

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

Лабораторная работа №3

дисциплина: Реляционные базы данных

Студент: Сахно Никита

Группа: НФИбд-02-23

МОСКВА

2025 г.

Цель работы

Изучить посредством Wireshark кадры Ethernet, проанализировать PDU протоколы транспортного и прикладного уровней стека TCP/IP.

Задание

1. Изучение возможностей команды ipconfig для ОС типа Windows (ifconfig для систем типа Linux).
 2. Определение MAC-адреса устройства и его типа.
 3. С помощью Wireshark захватить и проанализировать пакеты ARP и ICMP в части кадров канального уровня.
 4. С помощью Wireshark захватить и проанализировать пакеты HTTP, DNS в части заголовков и информации протоколов TCP, UDP, QUIC.
 5. С помощью Wireshark проанализировать handshake протокола TCP.

Выполнение лабораторной работы

№1

С помощью команды ipconfig для ОС типа Windows выведем информацию о текущем сетевом соединении. Мой провайдер – beeline. Отсюда мы можем узнать IPv6-адрес, IPv4-адрес (уникальный IPv4-адрес узла), маску подсети (используется для определения сетевой и узловой частей IPv4-адреса) и шлюз.

Рисунок 1. Команда ipconfig

Используем также опцию `/all` для вывода более подробной информации.

c:\Windows\system32>ipconfig /all

Настройка протокола TR для Windows

Имя компьютера : DESKTOP-0358DT0
Основной DNS-суффикс :
Тип узла : Гибридный
IP-маршрутизация включена : Нет
WINS-прокси включен : Нет
Порядок просмотра суффиксов DNS : beeline

Адаптер беспроводной локальной сети Подключение по локальной сети* 1:

Адаптер беспроводной локальной сети Подключение по локальной сети* 2:

Адаптер беспроводной локальной сети Беспроводная сеть:

Рисунок 2. Команда ipconfig /all

Определим MAC-адреса сетевых интерфейсов на моем компьютере.

У меня есть помимо основной беспроводной сети WI-FI еще две локальные сети. MAC-адрес для первого виртуального адаптера: 58-96-1D-DC-86-ED. MAC-адрес состоит из 6 октетов: первые 3 октета идентифицируют производителя, последние 3 октета идентифицируют сетевой интерфейс. Я проверила на специальном сайте, что у меня производитель INTEL. Далее разберем первый байт MAC-адреса (58), переведем в двоичный код 58 = 01011000. Нас интересуют последние два бита (нулевой и первый биты). У меня оба нули => мой адрес индивидуальный и глобально администрируемый.

MAC-адрес для второго виртуального адаптера: 5A-96-1D-DC-86-EC.

Адаптер беспроводной локальной сети Подключение по локальной сети* 1:

Адаптер беспроводной локальной сети Подключение по локальной сети* 2:

Состояние среды. Среда передачи недоступна.

DNS-суффикс подключения

Описание: Microsoft Wi-Fi Direct Virtual Adapter #2

Физический адрес : : 5A-96-1D-DC-86-EC

Переводим первый байт в двоичный код $5A = 01011010$. Этот адрес является индивидуальным и локально администрируемым.

MAC-адрес для беспроводной сети WI-FI: 58-96-1D-DC-86-EC.
Переводим первый байт в двоичный код $58 = 01011000$. Этот адрес является индивидуальным и глобально администрируемым (производитель INTEL).

Адаптер беспроводной локальной сети Беспроводная сеть:

Рисунок 4. MAC-адрес в случае WI-FI

№2

Из предыдущего задания мы узнали адрес основного шлюза: 192.168.1.1.

Теперь пропингуем его, предварительно запустив Wireshark и включив захват трафика. Посылаются 4 пакета, 4 пакета получено назад.

```
C:\WINDOWS\system32>ping 192.168.1.1

Обмен пакетами с 192.168.1.1 по с 32 байтами данных:
Ответ от 192.168.1.1: число байт=32 время=1мс TTL=64

Статистика Ping для 192.168.1.1:
    Пакетов: отправлено = 4, получено = 4, потеряно = 0
              (0% потеря)
Приблизительное время приема-передачи в мс:
    Минимальное = 1мсек, Максимальное = 1 мсек, Среднее = 1 мсек
```

Рисунок 5. Пингование шлюза

В строке фильтра пропишем фильтр `icmp`. Убедимся, что в списке

No.	icmp icmpv6		Source	Destination	Protocol	Length	Info
	3322		192.168.1.72	192.168.1.1	ICMP	74	Echo (ping) request
15	4.069631		192.168.1.1	192.168.1.72	ICMP	74	Echo (ping) reply
20	5.085827		192.168.1.72	192.168.1.1	ICMP	74	Echo (ping) request
21	5.087092		192.168.1.1	192.168.1.72	ICMP	74	Echo (ping) reply
28	6.094188		192.168.1.72	192.168.1.1	ICMP	74	Echo (ping) request
29	6.095532		192.168.1.1	192.168.1.72	ICMP	74	Echo (ping) reply
32	7.112141		192.168.1.72	192.168.1.1	TCP	74	Echo (ping) request

Рисунок 6. Пакеты ICMP

пакетов отобразятся только пакеты ICMP, в частности пакеты, которые были сгенерированы с помощью команды ping, отправленной с моего устройства на шлюз по умолчанию.

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

На панели списка пакетов (верхний раздел) выберем первый указанный кадр ICMP — эхо-запрос. Изучим информацию на панели сведений о пакете в средней части экрана. На вкладке физического уровня можно найти длину кадра (74 бита), тип Ethernet – Ethernet (1).

```
✓ Frame 14: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface
  Section number: 1
  > Interface id: 0 (\Device\NPF_{D13C22B7-8811-4040-B55E-62D23A46C914})
    Encapsulation type: Ethernet (1)
    Arrival Time: Sep 23, 2023 19:39:07.251288000 RTZ 2 (зима)
      [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1695487147.251288000 seconds
      [Time delta from previous captured frame: 0.565813000 seconds]
      [Time delta from previous displayed frame: 0.000000000 seconds]
      [Time since reference or first frame: 4.068322000 seconds]
    Frame Number: 14
    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: IntelCor dc:86:ec (58:96:1d:dc:86:ec), Dst: SernetSu bf:13:f4
```

Рисунок 7. Кадр физического уровня

Чтобы узнать MAC-адрес источника и шлюза перейдем на канальный уровень. Адрес источника (Source, откуда запрос отправлен) – это адрес моего устройства (58-96-1D-DC-86-EC). Адрес шлюза (destination, то куда отправлен запрос) - 58-76-AC-BF-13-F4. Тип адреса тут указан (показаны нулевые и первые биты MAC-адресов). Что адрес источника, что адрес шлюза индивидуальные и глобально администрируемые.

▼ Ethernet II, Src: IntelCor_dc:86:ec (58:96:1d:dc:86:ec), Dst: SernetSu_bf:13:f4 (58:76:ac:bf:13:f4)
▼ Destination: SernetSu_bf:13:f4 (58:76:ac:bf:13:f4)
Address: SernetSu_bf:13:f4 (58:76:ac:bf:13:f4)
.... .0. = LG bit: Globally unique address (factory default)
.... .0. = IG bit: Individual address (unicast)
▼ Source: IntelCor_dc:86:ec (58:96:1d:dc:86:ec)
Address: IntelCor_dc:86:ec (58:96:1d:dc:86:ec)
.... .0. = LG bit: Globally unique address (factory default)
.... .0. = IG bit: Individual address (unicast)
Type: IPv4 (0x0800)

Рисунок 8. Кадр канального уровня

Далее посмотрим на полученный ответ. Тут все почти то же самое, что и в запросе (длина кадра 74 бита). Только теперь MAC-адрес источника – MAC-адрес шлюза (58-76-AC-BF-13-F4), а адрес назначения – адрес моего устройства (58-96-1D-DC-86-EC).

No.	Time	Source	Destination	Protocol	Length	Info
14	4.068322	192.168.1.72	192.168.1.1	ICMP	74	Echo (ping) request id=0x00
15	4.069631	192.168.1.1	192.168.1.72	ICMP	74	Echo (ping) reply id=0x00
20	5.085827	192.168.1.72	192.168.1.1	ICMP	74	Echo (ping) request id=0x00
21	5.087092	192.168.1.1	192.168.1.72	ICMP	74	Echo (ping) reply id=0x00
28	6.094188	192.168.1.72	192.168.1.1	ICMP	74	Echo (ping) request id=0x00
29	6.095532	192.168.1.1	192.168.1.72	ICMP	74	Echo (ping) reply id=0x00
32	7.112141	192.168.1.72	192.168.1.1	ICMP	74	Echo (ping) request id=0x00
33	7.113436	192.168.1.1	192.168.1.72	ICMP	74	Echo (ping) reply id=0x00

▼ Frame 15: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface \Device\NPF_{D13C22B7-8811-4040-B55E-62D23A46C914}
Section number: 1
Interface id: 0 (\Device\NPF_{D13C22B7-8811-4040-B55E-62D23A46C914})
Encapsulation type: Ethernet (1)
Arrival Time: Sep 23, 2023 19:39:07.252597000 RTZ 2 (зима)
[Time shift for this packet: 0.000000000 seconds]
Epoch Time: 1695487147.252597000 seconds
[Time delta from previous captured frame: 0.001309000 seconds]
[Time delta from previous displayed frame: 0.001309000 seconds]
[Time since reference or first frame: 4.069631000 seconds]
Frame Number: 15
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: SernetSu_bf:13:f4 (58:76:ac:bf:13:f4), Dst: IntelCor_dc:86:ec (58:96:1d:dc:86:ec)
▼ Destination: IntelCor_dc:86:ec (58:96:1d:dc:86:ec)
Address: IntelCor_dc:86:ec (58:96:1d:dc:86:ec)
.... .0. = LG bit: Globally unique address (factory default)
.... .0. = IG bit: Individual address (unicast)
▼ Source: SernetSu_bf:13:f4 (58:76:ac:bf:13:f4)
Address: SernetSu_bf:13:f4 (58:76:ac:bf:13:f4)
.... .0. = LG bit: Globally unique address (factory default)
.... .0. = IG bit: Individual address (unicast)
Type: IPv4 (0x0800)

Рисунок 9. Эхо-ответ

Изучим кадры данных протокола ARP. Hardware type – это адрес канального уровня (Ethernet (1)), Protocol type – сетевой уровень (протокол IPv4), далее указаны размеры MAC-адреса (6 байт) и размер IPv4-адреса (4 байта). Код запроса – 1.

arp						
No.	Time	Source	Destination	Protocol	Length	Info
198	50.075300	SernetSu_bf:13:f4	Broadcast	ARP	60	Who has 192.16
199	50.075301	SernetSu_bf:13:f4	Broadcast	ARP	60	Who has 192.16
200	50.075301	SernetSu_bf:13:f4	Broadcast	ARP	60	Who has 192.16
201	50.075302	SernetSu_bf:13:f4	Broadcast	ARP	60	Who has 192.16
202	50.075302	SernetSu_bf:13:f4	Broadcast	ARP	60	Who has 192.16
203	50.075303	SernetSu_bf:13:f4	Broadcast	ARP	60	Who has 192.16
204	50.075303	SernetSu_bf:13:f4	Broadcast	ARP	60	Who has 192.16
205	50.075343	IntelCor_dc:86:ec	SernetSu_bf:13:f4	ARP	42	192.168.1.72 i

```

> Frame 204: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface \Device\NPF_{...}
```
Ethernet II, Src: SernetSu_bf:13:f4 (58:76:ac:bf:13:f4), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
 > Destination: Broadcast (ff:ff:ff:ff:ff:ff)
 > Source: SernetSu_bf:13:f4 (58:76:ac:bf:13:f4)
 Type: ARP (0x0806)
 Padding: 00
```
▼ Address Resolution Protocol (request)
  Hardware type: Ethernet (1)
  Protocol type: IPv4 (0x0800)
  Hardware size: 6
  Protocol size: 4
  Opcode: request (1)
  Sender MAC address: SernetSu_bf:13:f4 (58:76:ac:bf:13:f4)
  Sender IP address: 192.168.1.1
  Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
  Target IP address: 192.168.1.103
```

```

Рисунок 10. Протокол ARP

Изучим данные в полях заголовка Ethernet II.

Здесь указаны MAC-адреса источника и получателя. Получатель в нашем случае – широковещательный адрес (групповой и локально администрируемый). Источник – адрес нашего шлюза (индивидуальный и глобально администрируемый).

Начнем новый процесс захвата трафика в Wireshark. Пропингуем сайт Яндекса.

```
c:\WINDOWS\system32>ping www.yandex.ru

Обмен пакетами с www.yandex.ru [77.88.55.88] с 32 байтами данных:
Ответ от 77.88.55.88: число байт=32 время=9мс TTL=55

Статистика Ping для 77.88.55.88:
 Пакетов: отправлено = 4, получено = 4, потеряно = 0
 (% потерян)
Приблизительное время приема-передачи в мс:
 Минимальное = 9мсек, Максимальное = 9 мсек, Среднее = 9 мсек

c:\WINDOWS\system32>
```

Рисунок 11. Пингование сайта www.yandex.ru

Изучим запрос протокола ICMP. Адрес источника (Source, откуда запрос отправлен) – это адрес моего устройства (58-96-1D-DC-86-EC). Адрес получателя (destination, то куда отправлен запрос) - 58-76-AC-BF-13-F4. Что адрес источника, что адрес шлюза индивидуальные и глобально администрируемые.

| icmp |            |              |              |          |        |                           |
|------|------------|--------------|--------------|----------|--------|---------------------------|
| No.  | Time       | Source       | Destination  | Protocol | Length | Info                      |
| 378  | 98.309937  | 192.168.1.72 | 77.88.55.88  | ICMP     | 74     | Echo (ping) request id=0x |
| 379  | 98.319199  | 77.88.55.88  | 192.168.1.72 | ICMP     | 74     | Echo (ping) reply id=0x   |
| 390  | 99.315724  | 192.168.1.72 | 77.88.55.88  | ICMP     | 74     | Echo (ping) request id=0x |
| 391  | 99.325440  | 77.88.55.88  | 192.168.1.72 | ICMP     | 74     | Echo (ping) reply id=0x   |
| 404  | 100.327928 | 192.168.1.72 | 77.88.55.88  | ICMP     | 74     | Echo (ping) request id=0x |
| 405  | 100.337280 | 77.88.55.88  | 192.168.1.72 | ICMP     | 74     | Echo (ping) reply id=0x   |
| 412  | 101.331082 | 192.168.1.72 | 77.88.55.88  | ICMP     | 74     | Echo (ping) request id=0x |
| 413  | 101.340127 | 77.88.55.88  | 192.168.1.72 | ICMP     | 74     | Echo (ping) reply id=0x   |

> Frame 378: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface \Device\NPF\_{D13C22B:

  `- Ethernet II, Src: IntelCor\_dc:86:ec (58:96:1d:dc:86:ec), Dst: SernetSu\_bf:13:f4 (58:76:ac:bf:13:f4)

    `- Destination: SernetSu\_bf:13:f4 (58:76:ac:bf:13:f4)

    `- Source: IntelCor\_dc:86:ec (58:96:1d:dc:86:ec)

      `- Type: IPv4 (0x0800)

  `- Internet Protocol Version 4, Src: 192.168.1.72, Dst: 77.88.55.88

  `- Internet Control Message Protocol

Рисунок 12. Запрос протокола ICMP

Изучим запрос протокола ICMP. Адрес источника (Source, то куда откуда запрос отправлен) - 58-76-AC-BF-13-F4. Адрес получателя (Destination, то куда отправлен запрос) – это адрес моего устройства (58-96-1D-DC-86-EC). Что адрес источника, что адрес шлюза индивидуальные и глобально администрируемые.

| No. | Time       | Source       | Destination  | Protocol | Length | Info                |
|-----|------------|--------------|--------------|----------|--------|---------------------|
| 378 | 98.309937  | 192.168.1.72 | 77.88.55.88  | ICMP     | 74     | Echo (ping) request |
| 379 | 98.319199  | 77.88.55.88  | 192.168.1.72 | ICMP     | 74     | Echo (ping) reply   |
| 390 | 99.315724  | 192.168.1.72 | 77.88.55.88  | ICMP     | 74     | Echo (ping) request |
| 391 | 99.325440  | 77.88.55.88  | 192.168.1.72 | ICMP     | 74     | Echo (ping) reply   |
| 404 | 100.327928 | 192.168.1.72 | 77.88.55.88  | ICMP     | 74     | Echo (ping) request |
| 405 | 100.337280 | 77.88.55.88  | 192.168.1.72 | ICMP     | 74     | Echo (ping) reply   |
| 412 | 101.331082 | 192.168.1.72 | 77.88.55.88  | ICMP     | 74     | Echo (ping) request |
| 413 | 101.340127 | 77.88.55.88  | 192.168.1.72 | ICMP     | 74     | Echo (ping) reply   |

```
> Frame 379: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface \Device\NPF_{D1
< Ethernet II, Src: SernetSu_bf:13:f4 (58:76:ac:bf:13:f4), Dst: IntelCor_dc:86:ec (58:96:1d:dc:86:ec)
 > Destination: IntelCor_dc:86:ec (58:96:1d:dc:86:ec)
 > Source: SernetSu_bf:13:f4 (58:76:ac:bf:13:f4)
 Type: IPv4 (0x0800)
> Internet Protocol Version 4, Src: 77.88.55.88, Dst: 192.168.1.72
> Internet Control Message Protocol
```

Рисунок 13. Ответ протокола ICMP

### №3

В браузере перейдем на сайт, работающий по протоколу HTTP (например, на сайт CERN <http://info.cern.ch/>). Для получения большей информации для Wireshark поперемещались по ссылкам или разделам сайта в браузере.

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

Открываем раздел протокола TCP в случае запроса. Видим, что порт получателя – 80 (это стандартный порт для http). Порт источника – 50758 (он определяется случайным образом из незанятых и непrivилегированных портов). Также тут есть поле Порядковый номер (Sequence Number) и поле Номер подтверждения (Acknowledgment Number).

| No. | Time     | Source         | Destination    | Protocol | Length | Info                                   |
|-----|----------|----------------|----------------|----------|--------|----------------------------------------|
| 104 | 5.111717 | 192.168.1.72   | 188.185.22.9   | HTTP     | 579    | GET /www/hypertext/WWW/TheProject.html |
| 107 | 5.167733 | 188.185.22.9   | 192.168.1.72   | HTTP     | 191    | HTTP/1.1 302 Found                     |
| 172 | 8.147097 | 192.168.1.72   | 188.184.103.21 | HTTP     | 550    | GET /about HTTP/1.1                    |
| 176 | 8.201340 | 188.184.103.21 | 192.168.1.72   | HTTP     | 162    | HTTP/1.1 302 Found                     |

> Frame 104: 579 bytes on wire (4632 bits), 579 bytes captured (4632 bits) on interface \Device\NPF\_{D13C22B7-0000-0000-0000-000000000000  
> Ethernet II, Src: IntelCor\_dc:86:ec (58:96:1d:dc:86:ec), Dst: SernetSu\_bf:13:f4 (58:76:ac:bf:13:f4)  
> Internet Protocol Version 4, Src: 192.168.1.72, Dst: 188.185.22.9  
▼ Transmission Control Protocol, Src Port: 50758, Dst Port: 80, Seq: 1, Ack: 1, Len: 525  
    Source Port: 50758  
    Destination Port: 80  
    [Stream index: 4]  
    [Conversation completeness: Incomplete, DATA (15)]  
    [TCP Segment Len: 525]  
    Sequence Number: 1 (relative sequence number)  
    Sequence Number (raw): 3878750177  
    [Next Sequence Number: 526 (relative sequence number)]  
    Acknowledgment Number: 1 (relative ack number)  
    Acknowledgment number (raw): 3157049948  
    0101 .... = Header Length: 20 bytes (5)  
    > Flags: 0x018 (PSH, ACK)  
    Window: 513  
    [Calculated window size: 131328]  
    [Window size scaling factor: 256]  
    Checksum: 0xbcc5 [unverified]  
    [Checksum Status: Unverified]  
    Urgent Pointer: 0  
    > [Timestamps]  
    > [SEQ/ACK analysis]  
    TCP payload (525 bytes)

Рисунок 14. Протокол TCP (случай запроса)

Далее рассмотрим ответ. Здесь у нас поменялись местами порты источника и получателя. Теперь порт источника – порт сайта (80). А порт получателя – 50758 (выбранный случайным образом).

Frame 107: 191 bytes on wire (1528 bits), 191 bytes captured (1528 bits) on interface \Device\NPF\_{D13C22B7-88E0-4A9E-BE8F-000000000000  
 Ethernet II, Src: SernetSu\_bf:13:f4 (58:76:ac:bf:13:f4), Dst: IntelCor\_dc:86:ec (58:96:1d:dc:86:ec)  
 Internet Protocol Version 4, Src: 188.185.22.9, Dst: 192.168.1.72  
 Transmission Control Protocol, Src Port: 80, Dst Port: 50758, Seq: 1, Ack: 526, Len: 137  
 Source Port: 80  
 Destination Port: 50758  
 [Stream index: 4]  
 [Conversation completeness: Complete, WITH\_DATA (31)]  
 [TCP Segment Len: 137]  
 Sequence Number: 1 (relative sequence number)  
 Sequence Number (raw): 3157049948  
 [Next Sequence Number: 138 (relative sequence number)]  
 Acknowledgment Number: 526 (relative ack number)  
 Acknowledgment number (raw): 3878750702  
 0101 .... = Header Length: 20 bytes (5)  
 Flags: 0x018 (PSH, ACK)

Рисунок 15. Протокол TCP (случай ответа)

В Wireshark в строке фильтра укажем dns и проанализируем информацию по протоколу UDP в случае запросов и ответов.

В случае запроса: порт источника – 58034 (выбранный случайным образом из незанятых и непривилегированных портов). Порт получателя – 53.

Frame 93: 77 bytes on wire (616 bits), 77 bytes captured (616 bits) on interface \Device\NPF\_{D13C22B7-88E0-4A9E-BE8F-000000000000  
 Ethernet II, Src: IntelCor\_dc:86:ec (58:96:1d:dc:86:ec), Dst: SernetSu\_bf:13:f4 (58:76:ac:bf:13:f4)  
 Internet Protocol Version 4, Src: 192.168.1.72, Dst: 192.168.1.1  
 User Datagram Protocol, Src Port: 58034, Dst Port: 53  
 Source Port: 58034  
 Destination Port: 53  
 Length: 43  
 Checksum: 0xd93b [unverified]  
 [Checksum Status: Unverified]  
 [Stream index: 11]  
 [Timestamps]  
 UDP payload (35 bytes)  
 Domain Name System (query)

Рисунок 16. Протокол UDP (случай запроса)

В случае ответа порт источника – 53, а порт получателя – 58034.

| No. | Time     | Source       | Destination  | Protocol | Length | Info             |
|-----|----------|--------------|--------------|----------|--------|------------------|
| 88  | 3.989374 | 192.168.1.1  | 192.168.1.72 | DNS      | 155    | Standard query r |
| 89  | 3.989515 | 192.168.1.1  | 192.168.1.72 | DNS      | 155    | Standard query r |
| 90  | 4.733025 | 192.168.1.72 | 192.168.1.1  | DNS      | 77     | Standard query & |
| 91  | 4.733409 | 192.168.1.72 | 192.168.1.1  | DNS      | 77     | Standard query & |
| 92  | 4.839409 | 192.168.1.1  | 192.168.1.72 | DNS      | 157    | Standard query r |
| 93  | 4.847326 | 192.168.1.72 | 192.168.1.1  | DNS      | 77     | Standard query & |
| 94  | 4.891097 | 192.168.1.1  | 192.168.1.72 | DNS      | 168    | Standard query r |
| 95  | 4.891349 | 192.168.1.1  | 192.168.1.72 | DNS      | 168    | Standard query r |
| 108 | 5.171493 | 192.168.1.72 | 192.168.1.1  | DNS      | 77     | Standard query & |

> Frame 95: 168 bytes on wire (1344 bits), 168 bytes captured (1344 bits) on interface \Device  
> Ethernet II, Src: SernetSu\_bf:13:f4 (58:76:ac:bf:13:f4), Dst: IntelCor\_dc:86:ec (58:96:1d:dc  
> Internet Protocol Version 4, Src: 192.168.1.1, Dst: 192.168.1.72  
▼ User Datagram Protocol, Src Port: 53, Dst Port: 58034  
    Source Port: 53  
    Destination Port: 58034  
    Length: 134  
    Checksum: 0x45fa [unverified]  
    [Checksum Status: Unverified]  
    [Stream index: 11]  
    > [Timestamps]  
    UDP payload (126 bytes)  
    > Domain Name System (response)

Рисунок 17. Протокол UDP (случай ответа)

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

Как и в случае dns можем посмотреть информацию транспортного уровня по протоколу UDP. Порт источника задан случайно, выбором из непривилегированных и незанятых портов, и равен 64952, порт получателя равен 443 - это стандартный порт HTTPS, то есть quic сразу криптуется.

Для создания альтернативы TCP поверх UDP строятся протоколы прикладного уровня QUIC IETF, которые управляют трафиком, управляют качеством обслуживания.

| quic |          |                |                |          |        |                        |  |
|------|----------|----------------|----------------|----------|--------|------------------------|--|
| No.  | Time     | Source         | Destination    | Protocol | Length | Info                   |  |
| 12   | 2.441944 | 192.168.1.72   | 142.250.74.131 | QUIC     | 1292   | Initial, DCID=2604f68  |  |
| 13   | 2.444788 | 192.168.1.72   | 142.250.74.131 | QUIC     | 119    | 0-RTT, DCID=2604f6824  |  |
| 14   | 2.464871 | 142.250.74.131 | 192.168.1.72   | QUIC     | 1292   | Initial, SCID=e604f68  |  |
| 15   | 2.467843 | 192.168.1.72   | 142.250.74.131 | QUIC     | 1292   | Initial, DCID=e604f68  |  |
| 16   | 2.471307 | 142.250.74.131 | 192.168.1.72   | QUIC     | 1292   | Handshake, SCID=e604f  |  |
| 17   | 2.471310 | 142.250.74.131 | 192.168.1.72   | QUIC     | 1292   | Handshake, SCID=e604f  |  |
| 18   | 2.471311 | 142.250.74.131 | 192.168.1.72   | QUIC     | 1292   | Handshake, SCID=e604f  |  |
| 19   | 2.471312 | 142.250.74.131 | 192.168.1.72   | QUIC     | 242    | Protected Payload (KP) |  |
| 20   | 2.471545 | 192.168.1.72   | 142.250.74.131 | QUIC     | 81     | Handshake, DCID=e604f  |  |

> Frame 12: 1292 bytes on wire (10336 bits), 1292 bytes captured (10336 bits) on interface \Device\NPF\_{...}

> Ethernet II, Src: IntelCor\_dc:86:ec (58:96:1d:dc:86:ec), Dst: SernetSu\_bf:13:f4 (58:76:ac:bf:13:f4)

> Internet Protocol Version 4, Src: 192.168.1.72, Dst: 142.250.74.131

>User Datagram Protocol, Src Port: 64952, Dst Port: 443

    Source Port: 64952  
    Destination Port: 443  
    Length: 1258  
    Checksum: 0xe3f0 [unverified]  
    [Checksum Status: Unverified]  
    [Stream index: 3]  
    > [Timestamps]  
    UDP payload (1250 bytes)

    QUIC IETF

        QUIC Connection information  
            [Connection Number: 0]  
            [Packet Length: 1250]  
            1... .... = Header Form: Long Header (1)  
            .1... .... = Fixed Bit: True  
            ..00 .... = Packet Type: Initial (0)  
            .... 00.. = Reserved: 0  
            .... ..00 = Packet Number Length: 1 bytes (0)  
            Version: 1 (0x00000001)  
            Destination Connection ID Length: 8  
            Destination Connection ID: 2604f6824825e9cb  
            Source Connection ID Length: 0  
            Token Length: 70  
            Token: 004630e0538a5bdbca09c1fec23d67ffb83eb6e46d704e12e134463c0b3a5aed1cff2023...  
            Length: 1161  
            Packet Number: 1  
            Payload: df06381fab53acd6636b311f8881fc04761bc282b9fc6735746908f8dce3f7db2bc4619a...

Рисунок 18. Запрос quic

В случае ответа порты заданы наоборот, то есть источник - 443 порт, получатель – 64952.

| No. | Time     | Source         | Destination    | Protocol | Length | Info       |
|-----|----------|----------------|----------------|----------|--------|------------|
| 28  | 2.518941 | 142.250.74.131 | 192.168.1.72   | QUIC     | 66     | Protected  |
| 29  | 2.520401 | 192.168.1.72   | 142.250.74.131 | QUIC     | 74     | Protected  |
| 30  | 2.521085 | 142.250.74.131 | 192.168.1.72   | QUIC     | 772    | Protected  |
| 31  | 2.521284 | 192.168.1.72   | 142.250.74.131 | QUIC     | 77     | Protected  |
| 32  | 2.523147 | 142.250.74.131 | 192.168.1.72   | QUIC     | 354    | Protected  |
| 33  | 2.523147 | 142.250.74.131 | 192.168.1.72   | QUIC     | 173    | Protected  |
| 34  | 2.523311 | 192.168.1.72   | 142.250.74.131 | QUIC     | 73     | Protected  |
| 52  | 2.567630 | 142.250.74.131 | 192.168.1.72   | QUIC     | 66     | Protected  |
| 243 | 8.715425 | 192.168.1.72   | 64.233.165.198 | QUIC     | 1292   | Initial, D |

```

> Frame 52: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface \Device\NPF_{...}
> Ethernet II, Src: SernetSu_bf:13:f4 (58:76:ac:bf:13:f4), Dst: IntelCor_dc:86:ec (58:96:...
> Internet Protocol Version 4, Src: 142.250.74.131, Dst: 192.168.1.72
▼ User Datagram Protocol, Src Port: 443, Dst Port: 64952
 Source Port: 443
 Destination Port: 64952
 Length: 32
 Checksum: 0xb8fc [unverified]
 [Checksum Status: Unverified]
 [Stream index: 3]
 > [Timestamps]
 UDP payload (24 bytes)
▼ QUIC IETF
 ▼ QUIC Connection information
 [Connection Number: 0]
 [Packet Length: 24]
 ▼ QUIC Short Header
 0.... = Header Form: Short Header (0)
 .1.. = Fixed Bit: True
 ..0. = Spin Bit: False
 Remaining Payload: ce724ae93ba9462647a982ea9144d3cc1d8053b1706323

```

Рисунок 19. Ответ quic

#### №4

На моем устройстве используем соединение по HTTP с каким-то сайтом для захвата в Wireshark пакетов TCP.

Проанализируем handshake протокола TCP.

|              |                |                |     |                                                                                |
|--------------|----------------|----------------|-----|--------------------------------------------------------------------------------|
| 126 8.469032 | 192.168.1.72   | 188.185.35.172 | TCP | 66 50167 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM            |
| 127 8.560337 | 192.168.1.72   | 142.250.74.174 | TCP | 571 [TCP Retransmission] 50164 → 443 [PSH, ACK] Seq=1 Ack=1 Win=131072 Len=517 |
| 128 8.560348 | 192.168.1.72   | 142.250.74.174 | TCP | 571 [TCP Retransmission] 50160 → 443 [PSH, ACK] Seq=1 Ack=1 Win=131072 Len=517 |
| 129 8.560364 | 192.168.1.72   | 142.250.74.174 | TCP | 571 [TCP Retransmission] 50161 → 443 [PSH, ACK] Seq=1 Ack=1 Win=131072 Len=517 |
| 130 8.560379 | 192.168.1.72   | 142.250.74.174 | TCP | 571 [TCP Retransmission] 50162 → 443 [PSH, ACK] Seq=1 Ack=1 Win=131072 Len=517 |
| 131 8.560381 | 192.168.1.72   | 142.250.74.174 | TCP | 571 [TCP Retransmission] 50163 → 443 [PSH, ACK] Seq=1 Ack=1 Win=131072 Len=517 |
| 132 8.888024 | 188.185.35.172 | 192.168.1.72   | TCP | 66 80 → 50167 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM WS=128 |
| 133 8.888129 | 192.168.1.72   | 188.185.35.172 | TCP | 54 50167 → 80 [ACK] Seq=1 Ack=1 Win=131328 Len=0                               |

Рисунок 20. Handshake TCP

Режим активного доступа (Active Open). Клиент посыпает сообщение SYN, ISSa, т.е. в передаваемом сообщении установлен бит SYN (Synchronize Sequence Number), а в поле Порядковый номер (Sequence Number) — начальное 32-битное значение ISSa (Initial Sequence Number).

Значение Sequence Number равно 848625695 (ISSa), значение Acknowledgment Number равно 0. Также видим, что установлен флаг SYN.

```
✓ Transmission Control Protocol, Src Port: 50167, Dst Port: 80, Seq: 0, Len: 0
 Source Port: 50167
 Destination Port: 80
 [Stream index: 15]
 [Conversation completeness: Incomplete, ESTABLISHED (7)]
 [TCP Segment Len: 0]
 Sequence Number: 0 (relative sequence number)
 Sequence Number (raw): 848625695
 [Next Sequence Number: 1 (relative sequence number)]
 Acknowledgment Number: 0
 Acknowledgment number (raw): 0
 1000 = Header Length: 32 bytes (8)
 ✓ Flags: 0x002 (SYN)
 000. = Reserved: Not set
 ...0 = Accurate ECN: Not set
 0.... = Congestion Window Reduced: Not set
 0.. = ECN-Echo: Not set
 0. = Urgent: Not set
 0 = Acknowledgment: Not set
 0... = Push: Not set
 0.. = Reset: Not set
 ✓1. = Syn: Set
```

Рисунок 21. Протокол TCP для первой ступени handshake

Режим пассивного доступа (Passive Open). Сервер откликается, посылая сообщение SYN, ACK, ISSb, ACK(ISSa+1), т.е. установлены биты SYN и ACK; в поле Порядковый номер (Sequence Number) хостом В устанавливается начальное значение счётчика — ISSb; поле Номер подтверждения (Acknowledgment Number) содержит значение ISSa, полученное в первом пакете от хоста А и увеличенное на единицу.

Действительно, Acknowledgment Number равно 848625695 (значение ISSa) + 1 = 848625696. Sequence Number равен 2149321476 (начальное значение счётчика — ISSb).

```
▼ Transmission Control Protocol, Src Port: 80, Dst Port: 50167, Seq: 0, Ack: 1, Len: 0
 Source Port: 80
 Destination Port: 50167
 [Stream index: 15]
 [Conversation completeness: Incomplete, ESTABLISHED (7)]
 [TCP Segment Len: 0]
 Sequence Number: 0 (relative sequence number)
 Sequence Number (raw): 2149321476
 [Next Sequence Number: 1 (relative sequence number)]
 Acknowledgment Number: 1 (relative ack number)
 Acknowledgment number (raw): 848625696
 1000 = Header Length: 32 bytes (8)
 ▼ Flags: 0x012 (SYN, ACK)
```

Рисунок 22. Протокол TCP для второй ступени handshake

Установлены флаги SYN, ACK.

Клиент отправляет подтверждение получения SYN сегмента от сервера с идентификатором, равным ISN (сервера)+1: ACK, ISSa+1, ACK(ISSb+1). В этом пакете установлен битACK, поле Порядковый номер (Sequence Number) содержит ISSa+1, поле Номер подтверждения (Acknowledgment Number) содержит значение ISSb+1. Посылкой этого пакета заканчивается трёхступенчатый handshake, и TCP-соединение считается установленным.

Действительно, Acknowledgment Number равно 2149321476 (значение ISSb) + 1 = 2149321477. Sequence Number равен 848625695 (ISSa) + 1 = 848625696. Установлен флаг ACK.

```
▼ Transmission Control Protocol, Src Port: 50167, Dst Port: 80, Seq: 1, Ack: 1, Len: 0
 Source Port: 50167
 Destination Port: 80
 [Stream index: 15]
 [Conversation completeness: Incomplete, ESTABLISHED (7)]
 [TCP Segment Len: 0]
 Sequence Number: 1 (relative sequence number)
 Sequence Number (raw): 848625696
 [Next Sequence Number: 1 (relative sequence number)]
 Acknowledgment Number: 1 (relative ack number)
 Acknowledgment number (raw): 2149321477
 0101 = Header Length: 20 bytes (5)
 ▼ Flags: 0x010 (ACK)
```

Рисунок 23. Протокол TCP для третьей ступени handshake

Далее посмотрим график потока. Здесь в принципе все то же самое, что мы уже разобрали, только на графике. Клиент посыпает запрос серверу (установлен бит SYN), Seq = 0. Далее сервер отвечает клиенту (установлены биты SYN, ACK), Seq = 0, Ack = 1 (это относительные значения). Ну зврение рукопожатия: клиент отправляет серверу подтверждение получения SYN сегмента, Seq = 1, Ack = 1.

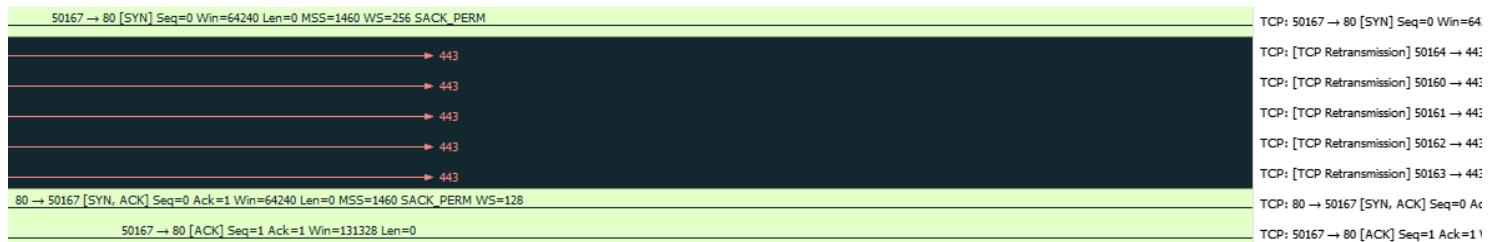


Рисунок 24. График потока

## Выводы

В процессе выполнения данной лабораторной работы я изучил посредством Wireshark кадры Ethernet, проанализировала PDU протоколы транспортного и прикладного уровней стека TCP/IP.