

Лабораторная работа №3

Отчет: Анализ трафика в Wireshark

Лисовская Арина Валерьевна

Содержание

1 Цель работы	4
2 Задание	5
3 Выполнение лабораторной работы	6
3.1 Анализ сетевых интерфейсов	6
3.2 Работа с протоколами ICMP и ARP	8
3.3 Исследование протоколов TCP и HTTP	10
4 Выводы	14
5 Ответы на контрольные вопросы	15

Список иллюстраций

3.1	Начало работы с терминалом	6
3.2	Вывод команды ipconfig /all	7
3.3	Вывод команды GETMAC	7
3.4	Сведения о сетевом подключении в GUI	8
3.5	Пинг шлюза по умолчанию	8
3.6	Запуск захвата трафика	8
3.7	Анализ ICMP-пакета в Wireshark	9
3.8	Настройка фильтров отображения	9
3.9	Пинг доменаrudn.ru	9
3.10	Статистика выполнения ping	10
3.11	Захват ARP-пакетов в Wireshark	10
3.12	Веб-страница info.cern.ch	11
3.13	Обнаружение HTTP-трафика	11
3.14	Анализ HTTP-ответа	12
3.15	Детали TCP-сегмента	12
3.16	Анализ TCP Handshake	12
3.17	График потока в Wireshark	13
3.18	Завершение работы и общая статистика	13

1 Цель работы

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

2 Задание

1. Изучить сетевые настройки ОС с помощью консольных утилит `ipconfig` и `getmac`.
2. Проанализировать работу протоколов ICMP и ARP при помощи захвата трафика в Wireshark.
3. Исследовать структуру HTTP-трафика при обращении к веб-ресурсам.
4. Провести детальный анализ процесса установления соединения (handshake) протокола TCP.

3 Выполнение лабораторной работы

3.1 Анализ сетевых интерфейсов

Для начала работы необходимо определить текущие параметры сетевого подключения. Запускаю терминал и ввожу команду для вывода общей информации об интерфейсах (рис. 3.1):

```
PS C:\Users\Арина> ipconfig

Настройка протокола IP для Windows

Адаптер Ethernet Ethernet 2:

    DHCP-суффикс подключения . . . . . :
    Локальный IPv6-адрес канала . . . . . : fe80::154b:8e54:709f:48cf
%3
    IPv4-адрес . . . . . : 192.168.56.1
    Маска подсети . . . . . : 255.255.255.0
    Основной шлюз. . . . . :

Адаптер беспроводной локальной сети Подключение по локальной сети* 1:

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

Адаптер беспроводной локальной сети Подключение по локальной сети* 2:

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

Рис. 3.1: Начало работы с терминалом

Более подробную информацию о конфигурации, включая адреса DNS-серверов и статус DHCP, получаю с помощью команды `ipconfig /all` (рис. 3.2):

Рис. 3.2: Вывод команды ipconfig /all

Далее перехожу к определению физических адресов сетевых адаптеров. Для этого использую специализированную команду `getmac` (рис. 3.3):

```
S C:\WINDOWS\system32> GETMAC

Физический адрес      Имя транспорта
=====
0-F6-0A-33-73-8B  \Device\Tcpip_{D4AE9452-96DB-4762-9A46-8F3DDDAC91D2}
A-00-27-00-00-03  \Device\Tcpip_{2CEC490E-8881-4A89-8358-1B80945FC152}
0-FF-2F-C5-AC-F9  Носитель отключен
S C:\WINDOWS\system32> -
```

Рис. 3.3: Вывод команды GETMAC

Для верификации данных открываю графический интерфейс настроек Windows, где сопоставляю полученные ранее MAC-адреса и параметры IPv4 (рис. 3.4):

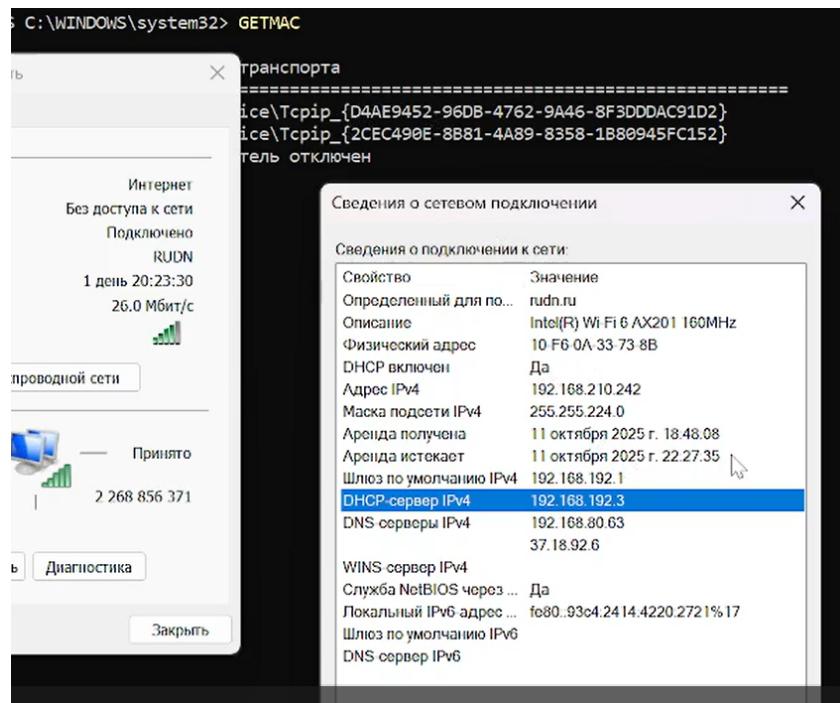


Рис. 3.4: Сведения о сетевом подключении в GUI

3.2 Работа с протоколами ICMP и ARP

Приступаю к генерации сетевого трафика для последующего анализа. Проверяю доступность шлюза по умолчанию с помощью утилиты ping (рис. 3.5):

Пинг шлюза по умолчанию

Рис. 3.5: Пинг шлюза по умолчанию

Запускаю процесс захвата в Wireshark и анализирую структуру сформированных ICMP-пакетов (рис. 3.6):

Запуск захвата трафика

Рис. 3.6: Запуск захвата трафика

В окне Wireshark детально рассматриваю кадр Ethernet II и вложенный в него ICMP эхо-запрос (рис. 3.7):

```

Основной шлюз. . . . . : 192.168.192.1
PS C:\Users\Арина> ping 192.168.192.1

Обмен пакетами с 192.168.192.1 по с 32 байтами данных:
Ответ от 192.168.192.1: число байт=32 время=2мс TTL=254
Ответ от 192.168.192.1: число байт=32 время=7мс TTL=254
Ответ от 192.168.192.1: число байт=32 время=2мс TTL=254
Ответ от 192.168.192.1: число байт=32 время=18мс TTL=254

Статистика Ping для 192.168.192.1:
    Пакетов: отправлено = 4, получено = 4, потеряно = 0
        (0% потеря)
Приблизительное время приема–передачи в мс:
    Минимальное = 2 мсек  Максимальное = 18 мсек Среднее = 7 мсек

```

Рис. 3.7: Анализ ICMP-пакета в Wireshark

Для удобства анализа применяю фильтрацию, оставляя только интересующие нас протоколы сетевого уровня (рис. 3.8):

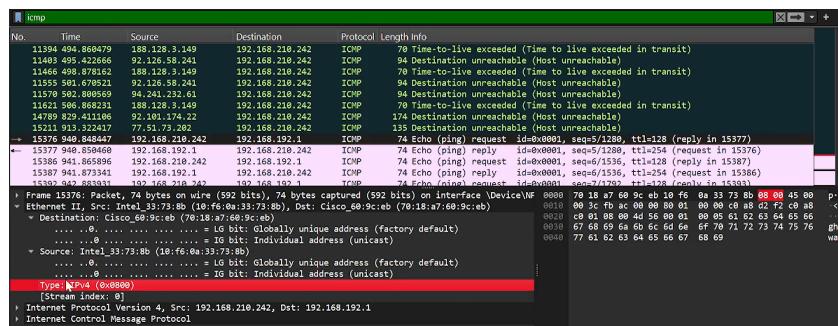


Рис. 3.8: Настройка фильтров отображения

Выполняю проверку связи с внешним доменомrudn.ru для отслеживания процесса разрешения имен и передачи пакетов через внешние шлюзы (рис. 3.9):

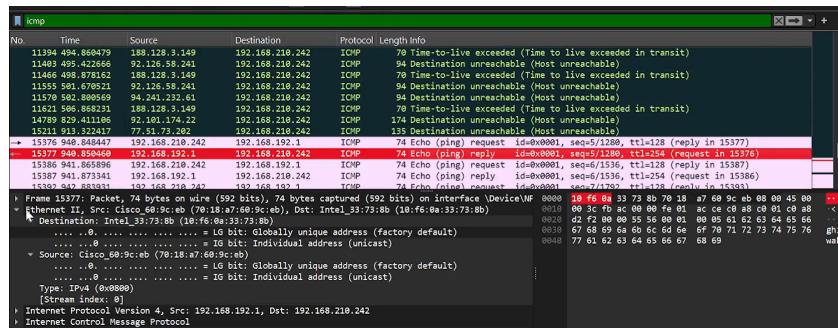


Рис. 3.9: Пинг домена rudn.ru

После завершения обмена пакетами изучаю итоговую статистику по потерям и времени задержки в консоли (рис. 3.10):

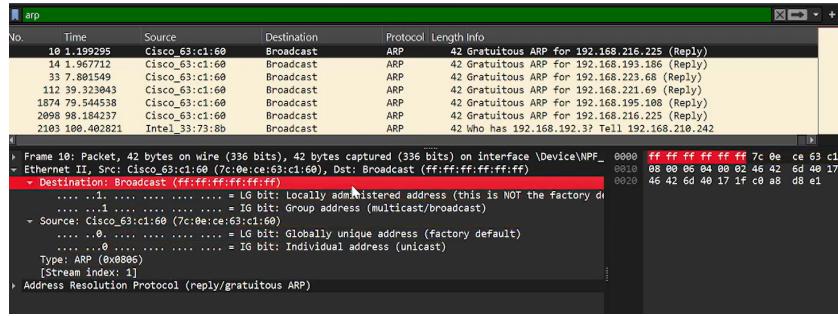


Рис. 3.10: Статистика выполнения ping

Анализирую захваченные ARP-сообщения, в частности Gratuitous ARP, которые используются для оповещения сети о MAC-адресе устройства (рис. 3.11):

```
PS C:\Users\Арина> ping rudn.ru

Обмен пакетами с rudn.ru [192.168.80.63] с 32 байтами данных:
Ответ от 192.168.80.63: число байт=32 время=20мс TTL=127
Ответ от 192.168.80.63: число байт=32 время=3мс TTL=127
Ответ от 192.168.80.63: число байт=32 время=10мс TTL=127
Ответ от 192.168.80.63: число байт=32 время=14мс TTL=127

Статистика Ping для 192.168.80.63:
Пакетов: отправлено = 4, получено = 4, потеряно = 0
(0% потеря)
Приблизительное время приема-передачи в мс:
```

Рис. 3.11: Захват ARP-пакетов в Wireshark

3.3 Исследование протоколов TCP и HTTP

Перехожу к анализу трафика прикладного уровня. Для этого открываю в браузере демонстрационный сайт, работающий по протоколу HTTP (рис. 3.12):

No.	Time	Source	Destination	Protocol	Length Info
210	2.546276	Cisco_63:c1:60	Broadcast	ARP	42 Gratuitous ARP
395	6.824311	Cisco_63:c1:60	Broadcast	ARP	42 Gratuitous ARP
747	21.873208	Cisco_63:c1:60	Broadcast	ARP	42 Gratuitous ARP
779	27.848223	Intel_33:73:8b	Cisco_60:9c:eb	ARP	42 Who has
780	27.850702	Cisco_60:9c:eb	Intel_33:73:8b	ARP	60 192.168.1.16
848	45.267253	Cisco_63:c1:60	Broadcast	ARP	42 Gratuitous ARP

Frame 210: Packet, 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface "eth0", IEEE 802.3 Ethernet II (ethernet), Src: Cisco_63:c1:60 (7c:0e:ce:63:c1:60), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
 Destination: Broadcast (ff:ff:ff:ff:ff:ff)
1.... = LG bit: Locally administered address
1.... = IG bit: Group address (multicast)
 Source: Cisco_63:c1:60 (7c:0e:ce:63:c1:60)
0.... = LG bit: Globally unique address
0.... = IG bit: Individual address (unicast)
 Type: ARP (0x0806)
 [Stream index: 1]
 Address Resolution Protocol (reply/gratuitous ARP)

Рис. 3.12: Веб-страница info.cern.ch

В Wireshark нахожу соответствующий HTTP GET запрос, инициированный браузером (рис. 3.13):

No.	Time	Source	Destination	Protocol	Length Info
55	0.675036	176.213.92.23	192.168.210.242	ICMP	70 Destination ICMP
210	2.546276	Cisco_63:c1:60	Broadcast	ARP	42 Gratuitous ARP
212	2.769828	212.129.76.174	192.168.210.242	ICMP	70 Destination ICMP
325	4.923619	94.181.132.25	192.168.210.242	ICMP	70 Destination ICMP
336	5.012431	46.158.242.41	192.168.210.242	ICMP	94 Destination ICMP
395	6.824311	Cisco_63:c1:60	Broadcast	ARP	42 Gratuitous ARP
428	7.678505	176.213.92.23	192.168.210.242	ICMP	70 Destination ICMP

Frame 210: Packet, 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface "eth0", IEEE 802.3 Ethernet II (ethernet), Src: Cisco_63:c1:60 (7c:0e:ce:63:c1:60), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
 Destination: Broadcast (ff:ff:ff:ff:ff:ff)
1.... = LG bit: Locally administered address
1.... = IG bit: Group address (multicast)
 Source: Cisco_63:c1:60 (7c:0e:ce:63:c1:60)
0.... = LG bit: Globally unique address
0.... = IG bit: Individual address (unicast)
 Type: ARP (0x0806)
 [Stream index: 1]
 Address Resolution Protocol (reply/gratuitous ARP)

Рис. 3.13: Обнаружение HTTP-трафика

Изучаю структуру HTTP-ответа от сервера, обращая внимание на коды состояния и заголовки (рис. 3.14):

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. 3.14: Анализ HTTP-ответа

Для понимания того, как данные передаются на транспортном уровне, анализирую содержимое TCP-сегмента (рис. 3.15):

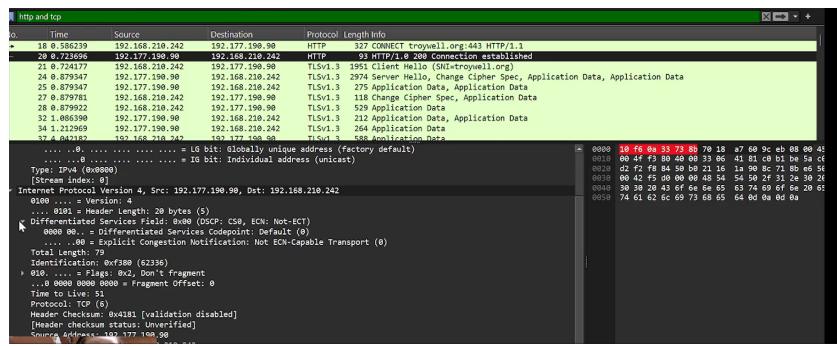


Рис. 3.15: Детали TCP-сегмента

Особое внимание уделяю процессу «трехэтапного рукопожатия» (SYN, SYN-ACK, ACK) перед началом передачи данных (рис. 3.16):

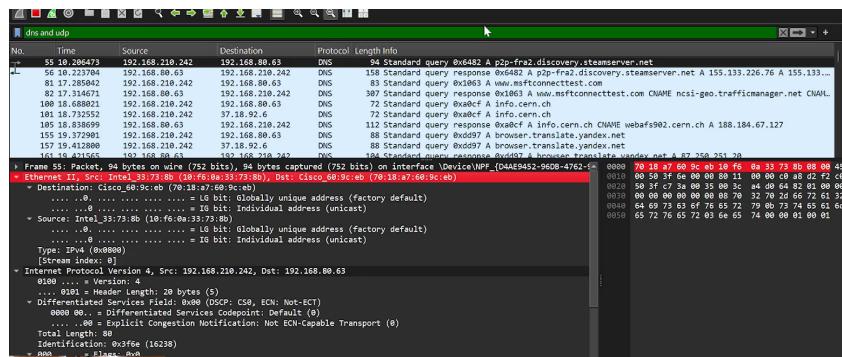


Рис. 3.16: Анализ TCP Handshake

Использую встроенный инструмент «График потока» (Flow Graph) для визуализации обмена сообщениями между клиентом и сервером (рис. 3.17):

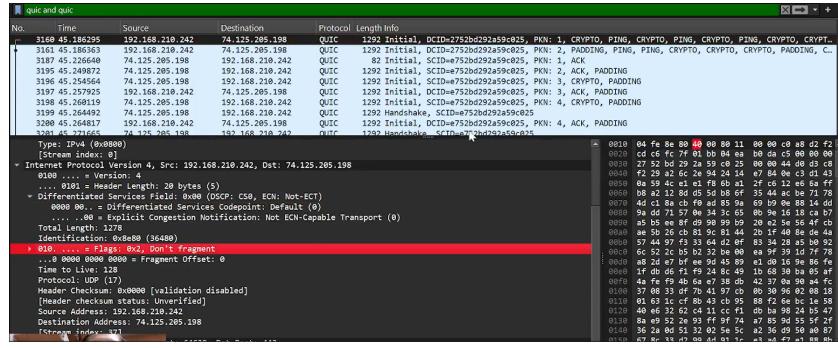


Рис. 3.17: График потока в Wireshark

Завершаю захват трафика и подвожу итоги по собранным данным в статистическом модуле (рис. 3.18):

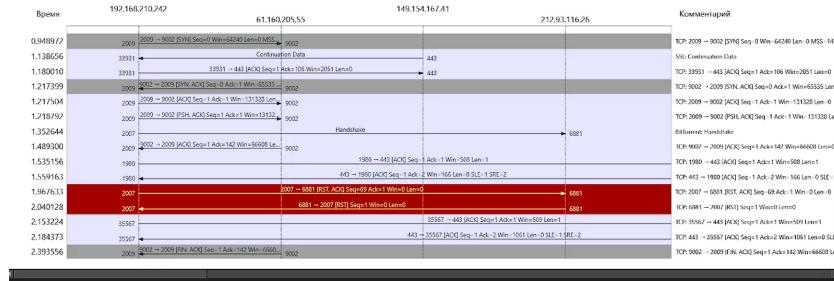


Рис. 3.18: Завершение работы и общая статистика

4 Выводы

В ходе выполнения лабораторной работы я освоил практические навыки работы с анализатором сетевого трафика Wireshark. Мною были изучены структуры кадров Ethernet, пакетов ICMP и ARP, а также механизмы работы транспортного протокола TCP и прикладного протокола HTTP. Я научился идентифицировать MAC-адреса, анализировать процесс установления TCP-соединения и использовать фильтры Wireshark для эффективного поиска сетевых событий.

5 Ответы на контрольные вопросы

1. Как называются PDU для 2, 3 и 4 уровней модели OSI?

- 2 уровень (Канальный): Кадр (Frame).
- 3 уровень (Сетевой): Пакет (Packet).
- 4 уровень (Транспортный): Сегмент (Segment) для TCP или Датаграмма (Datagram) для UDP.

2. Что такое OUI в MAC-адресе?

- OUI (Organizationally Unique Identifier) — это первые три байта (24 бита) MAC-адреса, которые однозначно идентифицируют производителя сетевого оборудования.

3. Какие основные флаги используются в TCP-заголовке для управления соединением?

- SYN (Synchronize): запрос на установление соединения.
- ACK (Acknowledgment): подтверждение получения.
- FIN (Finish): запрос на завершение соединения.
- RST (Reset): обрыв соединения при ошибке.
- PSH (Push): немедленная передача данных приложению.

4. В чем различие между утилитами Ping и Tracert?

- ping проверяет наличие связи с узлом и измеряет задержку.
- tracert (traceroute) показывает весь маршрут следования пакета, перечисляя все промежуточные узлы (маршрутизаторы), через которые он проходит.