162? Tell 172.16.94.39	
935	17.920771	Intel_ea:cf:68	Broadcast	ARP	60 Who has 172.16.95.30? Tell 172.16.94.39	
936	17.920999	Intel_ea:cf:68	Broadcast	ARP	60 Who has 172.16.94.201? Tell 172.16.94.39	
937	17.921205	Intel_ea:cf:68	Broadcast	ARP	60 Who has 172.16.95.43? Tell 172.16.94.39	

[Protocols in frame: eth:ethertype:ip:icmp:data]		0000	70 18 a7 60 9c e6 90 e8 68 fa 55 e3 00
[Coloring Rule Name: ICMP]		0010	00 3c 29 04 00 00 00 01 8c 8a ac 10 54
[Coloring Rule String: icmp icmpv6]		0020	2c f2 08 00 4d 56 00 01 00 05 61 62 6:
Ethernet II, Src: AzureWaveTec_fa:55:e3 (90:e8:68:fa:55:e3), Dst: Cisco_60:9c:e6 (70:18:a7:60:9c:e6)		0030	67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 7:
Destination: Cisco_60:9c:e6 (70:18:a7:60:9c:e6)		0040	77 61 62 63 64 65 66 67 68 69
.....0..... = IG bit: Globally unique address (factory default)			
.....0..... = IG bit: Individual address (unicast)			
Source: AzureWaveTec_fa:55:e3 (90:e8:68:fa:55:e3)			
.....0..... = LG bit: Globally unique address (factory default)			
.....0..... = LG bit: Individual address (unicast)			
Type: IPv4 (0x0800)			
[Stream index: 12]			
Internet Protocol Version 4, Src: 172.16.94.216, Dst: 77.88.44.242			
0100 = Version: 4			
.... 0101 = Header Length: 20 bytes (5)			
Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)			
Total Length: 60			
Identification: 0x2904 (10500)			
0000 = Flags: 0x0			

Рис. 2.14: ICMP эхо-запрос

Это ответ, который возвращается с удаленного сервера на ваш компьютер.

Содержание: 77.88.44.242 -> 172.16.94.216

Источник данных в кадре Ethernet II:

MAC-адрес: 70:18:a7:60:9c:e6 (Cisco)

Тип MAC-адреса: Индивидуальный, Глобально администрируемый.

Получатель данных (Destination) в кадре Ethernet II:

MAC-адрес: 90:e8:68:fa:55:e3 (мой комп)

Тип MAC-адреса: Индивидуальный, Глобально администрируемый.

Маршрутизатор получил ответ от сервера и теперь должен доставить его моему компьютеру в локальной сети. Он знает MAC-адрес ПК и отправляет кадр Ethernet напрямую на мой MAC-адрес. (рис. 15)

873	16.024582	172.16.94.216	77.88.44.242	ICMP	74 Echo (ping) request	id=0x0001, seq=5/1280, ttl=128 (reply in 874)
874	16.056519	77.88.44.242	172.16.94.216	ICMP	74 Echo (ping) reply	id=0x0001, seq=5/1280, ttl=57 (request in 873)
910	17.031803	172.16.94.216	77.88.44.242	ICMP	74 Echo (ping) request	id=0x0001, seq=6/1536, ttl=128 (reply in 920)
920	17.070511	77.88.44.242	172.16.94.216	ICMP	74 Echo (ping) reply	id=0x0001, seq=6/1536, ttl=57 (request in 910)
931	17.191556	Intel_ea:cf:68	Broadcast	ARP	60 Who has 172.16.94.234? Tell 172.16.94.39	
932	17.192040	Intel_ea:cf:68	Broadcast	ARP	60 Who has 172.16.94.180? Tell 172.16.94.39	
933	17.192046	Intel_ea:cf:68	Broadcast	ARP	60 Who has 172.16.94.246? Tell 172.16.94.39	
934	17.192051	Intel_ea:cf:68	Broadcast	ARP	60 Who has 172.16.94.162? Tell 172.16.94.39	
935	17.192071	Intel_ea:cf:68	Broadcast	ARP	60 Who has 172.16.95.30? Tell 172.16.94.39	
936	17.192099	Intel_ea:cf:68	Broadcast	ARP	60 Who has 172.16.94.201? Tell 172.16.94.39	
937	17.192105	Intel_ea:cf:68	Broadcast	ARP	60 Who has 172.16.95.43? Tell 172.16.94.39	

[Protocols in frame: eth:ethertype:ip:icmp:data]		0000	90 e8 68 fa 55 e3 70 18
[Coloring Rule Name: ICMP]		0010	00 3c 29 04 00 00 39 01
[Coloring Rule String: icmp icmpv6]		0020	5e d8 00 00 55 56 00 01
▼ Ethernet II, Src: Cisco_60:9c:e6 (70:18:a7:60:9c:e6), Dst: AzureWaveTec_fa:55:e3 (90:e8:68:fa:55:e3)		0030	67 68 69 6a 6b 6c 6d 6e
▼ Destination: AzureWaveTec_fa:55:e3 (90:e8:68:fa:55:e3)		0040	77 61 62 63 64 65 66 67
.....0. = LG bit: Globally unique address (factory default)			
.....0. = IG bit: Individual address (unicast)			
▼ Source: Cisco_60:9c:e6 (70:18:a7:60:9c:e6)			
.....0. = LG bit: Globally unique address (factory default)			
.....0. = IG bit: Individual address (unicast)			
Type: IPv4 (0x0800)			
[Stream index: 12]			
▼ Internet Protocol Version 4, Src: 77.88.44.242, Dst: 172.16.94.216			
0100 = Version: 4			
.... 0101 = Header Length: 20 bytes (5)			
> Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)			
Total Length: 60			
Identification: 0x2904 (10500)			
> 0000 = Flags: 0x0			

Рис. 2.15: ICMP эхо-ответ

2.3 Анализ протоколов транспортного уровня в Wireshark

Запустим Wireshark. Выберем активный на устройстве сетевой интерфейс. Убедимся, что начался процесс захвата трафика. (рис. 16)

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	Intel_ea:cf:68	Broadcast	ARP	60	Who has 172.16.94.72? Tell 172.16.94.39
2	0.181923	172.16.94.78	224.0.0.251	MDNS	290	Standard query response 0x0000 PTR, cache flush Android-6.local PTR, cache
3	0.409087	Intel_ea:cf:68	Broadcast	ARP	60	Who has 172.16.95.30? Tell 172.16.94.39
4	0.548383	172.16.94.216	5.255.255.77	TCP	1464	64995 → 443 [ACK] Seq=1 Ack=1 Win=509 Len=1410
5	0.548383	172.16.94.216	5.255.255.77	TCP	1464	64995 → 443 [ACK] Seq=1411 Ack=1 Win=509 Len=1410 [TCP PDU reassembled in
6	0.548383	172.16.94.216	5.255.255.77	TLSv1.2	174	Application Data
7	0.548444	172.16.94.216	5.255.255.77	TLSv1.2	93	Application Data
8	0.548454	172.16.94.216	5.255.255.77	TLSv1.2	86	Application Data
9	0.553696	5.255.255.77	172.16.94.216	TCP	60	443 → 64995 [ACK] Seq=1 Ack=1411 Win=1988 Len=0
10	0.553734	5.255.255.77	172.16.94.216	TCP	60	443 → 64995 [ACK] Seq=1 Ack=2941 Win=2000 Len=0
11	0.553734	5.255.255.77	172.16.94.216	TCP	60	443 → 64995 [ACK] Seq=1 Ack=2980 Win=2000 Len=0
12	0.553734	5.255.255.77	172.16.94.216	TCP	60	443 → 64995 [ACK] Seq=1 Ack=3012 Win=2000 Len=0
13	0.554277	5.255.255.77	172.16.94.216	TLSv1.2	123	Application Data
14	0.554408	172.16.94.216	5.255.255.77	TLSv1.2	93	Application Data
15	0.565813	5.255.255.77	172.16.94.216	TLSv1.2	204	Application Data
16	0.566657	172.16.94.216	5.255.255.77	TLSv1.2	89	Application Data
17	0.616361	5.255.255.77	172.16.94.216	TCP	60	443 → 64995 [ACK] Seq=220 Ack=3086 Win=2000 Len=0
18	0.922349	Intel_ea:cf:68	Broadcast	ARP	60	Who has 172.16.94.72? Tell 172.16.94.39

Рис. 2.16: Захват трафика

В браузере перейдем на сайт, работающий по протоколу HTTP (например, на сайт CERN <http://info.cern.ch/>). (рис. 17)

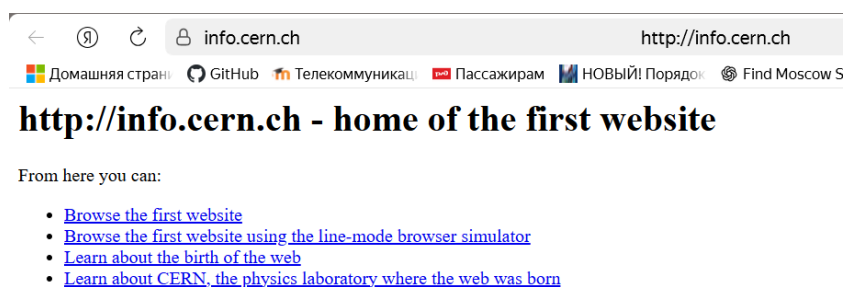


Рис. 2.17: Переход на сайт cern.ch

В Wireshark в строке фильтра укажем http и проанализируем информацию по протоколу TCP в случае запросов и ответов.

Это пакет, в котором компьютер отправляет HTTP-запрос на сервер.

Порты: Source Port (порт источника): 65232 — это исходный порт, который случайным образом выбирается компьютером для этой сессии.

Destination Port (порт назначения): 80 — это порт назначения, стандартный порт для HTTP-сервиса.

Sequence Number (последовательный номер): Relative (относительный): 1, Raw (абсолютный): 4193937875. Это номер первого байта данных в этом сегменте. Wireshark для удобства показывает относительный номер, начиная с 1.

Acknowledgment Number (номер подтверждения): Relative (относительный): 1, Raw (абсолютный): 2237714506. Это номер следующего байта, который отправитель этого пакета ожидает получить от противоположной стороны. Подтверждает успешное получение всех данных до этого номера.

Flags: PSH, ACK. ACK (Acknowledgment): Установлен, чтобы подтвердить получение предыдущих пакетов от сервера. PSH (Push): Указывает получателю (серверу) немедленно передать данные приложению (веб-серверу), не дожидаясь заполнения буфера.

Window: 255 (Расчетный размер: 65280 байт). Это размер окна приема моего компьютера. Он сообщает серверу, сколько данных ваш компьютер готов принять без подтверждения. Используется для управления потоком данных (Flow

Control).

TCP Segment Len (длина сегмента): 254. Размер полезных данных (HTTP-запроса), переносимых в этом TCP-сегменте. (рис. 18)

No.	Time	Source	Destination	Protocol	Length	Info
561	39.445586	104.81.99.218	172.16.94.216	HTTP	369	HTTP/1.1 304 Not Modified
557	39.398068	172.16.94.216	104.81.99.218	HTTP	308	GET /DigiCertGlobalRootG2.crl HTTP/1.1

> Frame 561: 369 bytes on wire (2952 bits), 369 bytes captured (2952 bits) on interface \Device\NPF_{7C57B6A3-1A75-4000-8000-000000000000}	0000	90 e8 68 fa 5
> Ethernet II, Src: Cisco_60:9c:e6 (70:18:a7:60:9c:e6), Dst: AzureWaveTec_fa:55:e3 (90:e8:68:fa:55:e3)	0010	01 63 e7 ed 4
> Internet Protocol Version 4, Src: 104.81.99.218, Dst: 172.16.94.216	0020	5e d8 00 50 f
> Transmission Control Protocol, Src Port: 80, Dst Port: 65232, Seq: 1, Ack: 255, Len: 315	0030	01 f5 60 ed 0
> [Conversation completeness: Incomplete, DATA (15)]	0040	30 34 20 4e 6
> [TCP Segment Len: 315]	0050	0a 43 6f 6e 7
> Sequence Number: 1 (relative sequence number)	0060	70 70 6c 69 6
> Sequence Number (raw): 2237714506	0070	63 72 6c 0d 0
> [Next Sequence Number: 316 (relative sequence number)]	0080	65 64 3a 20 5
> Acknowledgment Number: 255 (relative ack number)	0090	32 30 32 35 2
> Acknowledgment number (raw): 4193938129	00a0	54 0d 0a 45 5
> 0101 = Header Length: 20 bytes (5)	00b0	37 65 2d 34 3
> Flags: 0x018 (PSH, ACK)	00c0	6f 6e 74 72 6
> Window: 501	00d0	6d 61 78 2d 6
> [Calculated window size: 64128]	00e0	70 69 72 65 7
> [Window size scaling factor: 128]	00f0	65 70 20 32 3
> Checksum: 0x00ed [unverified]	0100	20 47 4d 54 0
> [Checksum Status: Unverified]	0110	20 32 39 20 5
> Urgent Pointer: 0	0120	32 33 3a 35 3
> [Timestamps]	0130	63 74 69 6f 6
> [SEQ/ACK analysis]	0140	65 0d 0a 41 6
> TCP payload (315 bytes)	0150	2e 36 35 33 6
	0160	38 36 33 37 2
	0170	0a

Рис. 2.18: Анализ TCP (запрос)

TCP в HTTP-ответе это пакет, в котором сервер отправляет HTTP-ответ моему компьютеру.

Порты: Source Port (порт источника): 80 — сервер отвечает с того же порта, на который был отправлен запрос. Destination Port (порт назначения): 65232 — сервер отправляет ответ на исходный порт компьютера.

Sequence Number (последовательный номер): Relative (относительный): 1, Raw (абсолютный): 2237714506. Это номер первого байта данных, которые сервер отправляет в этом сегменте.

Acknowledgment Number (номер подтверждения): Relative (относительный): 255, Raw (абсолютный): 4193938129

Сервер подтверждает получение 254 байт данных от клиента. Расчет: 1 (отн. Seq клиента) + 254 (длина данных) = 255 (отн. Ack сервера). Сервер ожидает получить от клиента следующий байт с номером 255.

Flags: PSH, ACK. ACK (Acknowledgment): Подтверждает получение HTTP-запроса от клиента. PSH (Push): Указывает компьютеру немедленно передать данные (HTTP-ответ) приложению (браузеру).

Window (окно): 501 (Расчетный размер: 64128 байт). Это размер окна приема сервера. Он сообщает компьютеру, сколько данных сервер готов принять.

TCP Segment Len (длина сегмента): 315. Размер полезных данных (HTTP-ответа), переносимых в этом TCP-сегменте. (рис. 19)

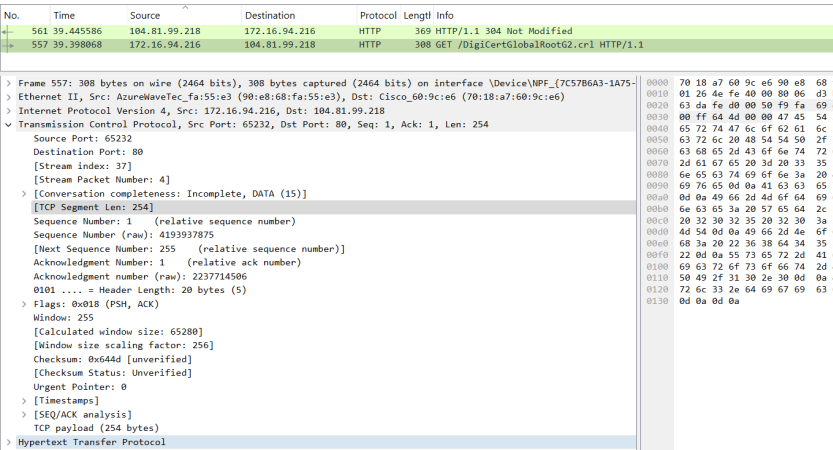


Рис. 2.19: Анализ TCP (ответ)

UDP в DNS-запросе. Это пакет, в котором компьютер отправляет вопрос DNS-серверу.

Source Port (порт источника): 62966. Это исходный порт, который случайным образом выбирается компьютером для этого запроса. Именно на этот порт клиент будет ожидать ответ.

Destination Port (порт назначения): 53. Это стандартный порт для службы DNS. Все DNS-серверы слушают запросы на этом порту.

Length (длина): 55 байт. Общая длина UDP-датаграммы (заголовок + данные). Заголовок UDP всегда 8 байт, значит, полезные данные (DNS-запрос) составляют $55 - 8 = 47$ байт.

Checksum (контрольная сумма): 0x3b31 (unverified). Контрольная сумма для проверки целостности заголовка и данных. Wireshark отмечает ее статус как Unverified, так как для проверки требуется полный пакет, включая фиктивный заголовок IP, который может отсутствовать в захвате. (рис. 20)

693	46.857470	37.18.92.5	172.16.94.216	DNS	540	Standard query response 0xa181 A v10.events.data.microsoft.com CNAME win-global-asia
692	46.853078	172.16.94.216	37.18.92.5	DNS	89	Standard query response 0xa181 A v10.events.data.microsoft.com
554	39.363036	193.232.218.194	172.16.94.216	DNS	212	Standard query response 0xa159 A crl3.digicert.com CNAME crl.edge.digicert.com CNAME
552	39.347310	37.18.92.5	172.16.94.216	DNS	533	Standard query response 0xa159 A crl3.digicert.com CNAME crl.edge.digicert.com CNAME
551	39.325490	172.16.94.216	193.232.218.194	DNS	77	Standard query response 0xa159 A crl3.digicert.com
550	39.289182	172.16.94.216	37.18.92.5	DNS	77	Standard query response 0xa159 A crl3.digicert.com
398	27.857349	37.18.92.5	172.16.94.216	DNS	149	Standard query response 0xa6f86 HTTPS suggest.sso.dzen.ru SOA ns1.mail.ru
391	27.845653	37.18.92.5	172.16.94.216	DNS	263	Standard query response 0xa6f86 A dzen.ru A 5.61.23.39 A 185.180.200.2 A 83.222.28.15
390	27.845653	37.18.92.5	172.16.94.216	DNS	145	Standard query response 0xa6f86 HTTPS suggest.sso.dzen.ru SOA ns1.mail.ru
389	27.845653	37.18.92.5	172.16.94.216	DNS	247	Standard query response 0xa6f86 A suggest.sso.dzen.ru A 87.250.254.106 NS ns2.mail.ru NS
384	27.845653	37.18.92.5	172.16.94.216	DNS	255	Standard query response 0xa6f86 A suggest.sso.dzen.ru A 87.250.254.106 NS ns2.mail.ru
376	27.827637	37.18.92.5	172.16.94.216	DNS	137	Standard query response 0xa6f86 HTTPS suggest.sso.dzen.ru SOA ns1.mail.ru
374	27.786857	172.16.94.216	37.18.92.5	DNS	91	Standard query response 0xa6f86 HTTPS suggest.sso.dzen.ru

Frame 693: 540 bytes on wire (4320 bits), 540 bytes captured (4320 bits) on interface \Device\NPF... (7C5786A3-1A75-4000-8000-000000000000) on interface 0x... Ethernet II, Src: Cisco, 68:9c:e6 (78:18:a7:68:9c:e6), Dst: AzureWaveTec_fa:55:e3 (90:e8:68:fa:55:e3) Internet Protocol Version 4, Src: 37.18.92.5, Dst: 172.16.94.216 User Datagram Protocol, Src Port: 53, Dst Port: 62966 Source Port: 53 Destination Port: 62966 Length: 506 Checksum: 0xbbc2 [unverified] [Checksum Status: Unverified] [Stream Index: 37] [Stream Packet Number: 2] [Timestamps] [Time since first frame: 0.004392000 seconds] [Time since previous frame: 0.004392000 seconds] UDP payload (498 bytes)		0020 5e 48 00 35 f5 f6 01 fa bb e2 a1 81 81 81 81 81 81 0030 00 03 00 04 00 08 03 76 31 30 06 65 76 31 30 06 65 76 31 0040 73 04 64 61 74 61 09 6d 69 63 72 6f 73 04 64 61 74 61 0050 01 63 6f 6d 00 00 01 00 01 c0 0c 00 05 01 c0 0c 00 05 0060 00 00 01 00 38 23 77 69 6e 2d 67 6c 6f 6e 2d 67 6c 6f 0070 2d 61 73 69 6d 6f 76 2d 6c 65 61 66 73 61 66 73 61 66 0080 65 6e 74 73 2d 64 63 74 61 0e 74 72 63 61 0e 74 72 63 0090 63 6d 61 6e 61 67 65 72 03 6e 65 74 00 03 6e 65 74 00 00a0 05 00 01 00 00 00 00 2c 10 6f 6e 65 6f 6e 65 6f 6e 65 00b0 6f 6c 70 72 64 63 75 30 34 09 63 65 6f 6c 70 72 64 63 00c0 61 6c 75 73 08 63 6c 6f 75 64 61 70 70 70 70 70 70 70 00d0 75 72 65 c0 26 c0 7f 00 01 00 01 00 00 01 00 01 00 00 00e0 04 34 b6 8f c0 c0 00 00 02 00 01 00 00 01 00 00 00 00f0 18 07 6e 73 34 2d 32 30 31 09 61 7a 75 75 75 75 75 0100 64 6e 73 04 69 6e 66 6f 00 c0 90 00 02 00 c0 90 00 02 0110 00 0d 25 00 17 07 6e 73 33 2d 32 30 31 31 31 31 31 0120 75 72 65 2d 64 6e 73 03 6f 67 67 00 c0 6f 67 67 00 c0 0130 00 01 00 0d 25 00 14 07 6e 73 31 2d 32 30 31 31 31 31
--	--	---

Рис. 2.20: Анализ UDP (запрос)

UDP в DNS-ответе. Это пакет, в котором DNS-сервер возвращает ответ компьютеру.

Source Port (порт источника): 53. Сервер отвечает со своего стандартного порта для DNS.

Destination Port (порт назначения): 62966. Сервер отправляет ответ именно на тот исходный порт, с которого клиент отправил запрос. Это важно, чтобы ответ попал в нужное приложение на клиенте.

Length (длина): 506 байт. Общая длина этой UDP-датаграммы. Полезные данные (DNS-ответ) составляют $506 - 8 = 498$ байт. Ответ значительно больше запроса, так как содержит не только запрошенную информацию, но и CNAME-запись (псевдоним).

Checksum (контрольная сумма): 0xbbc2 (unverified). Аналогично запросу, контрольная сумма для проверки целостности данных ответа. (рис. 21)

693	46.057470	37.18.92.5	172.16.94.216	DNS	540 Standard query response 0xa181 A v10.events.data.microsoft.com CNAME win-g
692	46.053078	172.16.94.216	37.18.92.5	DNS	89 Standard query 0xa181 A v10.events.data.microsoft.com
554	39.363036	193.232.218.194	172.16.94.216	DNS	212 Standard query response 0x4159 A crl3.digicert.com CNAME crl.edge.digicert
552	39.347310	37.18.92.5	172.16.94.216	DNS	533 Standard query response 0x4159 A crl3.digicert.com CNAME crl.edge.digicert
551	39.325490	172.16.94.216	193.232.218.194	DNS	77 Standard query 0x4159 A crl3.digicert.com
550	39.289182	172.16.94.216	37.18.92.5	DNS	77 Standard query 0x4159 A crl3.digicert.com
398	27.857349	37.18.92.5	172.16.94.216	DNS	149 Standard query response 0xf86 HTTPS suggest.sso.dzen.ru SOA ns1.mail.ru
391	27.845653	37.18.92.5	172.16.94.216	DNS	263 Standard query response 0xf268 A dzen.ru A 5.61.23.39 A 185.180.200.2 A 83
390	27.845653	37.18.92.5	172.16.94.216	DNS	145 Standard query response 0x9d92 HTTPS suggest.dzen.ru SOA ns1.mail.ru
389	27.845653	37.18.92.5	172.16.94.216	DNS	247 Standard query response 0xd6ec A suggest.dzen.ru A 87.250.254.106 NS ns1.m
384	27.845653	37.18.92.5	172.16.94.216	DNS	255 Standard query response 0x0a8a A suggest.sso.dzen.ru A 87.250.254.106 NS n
376	27.827637	37.18.92.5	172.16.94.216	DNS	137 Standard query response 0xec7f HTTPS dzen.ru SOA ns1.mail.ru
374	27.786857	172.16.94.216	37.18.92.5	DNS	91 Standard query 0xf86 HTTPS suggest.sso.dzen.ru

>	Frame 692: 89 bytes on wire (712 bits), 89 bytes captured (712 bits) on interface \Device\NPF_{7C5786A3-1A75-461D-8000-000000000000}	0000	70 18 a7 60 9c e6 90 e8 68 f2
>	Ethernet II, Src: AzureWaveTec_fa:55:e3 (90:e8:68:fa:55:e3), Dst: Cisco_60:9c:e6 (78:18:a7:60:9c:e6)	0010	00 4b 60 68 00 00 80 11 4e 34
>	Internet Protocol Version 4, Src: 172.16.94.216, Dst: 37.18.92.5	0020	5c 05 f5 f6 00 35 00 37 3b 31
>	User Datagram Protocol, Src Port: 62966, Dst Port: 53	0030	00 00 00 00 00 00 03 76 31 36
>	Source Port: 62966	0040	73 04 64 61 74 61 09 6d 69 61
>	Destination Port: 53	0050	03 63 6f 6d 00 00 01 00 01
>	Length: 55		
>	Checksum: 0x3b31 [unverified]		
>	[Checksum Status: Unverified]		
>	[Stream index: 37]		
>	[Stream Packet Number: 1]		
>	[Timestamps]		
>	Time since first frame: 0.000000000 seconds]		
>	Time since previous frame: 0.000000000 seconds]		
>	UDP payload (47 bytes)		
>	Domain Name System (query)		

Рис. 2.21: Анализ UDP (ответ)

Wireshark в строке фильтра укажем quic и проанализируем информацию по протоколу quic в случае запросов и ответов.

QUIC работает поверх UDP, поэтому в заголовках мы сначала видим Ethernet, затем IP, затем UDP, и только потом — QUIC.

QUIC-пакет от Клиента. Это пакет, который компьютер отправляет на сервер.

Транспорт (UDP): Source Port (порт источника): 62198, Destination Port (порт назначения): 443 (порт QUIC/HTTPS сервера)

QUIC Заголовок:

Header Form: Short Header (0). Короткий заголовок используется после установления соединения, что указывает на то, что это действующие данные уже установленного QUIC-соединения.

Fixed Bit: True. Обязательный бит, всегда должен быть установлен в 1 в корректных QUIC-пакетах.

Spin Bit: False. Экспериментальный бит, который может использоваться для измерения задержки в сети. Его значение не несет функциональной нагрузки для соединения.

Destination Connection ID (DCID): f8a9378badb7bea3. Идентификатор соединения назначения. Это уникальный идентификатор, который клиент использует для указания, с каким именно серверным соединением связан этот пакет. Сервер изначально предоставляет этот ID клиенту.

Полезная нагрузка: Remaining Payload: d6e43ad86a11... Основные данные протокола (например, кадры HTTP/3). Как видно из названия пакета в общем списке — “Protected Payload” — эта часть зашифрована. Это одно из ключевых преимуществ QUIC — шифрование , и даже заголовки полей, относящиеся к управлению соединением, шифруются. (рис. 22)

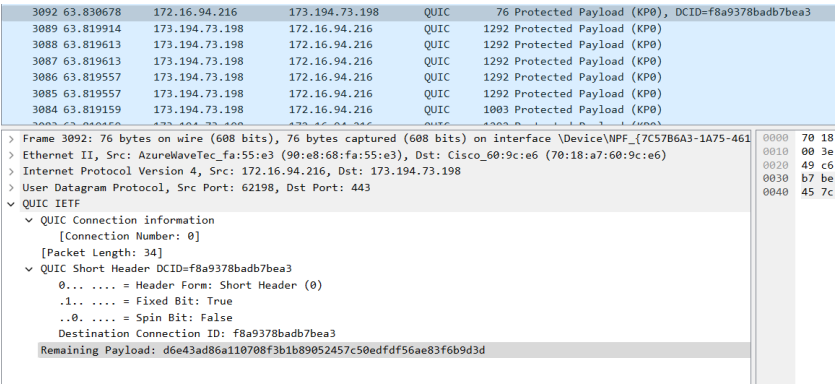


Рис. 2.22: Анализ QUIC (запрос)

QUIC-пакет от Сервера. Это один из многих пакетов, которые сервер отправляет компьютеру в ответ.

Транспорт (UDP): Source Port (порт источника): 443, Destination Port (порт назначения): 62198 (сервер отправляет ответ на исходный порт ПК)

QUIC Заголовок:

Header Form: Short Header (0). Сервер также использует короткий заголовок, подтверждая, что это данные установленного соединения.

Fixed Bit: True

Spin Bit: False

В ответах сервер будет использовать тот Connection ID, который был предоставлен клиенту (или новый, согласованный в процессе установления соединения).

Полезная нагрузка: Remaining Payload: 3fcea12c11cd... Аналогично клиентскому пакету, полезная нагрузка сервера также является “Protected Payload” (Защищенной) и зашифрована. (рис. 23)

3089	63.819914	173.194.73.198	172.16.94.216	QUIC	1292 Protected Payload (KP0)
3088	63.819613	173.194.73.198	172.16.94.216	QUIC	1292 Protected Payload (KP0)
3087	63.819613	173.194.73.198	172.16.94.216	QUIC	1292 Protected Payload (KP0)
3086	63.819557	173.194.73.198	172.16.94.216	QUIC	1292 Protected Payload (KP0)
3085	63.819557	173.194.73.198	172.16.94.216	QUIC	1292 Protected Payload (KP0)
3084	63.819159	173.194.73.198	172.16.94.216	QUIC	1003 Protected Payload (KP0)
Length: 1250					
Checksum: 0x2c11 [unverified]					
[Checksum Status: Unverified]					
[Stream index: 84]					
[Stream Packet Number: 819]					
[Timestamps]					
[Time since first frame: 0.292084000 seconds]					
[Time since previous frame: 0.000301000 seconds]					
UDP payload (1250 bytes)					
QUIC IETF					
QUIC Connection information					
[Connection Number: 0]					
[Packet Length: 1250]					
QUIC Short Header					
0... = Header Form: Short Header (0)					
1... = Fixed Bit: True					
..0. = Spin Bit: False					
Remaining Payload [-]: 3fcea12c11c43be28dfbb8ba073cdfb9edfe90082cc44a85085309ced00b8034443b5f3420bc3a1d7bf57					
0020	5e d8 01 bb f				
0030	c4 3b e2 8d f				
0040	c4 4a 85 08 5				
0050	bc 3a 1d 7b f				
0060	08 41 2e 3f 6				
0070	ed ba dc f5 f				
0080	3f 4f bf eb b				
0090	9e b7 5e 31 e				
00a0	9e 77 71 2b a				
00b0	49 59 84 04 b				
00c0	8f b2 39 c7 8				
00d0	a5 c7 d7 41 9				
00e0	68 1b d3 79 b				
00f0	06 02 0e f9 7				
0100	f0 ad 7d ed 3				
0110	f6 8f e1 b2 1				
0120	69 fb cb 13 4				
0130	5d 7f 80 66 d				
0140	f4 ce 9b 5a 8				
0150	fe 83 8d 93 f				
0160	30 16 1c 0e b				
0170	2c 4c 9b dc 2				

Рис. 2.23: Анализ QUIC (ответ)

2.4 Анализ handshake протокола TCP в Wireshark

Запустим Wireshark. Выберем активный на устройстве сетевой интерфейс. Убедимся, что начался процесс захвата трафика. (рис. 24)

Захват из Беспроводная сеть					
Файл Правка Вид Запуск Захват Анализ Статистика Телефония Беспроводная связь Инструменты Справка					
Примените фильтр отображения: <Ctrl>/>					
No.	Time	Source	Destination	Protocol	Length Info
1	0.000000	172.16.94.91	224.0.0.251	NDIS	336 Standard query response 0x0000 TXT, cache flush PTR _apple-mobdev2._tcp.local PTR 4
2	0.198634	Intel_ea:cf:68	Broadcast	ARP	60 Who has 172.16.94.72? Tell 172.16.94.39
3	0.198666	Intel_ea:cf:68	Broadcast	ARP	60 Who has 172.16.94.234? Tell 172.16.94.39
4	0.492084	172.16.94.216	80.239.137.146	TCP	54 65204 → 443 [FIN, ACK] Seq=1 Ack=1 Win=1022 Len=0
5	0.507998	172.16.95.60	224.0.0.251	NDIS	325 Standard query 0x0000 PTR _companion-link._tcp.local, "QM" question PTR _rdlink._tc
6	0.711214	Intel_ea:cf:68	Broadcast	ARP	60 Who has 172.16.94.148? Tell 172.16.94.39
7	2.659052	204.79.197.222	172.16.94.216	TCP	1514 443 → 65452 [ACK] Seq=1 Ack=1 Win=16379 Len=1460 [TCP PDU reassembled in 11]
8	2.659088	172.16.94.216	204.79.197.222	TCP	54 65452 → 443 [ACK] Seq=1 Ack=1461 Win=1024 Len=0
9	2.659114	204.79.197.222	172.16.94.216	TCP	1514 443 → 65452 [ACK] Seq=1461 Ack=1 Win=16379 Len=1460 [TCP PDU reassembled in 11]
10	2.659139	172.16.94.216	204.79.197.222	TCP	54 65452 → 443 [ACK] Seq=1 Ack=2021 Win=1024 Len=0
11	2.659146	204.79.197.222	172.16.94.216	TLSv1.2	1159 Application Data
12	2.659168	172.16.94.216	204.79.197.222	TCP	54 65452 → 443 [ACK] Seq=1 Ack=4026 Win=1019 Len=0
13	2.659174	204.79.197.222	172.16.94.216	TLSv1.2	133 Application Data
14	2.659197	172.16.94.216	204.79.197.222	TCP	54 65452 → 443 [ACK] Seq=1 Ack=4105 Win=1019 Len=0

Рис. 2.24: Захват трафика

Использую соединение по HTTP с каким-то сайтом для захвата в Wireshark пакетов TCP. В Wireshark проанализируем handshake протокола TCP.

Шаг 1: SYN (Синхронизация). Клиент инициирует соединение, отправляя серверу специальный пакет.

Источник: 172.16.94.216:50451, Назначение: 122.189.32.168:80

Ключевые поля TCP:

Sequence Number: 0. Клиент генерирует начальный номер последовательности (ISN). Wireshark для удобства показывает его как 0.

Flags: SYN (0x002). Установлен только флаг SYN. Это запрос на синхронизацию и начало соединения.

Window: 65535. Размер окна, который клиент готов принимать.

Параметры (Options): MSS=1460 (Maximum Segment Size) — максимальный размер сегмента, который клиент может принять, WS=256 (Window Scaling) — фактор масштабирования окна для увеличения его эффективного размера, SACK_PERM — поддержка выборочных подтверждений (Selective Acknowledgements).

Клиент говорит серверу: я хочу установить соединение. Мой начальный номер последовательности — 0, и вот мои параметры. (рис. 25)

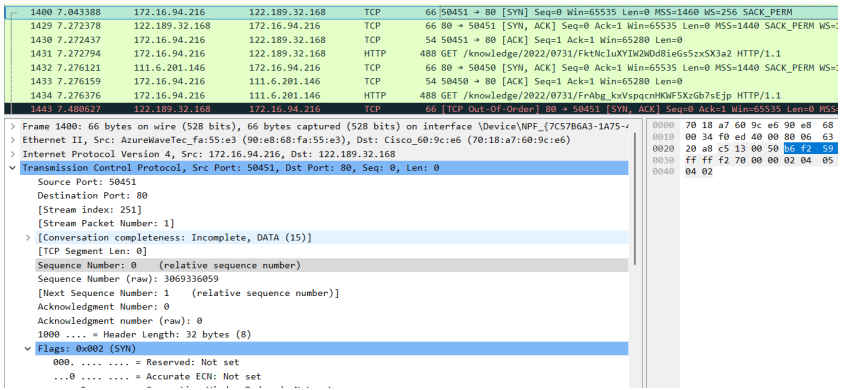


Рис. 2.25: TCP handshake шаг 1

Шаг 2: SYN-ACK (Синхронизация-Подтверждение). Сервер отвечает, подтверждая запрос клиента и отправляя свой собственный запрос на синхронизацию.

Источник: 122.189.32.168:80, Назначение: 172.16.94.216:50451

Ключевые поля TCP:

Sequence Number: 0. Сервер генерирует свой собственный начальный номер последовательности (ISN).

Acknowledgment Number: 1. Это поле подтверждения. Сервер подтверждает получение SYN-пакета клиента.

Flags: SYN, ACK (0x012). Установлены флаги SYN (запрос синхронизации от сервера) и ACK (подтверждение пакета клиента).

Сервер отвечает клиенту: Я получил запрос на соединение (ACK=1). Я согласен установить соединение, и мой начальный номер последовательности — 0 (SYN). Вот мои параметры. (рис. 26)

No.	Time	Source	Destination	Protocol	Length	Info
1428	7.262808	172.16.94.216	111.6.201.146	TCP	66	58454 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
1429	7.272378	122.189.32.168	172.16.94.216	TCP	66	80 → 58451 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 SACK_PERM WS=1024
1430	7.272437	172.16.94.216	122.189.32.168	TCP	54	58451 → 80 [ACK] Seq=1 Ack=1 Win=65280 Len=0
1431	7.272794	172.16.94.216	122.189.32.168	HTTP	488	GET /knowledge/2022/0731/FktNcluxYIN2MD8ieG5z5X3a2 HTTP/1.1
1432	7.276121	111.6.201.146	172.16.94.216	TCP	66	80 → 58450 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1440 SACK_PERM WS=1024
1433	7.276159	172.16.94.216	111.6.201.146	TCP	54	58450 → 80 [ACK] Seq=1 Ack=1 Win=65280 Len=0
1434	7.276196	172.16.94.216	111.6.201.146	HTTP	488	GET /knowledge/2022/0731/FrAbg_kvVspqcnHMF5XzG87aEjp HTTP/1.1
1435	7.371056	172.16.94.216	37.18.92.5	TCP	54	[TCP Retransmission] 50452 → 53 [FIN, ACK] Seq=42 Ack=584 Win=65280 Len=0
1437	7.371998	172.16.94.216	37.18.92.5	TCP	54	[TCP Retransmission] 50449 → 53 [FIN, ACK] Seq=35 Ack=108 Win=65280 Len=0
1438	7.417988	172.16.94.216	37.18.92.5	TCP	54	50395 → 53 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
1439	7.466151	172.16.94.216	37.18.92.5	TCP	54	[TCP Retransmission] 50452 → 53 [FIN, ACK] Seq=42 Ack=584 Win=64768 Len=0
1440	7.466208	172.16.94.216	37.18.92.5	TCP	54	[TCP Retransmission] 50453 → 53 [FIN, ACK] Seq=42 Ack=234 Win=65280 Len=0
1443	7.480627	122.189.32.168	172.16.94.216	TCP	66	[TCP Out-Of-Order] 80 → 58451 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460


```
[Stream index: 251]
[Stream Packet Number: 2]
> [Conversation completeness: Incomplete, DATA (15)]
[TCP Segment Len: 0]
Sequence Number: 0 (relative sequence number)
Sequence Number (raw): 2590184539
[Next Sequence Number: 1 (relative sequence number)]
Acknowledgment Number: 1 (relative ack number)
Acknowledgment number (raw): 3069336060
1000 ... = Header Length: 32 bytes (8)
▼ Flags: 0x012 (SYN, ACK)
 000. .... = Reserved: Not set
...0 .... = Accurate ECH: Not set
...0 .... = Congestion Window Reduced: Not set
...0 .... = ECH-Echo: Not set
...0 .... = Urgent: Not set
...1 .... = Acknowledgment: Set
...0 .... = Push: Not set
...0 .... = Reset: Not set
```

Рис. 2.26: TCP handshake шаг 2

Шаг 3: ACK (Подтверждение). Клиент завершает рукопожатие, подтверждая SYN-пакет сервера.

Источник: 172.16.94.216:50451, Назначение: 122.189.32.168:80

Ключевые поля TCP:

Sequence Number (относительный): 1. Номер последовательности клиента теперь 1, так как его SYN-пакет “потребил” номер 0.

Acknowledgment Number (относительный): 1. Это поле подтверждения. Клиент подтверждает получение SYN-пакета сервера. Номер подтверждения равен ISN сервера + 1. Ack=1.

Flags: ACK (0x010). Установлен только флаг ACK.

Window: 65280. Размер окна обновлен после применения масштабирования (65535 / 256 ≈ 255 -> 255 * 256 = 65280).

Клиент говорит серверу: Я получил SYN-пакет. Соединение установлено. (рис. 27)

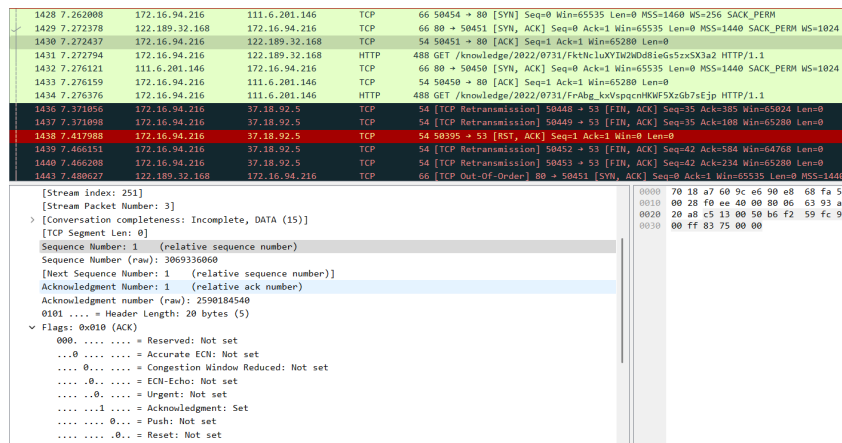


Рис. 2.27: TCP handshake шаг 3

В Wireshark в меню «Статистика» выберем «График Потока».

При установлении соединения TCP используется трёхстороннее рукопожатие. Клиент отправляет сегмент с флагом SYN, где указывает начальный номер последовательности, MSS, размер окна и другие параметры. Сервер отвечает сегментом SYN+ACK с собственными параметрами и подтверждает приём SYN клиента. Далее клиент отправляет ACK, подтверждая SYN сервера. После этого соединение считается установленным. Изменение значений Seq и Ack связано с нумерацией байтов и подтверждением их приёма, а параметры Win, MSS и WS — с настройкой характеристик передачи данных между сторонами.

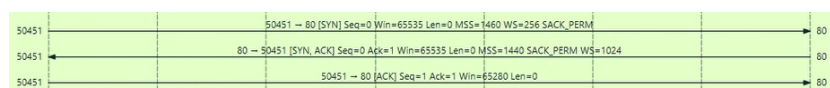


Рис. 2.28: График потока

3 Выводы

В ходе выполнения лабораторной работы мы изучили посредством Wireshark кадры Ethernet, проанализировали PDU протоколы транспортного и прикладного уровней стека TCP/IP.