Университет ИТМО

Факультет программной инженерии и компьютерной техники

Лабораторная работа №4

«Анализ трафика компьютерных сетей с помощью утилиты Wireshark»

по дисциплине “Компьютерные сети”

Выполнил:

Студент группы P3334

Галиуллин Р.Д.

Преподаватель:

Алиев Т. И.

Санкт-Петербург

2025 г.

ОГЛАВЛЕНИЕ

ЗАДАНИЕ

Описание лабораторной работы

Цель работы – изучить структуру протокольных блоков данных, анализируя реальный трафик на компьютере студента с помощью бесплатно распространяемой утилиты Wireshark.

В процессе выполнения домашнего задания выполняются наблюдения за передаваемым трафиком с компьютера пользователя в Интернет и в обратном направлении. Применение специализированной утилиты Wireshark позволяет наблюдать структуру передаваемых кадров, пакетов и сегментов данных различных сетевых протоколов. При выполнении УИР рекомендуется выполнить анализ последовательности команд и определить назначение служебных данных, используемых для организации обмена данными в протоколах: ARP, DNS, FTP, HTTP, DHCP.

Вариант для выполнения

Сайт для анализа трафика - rgd.ca

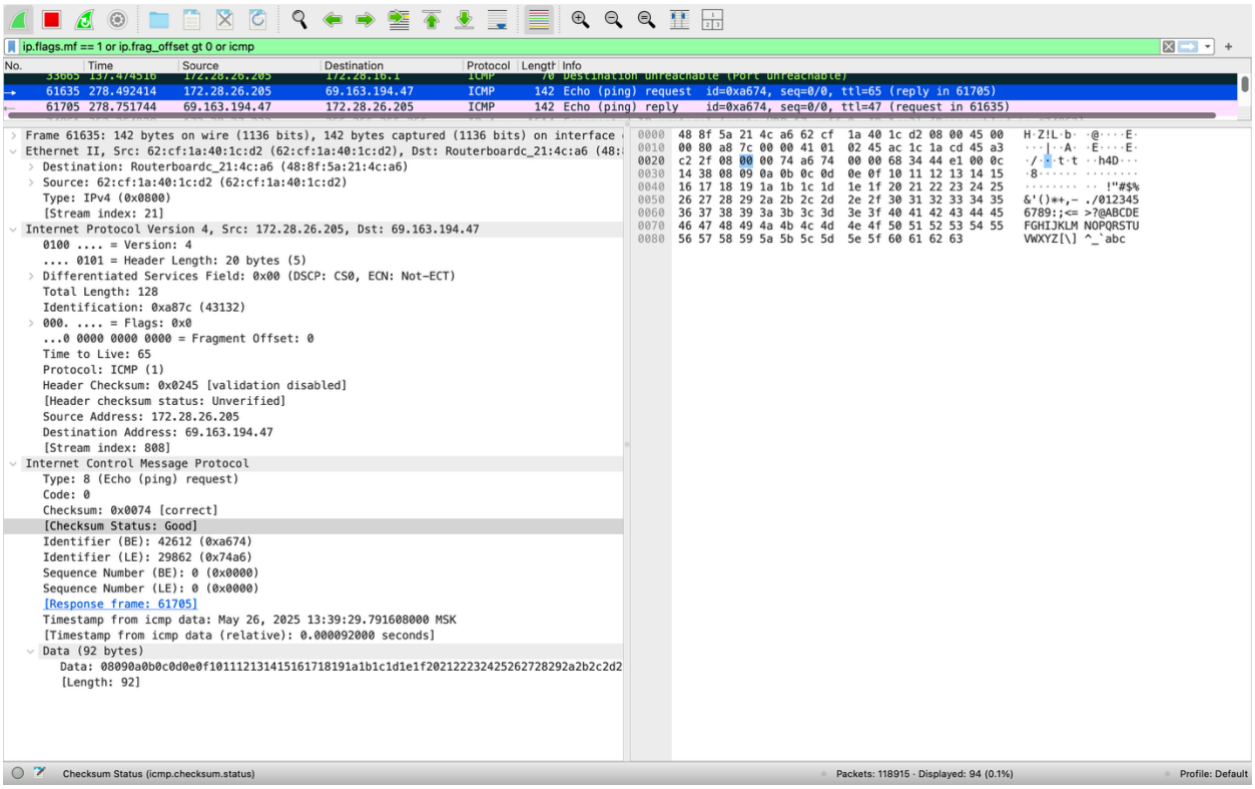
4.1. АНАЛИЗ ТРАФИКА УТИЛИТЫ PING

Запускаем из терминала (MacOS) команду для отправки пакетов на сайт rgd.ca:

ping -s “размер_пакета” -c “количество пакетов” rgd.ca

Чтобы отправлять один пакет, а не 4, как происходит по умолчанию, будем использовать флаг -n. Отправим пакеты размерами 100, 200, 500, 1000, 2000, 5000, 10000 байт.

Утилита управляет ICMP запросами и ответами.



Структура пакета:

Ethernet 2 (канальный уровень)

Заголовок содержит:

- Destination– MAC адрес получателя
- Source – MAC-адрес отправителя
- Type – тип протокола

IP-заголовок (сетевой уровень)

Заголовок содержит:

- Version
- Header Length
- Identification – идентификатор фрагмента
- Protocol – тип вложенного протокола
- Flags – указывается DF и MF

- TTL – ограничение на кол-во хопов

- Fragment offset – смещение фрагмента (если пакет был фрагментирован)

- Header Checksum – контрольная сумма заголовка

- Source IP address

- Destination IP address

ICMP (сетевой протокол)

Заголовок содержит:

- Type – request или reply

- Checksum - CRC

- Identifier – уникальный ID запроса

- Seq number – номер последовательности запроса

Data (поле данных)

Ответы на контрольные вопросы

1. Имеет ли место фрагментация исходного пакета, какое поле на это указывает?

2. Какая информация указывает, является ли фрагмент пакета последним или промежуточным?

Фрагментация пакетов — это процесс разделения большого сетевого пакета на более мелкие части (фрагменты), чтобы они могли быть переданы через сеть с ограничениями на максимальный размер передаваемого блока данных (MTU, Maximum Transmission Unit). Этот механизм реализуется на сетевом уровне (IP) и обеспечивает совместимость между устройствами и сетями с разными MTU.

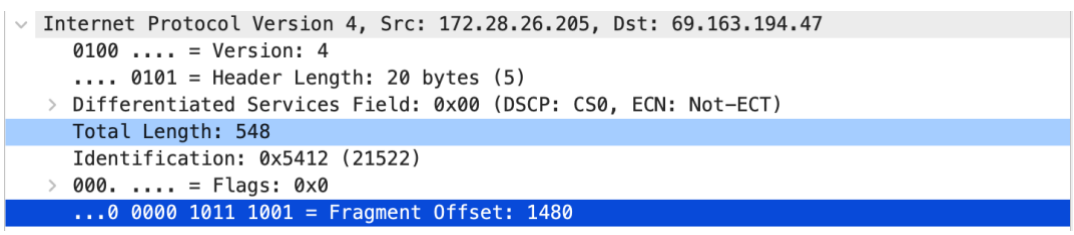
Да, имеет, на это указывают:

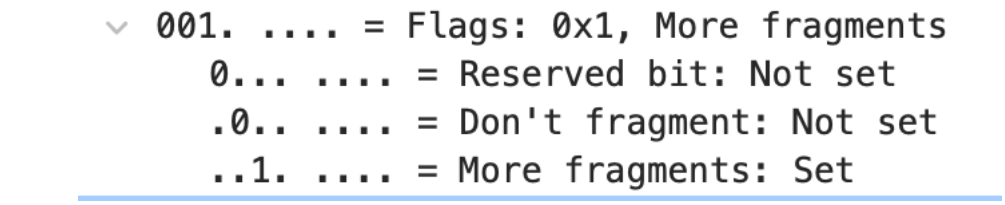
- флаг фрагментации More Fragments в IP-заголовке:

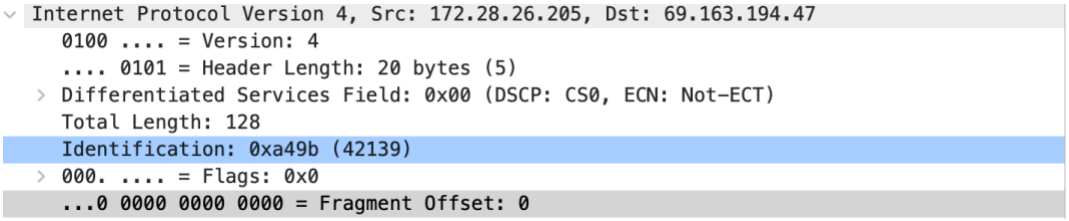
  MF = 1: промежуточный фрагмент.

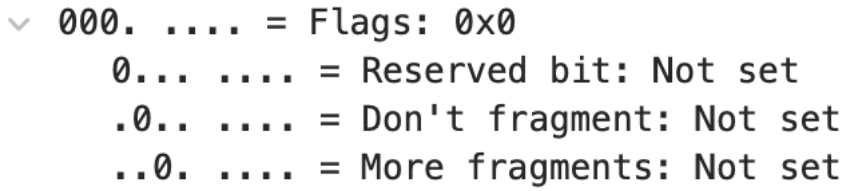
  MF = 0: последний фрагмент.

- Fragment Offset (смещение фрагмента) – на позицию фрагмента в исходном пакете (в 8-байтовых блоках).









3. Чему равно количество фрагментов при передаче ping-пакетов?

100, 200, 500, 1000, 2000, 5000, 10000

Размер пакета (байт)	Количество фрагментов
100	1
200	1
500	1
1000	1
2000	2
5000	4
10000	7

Необходимо округлить вверх до целого числа деление размера пакета на MTU = 1480

4. Построить график, в котором на оси абсцисс находится размер_пакета, а по оси ординат – количество фрагментов, на которое был разделён каждый ping-пакет.

5. Как изменить поле TTL с помощью утилиты ping?

С помощью флага –m “TTL”

Пример с командой ping -s 5000 -c 1 -m 5 rgd.ca

```
000. .... = Flags: 0x0
   0... .... = Reserved bit: Not set
   .0.. .... = Don't fragment: Not set
   ..0. .... = More fragments: Not set
...0 0010 0010 1011 = Fragment Offset: 4440
Time to Live: 5
Protocol: ICMP (1)
```

6. Что содержится в поле данных ping-пакета?

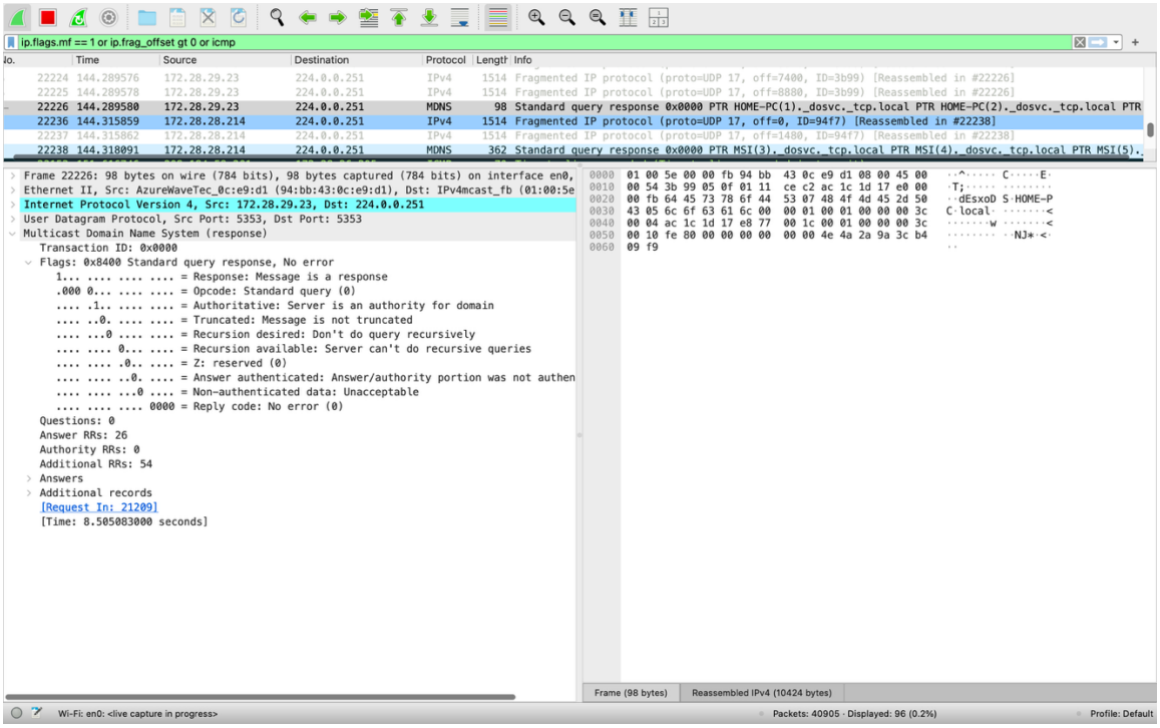
- Заголовок ICMP
- Идентификатор
- Номер последовательности
- Содержимое

4.2. АНАЛИЗ ТРАФИКА УТИЛИТЫ TRACERT (TRACEROUTE)

Запустим команду из терминала:

traceroute -n rgd.ca

Флаг -n указывает на то, что DNS пакеты отправлялись после построения маршрута, так как в них нет важного функционала



Заголовок содержит:

- - ID запроса
- - Flags – ошибки, авторитетность, тип запроса/ответа.
- Questions – кол-во запросов
- Answer– кол-во ответов
- Authority – кол-во записей авторитетных серверов
- Additional – кол-во дополнительных записей

Ответы на контрольные вопросы

1. Сколько байт содержится в заголовке IP? Сколько байт содержится в поле данных?

```
Internet Protocol Version 4, Src: 172.28.29.23, Dst: 224.0.0.251
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
```

Заголовок IPv4: 20 байт (без опций).

Поле данных:

Для ICMP-пакетов traceroute:

Заголовок ICMP — 8 байт.

Полезная нагрузка (данные) — зависит от реализации. В macOS traceroute по умолчанию отправляет UDP-пакеты с пустой нагрузкой.

Пример: UDP-пакет с заголовком 8 байт и данными 0 байт.

У ICMP заголовок равен 8 байт, а сами данные 64 байта.

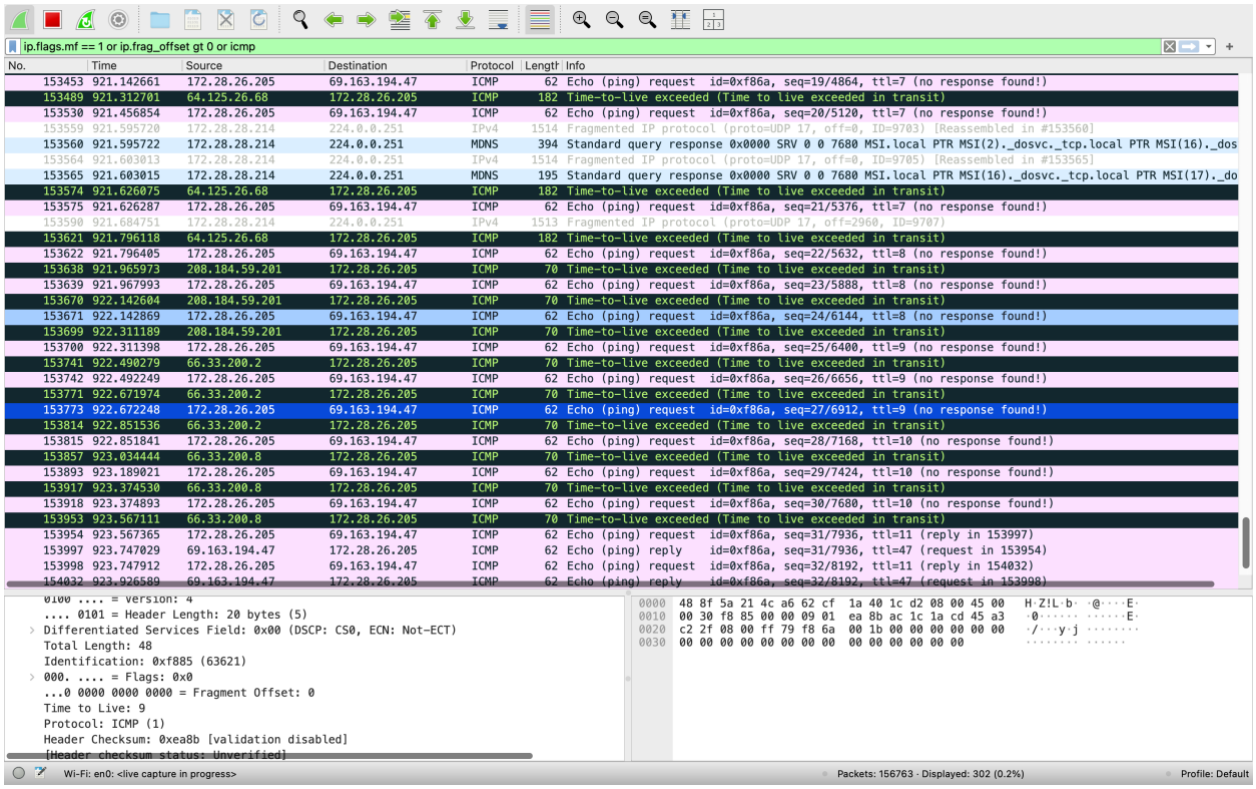
Тогда данные 72 байта. IP-заголовок 20 байт.

2. Как и почему изменяется поле TTL в следующих друг за другом ICMP- пакетах tracert? Для ответа на этот вопрос нужно проследить изменение TTL при передаче по маршруту, состоящему из более чем двух хопов.

   Принцип работы:

   traceroute отправляет пакеты с TTL до достижения цели.

   Каждый маршрутизатор уменьшает TTL на 1. При TTL=0 пакет отбрасывается, и отправитель получает ICMP-сообщение Time Exceeded.



3. Чем отличаются ICMP-пакеты, генерируемые утилитой tracert, от ICMP- пакетов, генерируемых утилитой ping (см. предыдущее задание).

   ping:

   Использует ICMP Echo Request (тип 8) и ICMP Echo Reply (тип 0).

Данные содержат временные метки и случайные байты.

traceroute (macOS):

По умолчанию отправляет UDP-пакеты на случайный порт (обычно 33434+).

Получает ICMP Time Exceeded (тип 11) от маршрутизаторов.

Для финального хоста может получить ICMP Port Unreachable (тип 3), если порт закрыт.

4. Чем отличаются полученные пакеты «ICMP reply» от «ICMP error» и зачем нужны оба этих типа ответов?

   Echo Reply подтверждает доступность узла.

   ICMP Error помогает диагностировать проблемы на пути (например, перегрузку маршрутизаторов). Приходит, когда TTL истекает

   ICMP Reply (Echo Reply):

   Ответ на успешный запрос (например, ping).

   Тип: 0.

   ICMP Error (Time Exceeded, Port Unreachable):

   Сообщает об ошибках:

   Time Exceeded (тип 11): TTL достиг нуля.

   Port Unreachable (тип 3): порт недоступен.

   Содержит заголовок исходного пакета для диагностики.

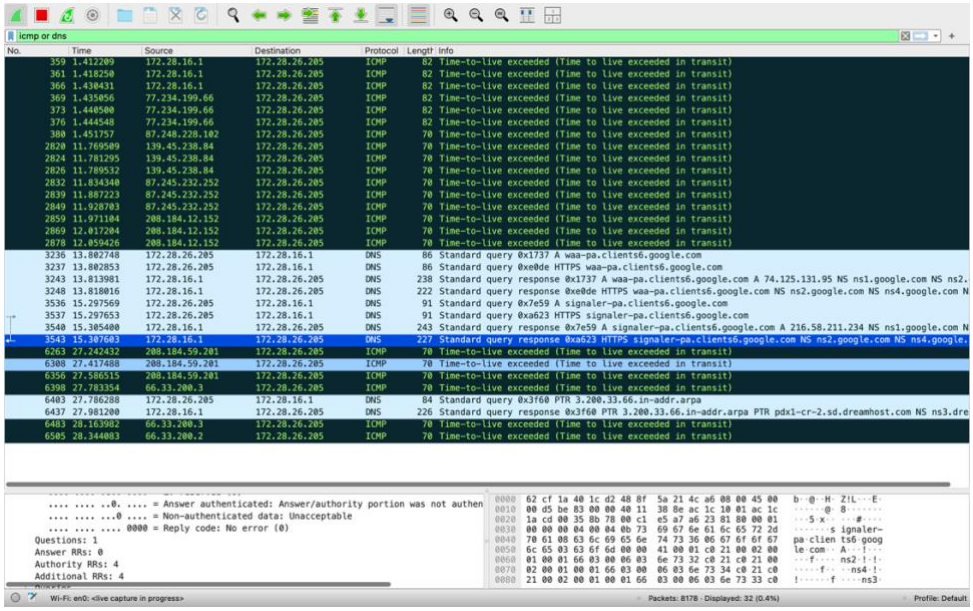
5. Что изменится в работе tracert, если убрать ключ «-d»? Какой дополнительный трафик при этом будет генерироваться?

   Без -n traceroute будет выполнять обратные DNS-запросы для IP-адресов маршрутизаторов.

   Дополнительный трафик:

   DNS-запросы (порт 53) для разрешения IP-адресов в доменные имена.

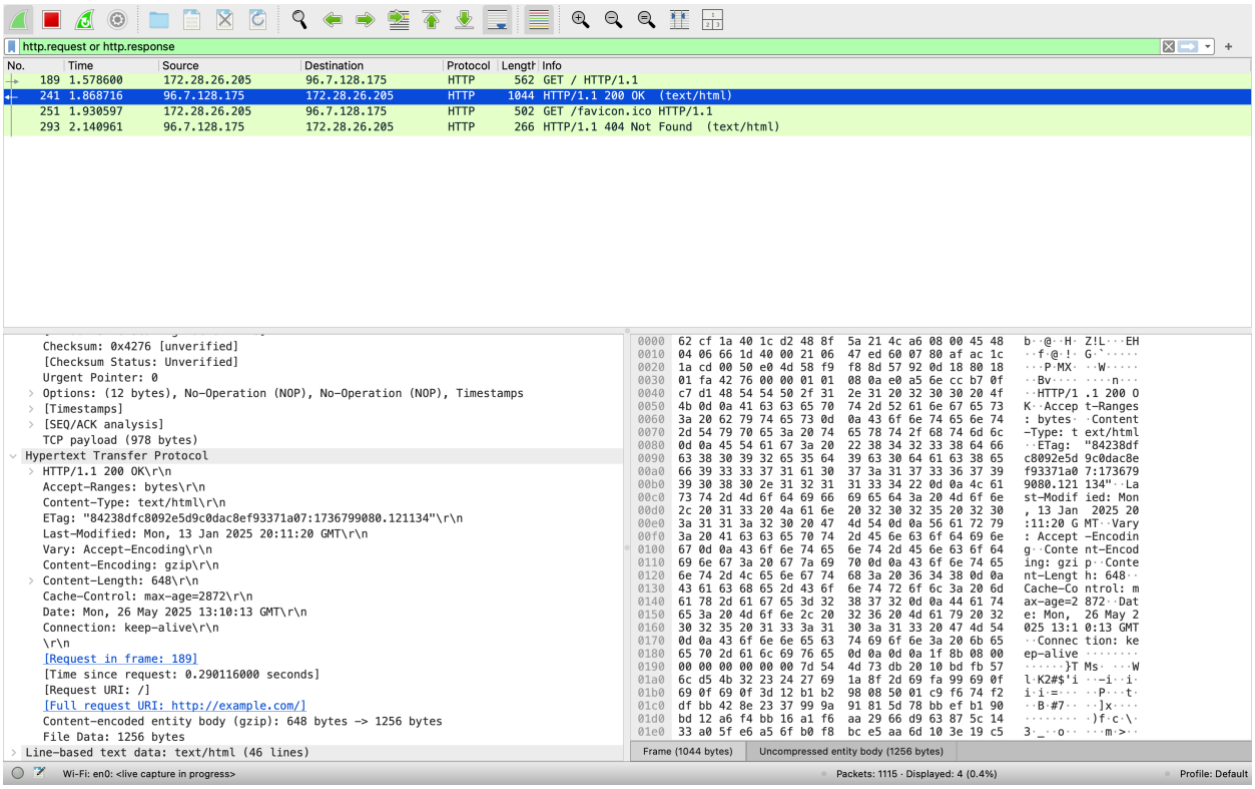
   В Wireshark будут видны пакеты с фильтром dns.



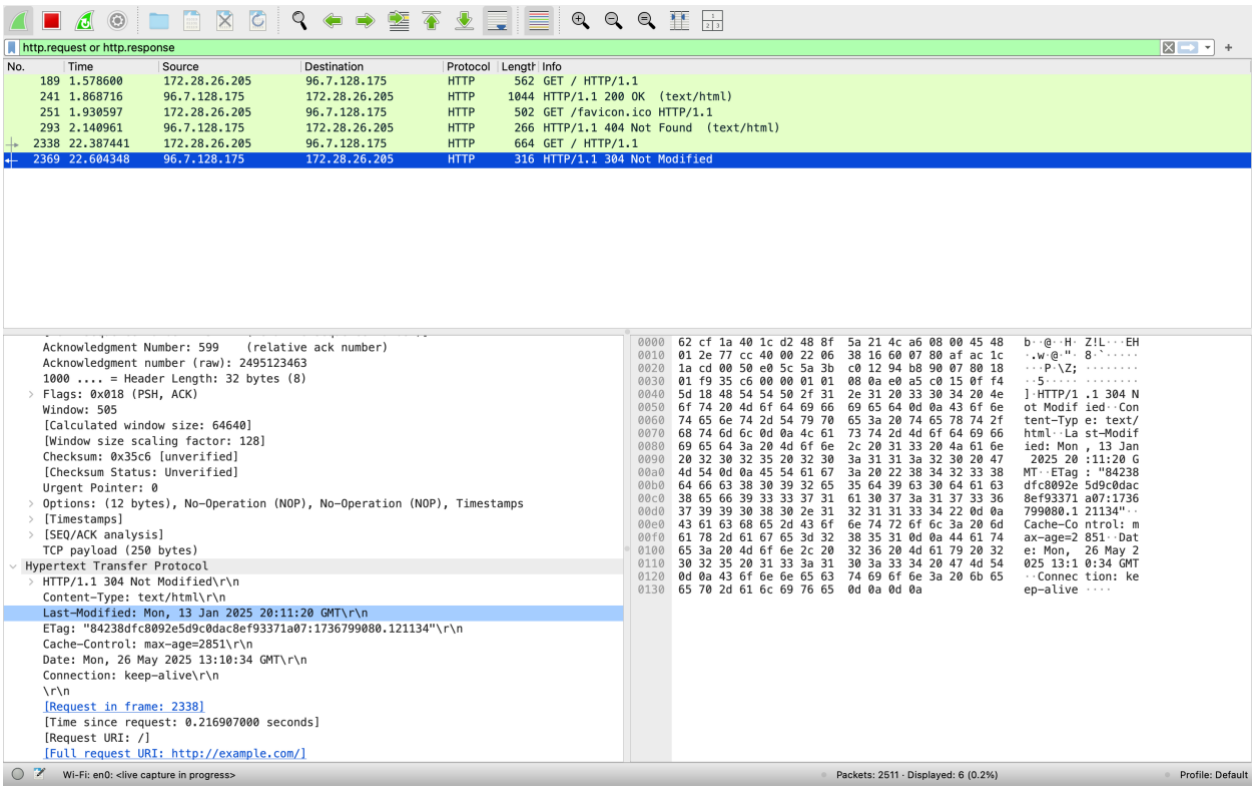
Отправка без ключа -n

4.3. АНАЛИЗ HTTP-ТРАФИКА

Установим фильтр http для захвата http пакетов. К сожалению, с нашим сайтом не получается получить ответ 304 от сервера (Not Modified), поэтому воспользуемся распространенным сайтом example.com для отправки GET-запроса



Первичный GET-запрос



Условный GET-запрос

```
Accept-Encoding: gzip, deflate\r\n
Accept-Language: ru-RU,ru;q=0.9,en-US;q=0.8,en;q=0.7\r\n
If-None-Match: "84238dfc8092e5d9c0dac8ef93371a07:1736799080.121134"\r\n
If-Modified-Since: Mon, 13 Jan 2025 20:11:20 GMT\r\n
\r\n
```

Заголовки:

If-Modified-Since — дата последнего изменения ресурса (из предыдущего ответа).

If-None-Match — уникальный идентификатор версии ресурса (ETag).

```
Hypertext Transfer Protocol
  HTTP/1.1 304 Not Modified\r\n
  Content-Type: text/html\r\n
  Last-Modified: Mon, 13 Jan 2025 20:11:20 GMT\r\n
  ETag: "84238dfc8092e5d9c0dac8ef93371a07:1736799080.121134"\r\n
```

Параметр	Первичный GET	Условный GET
Код ответа	200 OK	304 Not Modified / 200 OK
Тело ответа	Полный HTML	Отсутствует (при 304)
Заголовки запроса	Нет If-Modified-Since	Есть If-Modified-Since, ETag
Передача данных	Полная	Минимальная (кеширование)

4.8. АНАЛИЗ DHCP-ТРАФИКА

Отследим и проанализируем трафик протокола DHCP.

Проверим, что был использован DHCP. На скриншоте мы видим yiaddr и другие DHCP-поля — значит IP получен по DHCP

```
sudo ipconfig set en0 DHCP

admin@MacBook-Pro-RushB ~ % sudo ipconfig set en0 BOOTP
sleep 2
sudo ipconfig set en0 DHCP

admin@MacBook-Pro-RushB ~ % ipconfig getpacket en0

op = BOOTREPLY
htype = 1
flags = 0x0
hlen = 6
hops = 0
xid = 0x581a5b53
secs = 0
ciaddr = 0.0.0.0
yiaddr = 172.28.26.205
siaddr = 172.28.16.1
giaddr = 0.0.0.0
chaddr = 62:cf:1a:40:1c:d2
sname =
file =
options:
Options count is 7
dhcp_message_type (uint8): ACK 0x5
server_identifier (ip): 172.28.16.1
lease_time (uint32): 0x15180
subnet_mask (ip): 255.255.240.0
router (ip_mult): {172.28.16.1}
domain_name_server (ip_mult): {172.28.16.1, 77.234.194.2, 77.234.216.3}
end (none):
admin@MacBook-Pro-RushB ~ %
```

Настроим фильтр bootp - отображающий фильтр, который используется для анализа трафика протокола DHCP.

BOOTP (Bootstrap Protocol) — это более старый протокол автоматической настройки IP-адресов, появившийся в 1980-х.

DHCP (Dynamic Host Configuration Protocol) был создан как расширение BOOTP, чтобы добавить:

- аренду IP-адресов,
- повторное использование адресов,
- автоматическую настройку DNS и шлюзов,
- и другие функции.

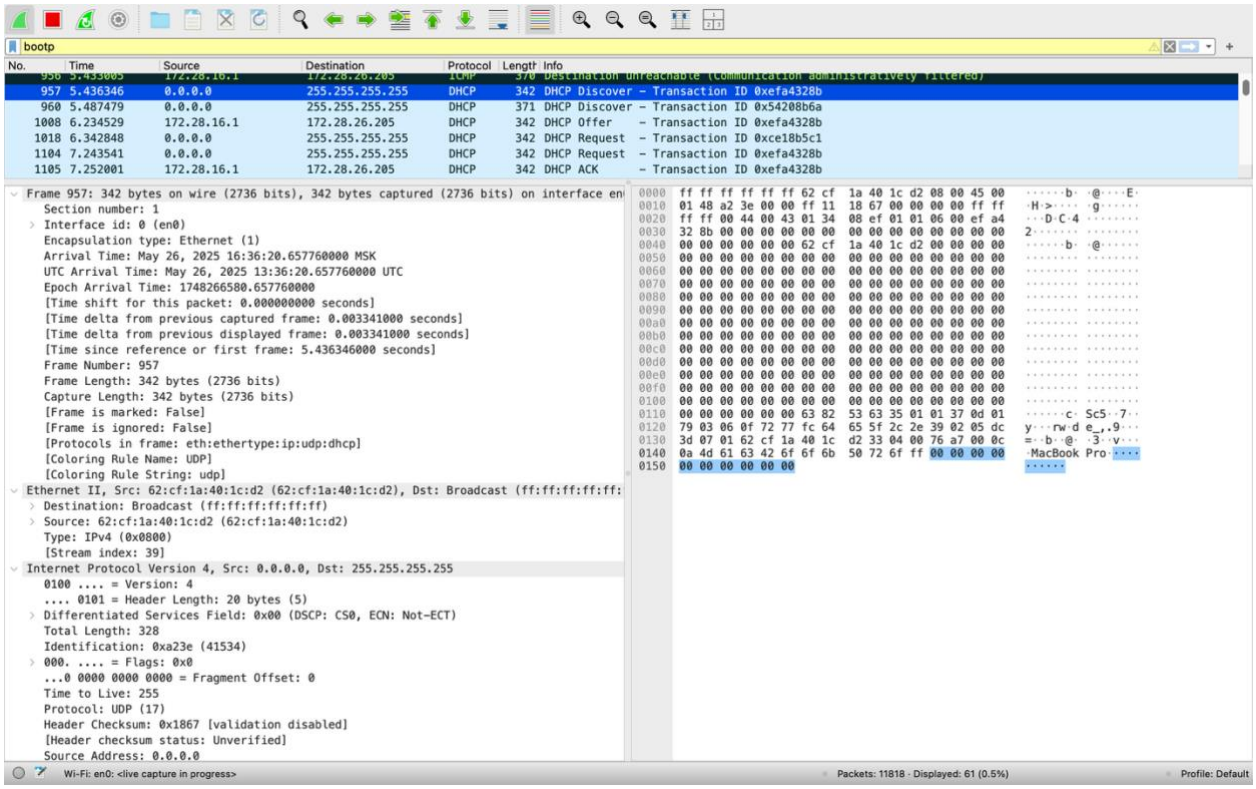
DHCP использует тот же формат сообщений, что и BOOTP, и те же порты:

- UDP порт 67 (сервер)
- UDP порт 68 (клиент)

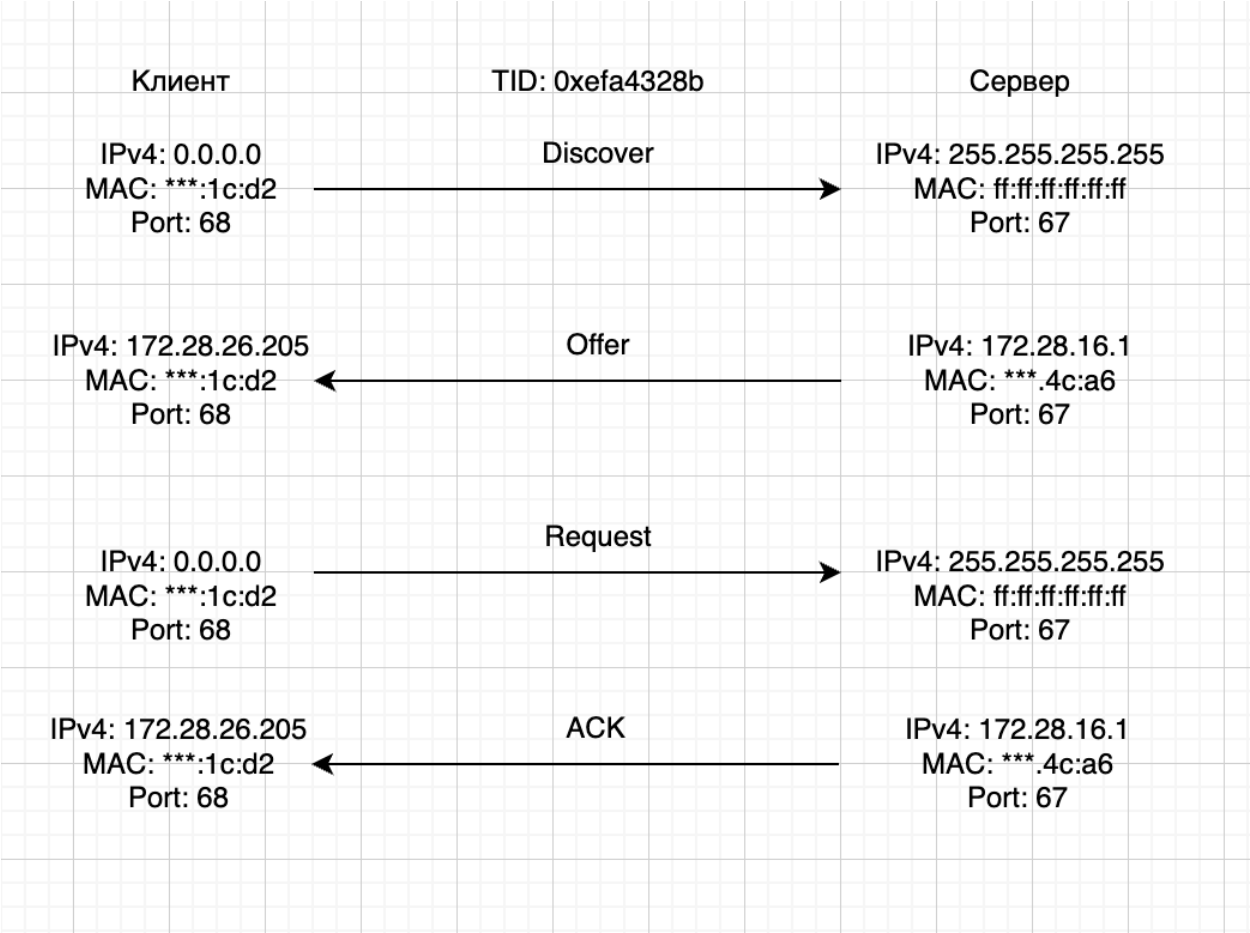
Далее несколько раз сбросим и запросим новый IP-адрес:

```
xid = 0x581a5b53
secs = 0
ciaddr = 0.0.0.0
yiaddr = 172.28.26.205
siaddr = 172.28.16.1
giaddr = 0.0.0.0
chaddr = 62:cf:1a:40:1c:d2
sname =
file =
options:
Options count is 7
dhcp_message_type (uint8): ACK 0x5
server_identifier (ip): 172.28.16.1
lease_time (uint32): 0x15180
subnet_mask (ip): 255.255.240.0
router (ip_mult): {172.28.16.1}
domain_name_server (ip_mult): {172.28.16.1, 77.234.194.2, 77.234.216.3}
end (none):
admin@MacBook-Pro-RushB ~ % sudo ipconfig set en0 BOOTP
sleep 2
sudo ipconfig set en0 DHCP

Password:
admin@MacBook-Pro-RushB ~ % sudo ipconfig set en0 BOOTP
sleep 2
sudo ipconfig set en0 DHCP

admin@MacBook-Pro-RushB ~ % sudo ipconfig set en0 BOOTP
sleep 2
sudo ipconfig set en0 DHCP

admin@MacBook-Pro-RushB ~ %
```

Получим в WireShark:



Временная диаграмма:



Ответы на контрольные вопросы

1. Чем различаются пакеты «DHCP Discover» и «DHCP Request»?

Параметр	DHCP Discover	DHCP Request
Назначение	Найти DHCP-сервер	Запросить IP-адрес от сервера
Тип	Исходная широковещательная заявка	Ответ на предложение или обновление аренды
Ответы	Получает предложения от серверов	Подтверждает выбор конкретного сервера

2. Как и почему менялись MAC- и IP-адреса источника и назначения в переданных DHCP-пакетах.

   В DHCP Discover и Request:

   MAC-адрес клиента используется, так как у клиента ещё нет IP.

   IP-адрес источника: 0.0.0.0

   IP-адрес назначения: 255.255.255.255 (широковещательный)

В ответах сервера (Offer, ACK):

MAC-адрес назначения — MAC клиента

IP-адрес источника — IP сервера

IP назначения — IP клиента

3. Каков IP-адрес DHCP-сервера?
   172.28.16.1

4. Что произойдёт, если очистить использованный фильтр «bootp»?
   Wireshark отобразит весь трафик на интерфейсе, включая ARP, DNS, HTTP и т.д.
   Это затруднит анализ DHCP-пакетов.

ВЫВОД

В ходе данной лабораторной работы было изучено строение протокольных блоков данных посредством анализа сетевого трафика с использованием утилиты Wireshark.

В частности, были проанализированы следующие протоколы:

- ICMP (ping): Определены структура и назначение заголовков Ethernet, IP и ICMP, а также изучен процесс фрагментации IP-пакетов.
- ICMP (traceroute): Установлены отличия ICMP-пакетов, генерируемых утилитами ping и traceroute, а также особенности изменения поля TTL. о
- DNS: Проанализированы DNS-запросы и ответы, а также определены основные поля DNS-заголовков.
- DHCP: Изучены этапы получения IP-адреса с использованием протокола DHCP, а также выявлены различия между пакетами DHCP Discover и DHCP Request.

Полученные результаты позволили углубить понимание принципов передачи данных в компьютерных сетях и роли каждого уровня OSI-модели в этом процессе.