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 P3331
Нодири Хисравхон

Преподаватель:
Алиев Тауфик Измайлович

г. Санкт-Петербург
2025г.

Содержание

1 Введение **3**

2 Вариант лабораторной работы **3**
2.1 Исходный URL 3

3 Анализ трафика утилиты ping **4**
3.1 Имеет ли место фрагментация исходного пакета, какое поле на это указывает? 4
3.2 Какая информация указывает, является ли фрагмент пакета последним или промежуточным? 4
3.3 Чему равно количество фрагментов при передаче ping-пакетов? 5
3.4 Построить график, в котором на оси абсцисс находится *размер_ пакета*, а по оси ординат – количество фрагментов, на которое был разделён каждый ping-пакет. 5
3.5 Как изменить поле TTL с помощью утилиты ping? 5
3.6 Что содержится в поле данных ping-пакета? 6

4 Анализ трафика утилиты tracert **6**
4.1 Сколько байт содержится в заголовке IP? Сколько байт содержится в поле данных? 6
4.2 Как и почему изменяется поле TTL в следующих друг за другом ICMP-пакетах tracert? . . 7
4.3 Чем отличаются ICMP-пакеты, генерируемые утилитой tracert, от ICMP-пакетов, генерируемых утилитой ping? 7
4.4 Чем отличаются полученные пакеты «ICMP reply» от «ICMP error» и зачем нужны оба этих типа ответов? 7
4.5 Что изменится в работе tracert, если убрать ключ «-d»? Какой дополнительный трафик при этом будет генерироваться? 8

5 Анализ HTTP-трафика **8**
5.1 Первичное посещение — обычный GET 8
5.2 Повторное посещение — условный GET 9
5.3 Вывод 10

6 Анализ ARP-трафика **10**
6.1 Какие MAC-адреса присутствуют в захваченных пакетах ARP-протокола? Что они означают? Какие устройства они идентифицируют? 11
6.2 Какие MAC-адреса присутствуют в захваченных HTTP-пакетах и что означают эти адреса? Какие устройства они идентифицируют? 11
6.3 Для чего ARP-запрос содержит IP-адрес источника? 11

7 Вывод **12**

1 Введение

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

2 Вариант лабораторной работы

2.1 Исходный URL

`https://info.kuchizu.ru`

3 Анализ трафика утилиты ping

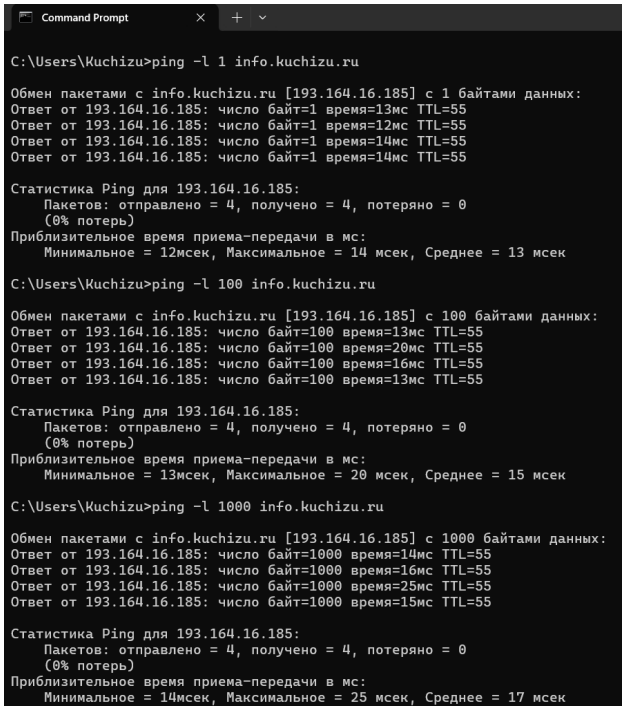

Рис. 1: 1 - 1000

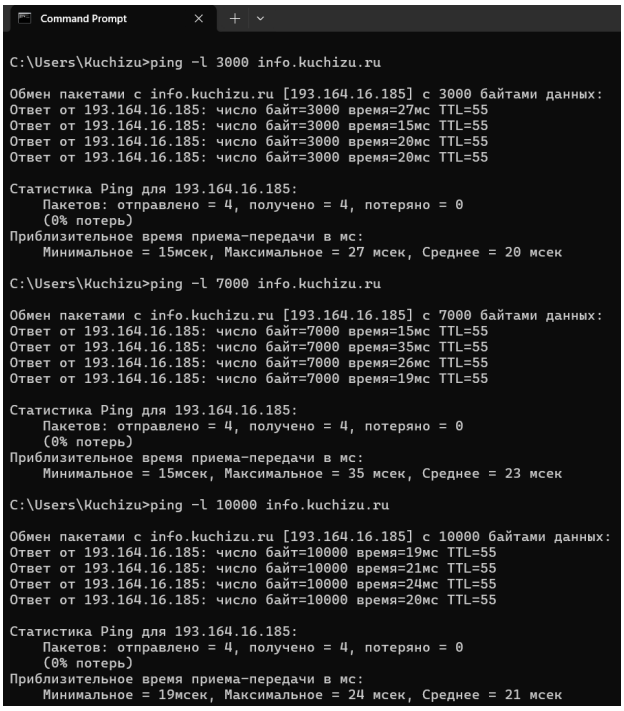

Рис. 2: 3000 - 10000

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

Да. При размерах ICMP-пакета, превышающих величину MTU (обычно 1500 байт), IP-уровень разбивает его на несколько фрагментов. Наличие фрагментации указывает поле `Flags` в заголовке IPv4, а именно бит `MF` (More Fragments) = 1.

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

Статус фрагмента определяется битом `MF`:

- `MF = 1` — это промежуточный фрагмент (дальнейшие фрагменты за ним пойдут);
- `MF = 0` — это последний фрагмент (или пакет не был фрагментирован).

Дополнительно поле `Fragment Offset` показывает смещение этого фрагмента в исходном пакете.

3.3 Чему равно количество фрагментов при передаче ping-пакетов?

По результатам серии экспериментов (Windows `ping -l <size>`) получена следующая таблица:

Размер ICMP-данных, байт	Число фрагментов
100	1
1000	1
2000	2
5000	4
10000	7
50000	34

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

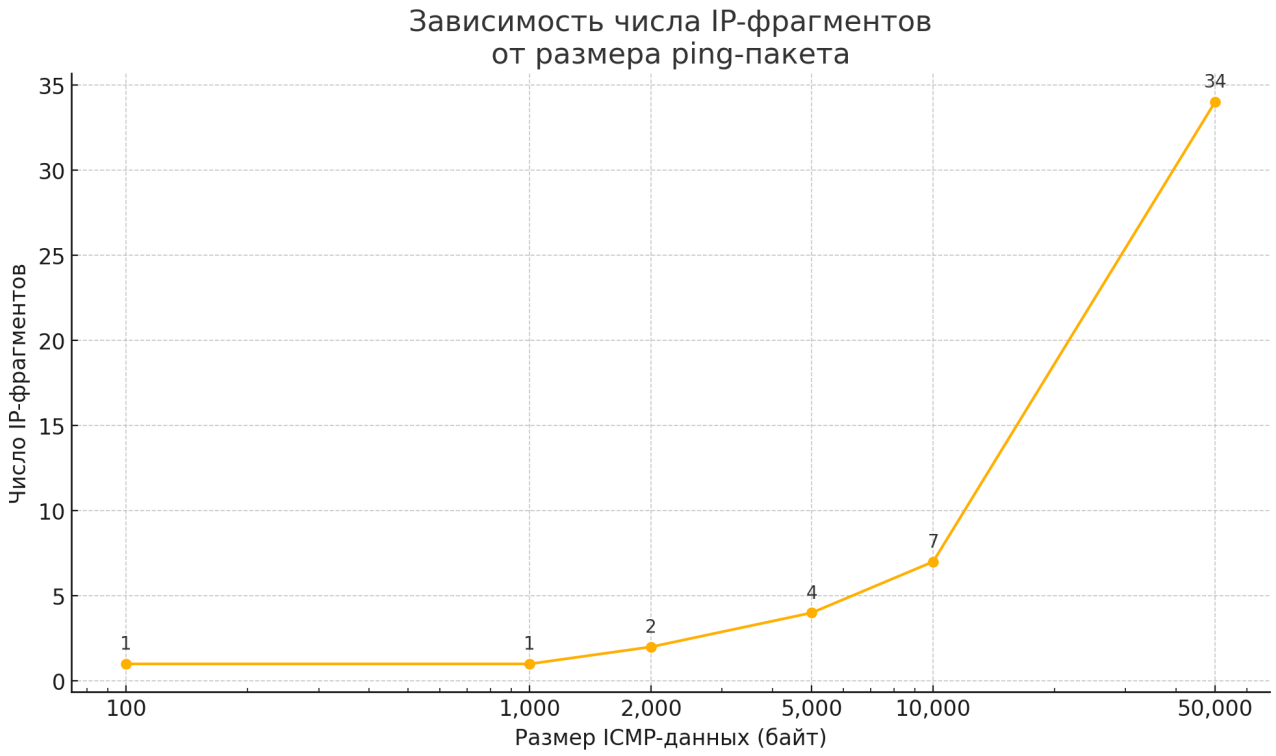

Рис. 3: Зависимость числа IP-фрагментов от размера ICMP-пакета

3.5 Как изменить поле TTL с помощью утилиты ping?

- **Windows**: `ping -i <TTL> <адрес>`
- **Linux**: `ping -t <TTL> <адрес>`

3.6 Что содержится в поле данных ping-пакета?

В разделе данных (ICMP payload) по умолчанию передаётся заполнение шаблоном ASCII-символов (обычно последовательность от `0x61` до `0x7a`, то есть «a. . . z», повторяемая до требуемого размера). Это позволяет проверить целостность и пропускную способность канала.

4 Анализ трафика утилиты tracert

```
C:\Users\Kuchizu>tracert info.kuchizu.ru

Трассировка маршрута к info.kuchizu.ru [193.164.16.185]
с максимальным числом прыжков 30:

  1     2 ms     1 ms     1 ms  XiaoQiang [192.168.31.1]
  2     *        *        *     Превышен интервал ожидания для запроса.
  3     *        *        *     Превышен интервал ожидания для запроса.
  4     *        *        *     Превышен интервал ожидания для запроса.
  5    14 ms    14 ms    14 ms  as59796.stormwall.pro [178.18.224.243]
  6    14 ms    13 ms    14 ms  10.40.64.34
  7    13 ms    13 ms    13 ms  10.40.24.1
  8    14 ms    15 ms    15 ms  10.40.99.1
  9    17 ms    16 ms    18 ms  185x108x163x142.static-business.msk.ertelecom.ru [185.108.163.142]
 10    15 ms    14 ms    14 ms  jules.dorian.example.com [193.164.16.185]

Трассировка завершена.

C:\Users\Kuchizu>tracert -d info.kuchizu.ru

Трассировка маршрута к info.kuchizu.ru [193.164.16.185]
с максимальным числом прыжков 30:

  1     2 ms     1 ms     1 ms  192.168.31.1
  2     *        *        *     Превышен интервал ожидания для запроса.
  3     *        *        *     Превышен интервал ожидания для запроса.
  4     *        *        *     Превышен интервал ожидания для запроса.
  5    14 ms    14 ms    13 ms  178.18.224.243
  6    14 ms    14 ms    21 ms  10.40.64.34
  7    17 ms    13 ms    13 ms  10.40.24.1
  8    14 ms    13 ms    13 ms  10.40.99.1
  9    16 ms    15 ms    15 ms  185.108.163.142
 10    13 ms    14 ms    13 ms  193.164.16.185

Трассировка завершена.
```

4.1 Сколько байт содержится в заголовке IP? Сколько байт содержится в поле данных?

В захваченном ICMP Echo Request, генерируемом `tracert`, IPv4-заголовок всегда составляет 20 байт (без учёта опций). Поле данных IP (IP-пейлоад) состоит из:

- 8 байт — ICMP-заголовок (Type, Code, Checksum и т.д.),
- 32 байта — пользовательские данные.

Итого IP-пейлоад занимает $8 + 32 = 40$ байт.

4.2 Как и почему изменяется поле TTL в следующих друг за другом ICMP-пакетах tracert?

`tracert` последовательно отправляет пакеты с TTL, увеличивающимся от 1 до N:

1. Первый пакет: `TTL=1` → истечение TTL на первом роутере → роутер возвращает ICMP Time Exceeded.
2. Второй пакет: `TTL=2` → проходит через первый роутер, истекает на втором → второй роутер возвращает Time Exceeded.
3. И т. д., пока TTL не станет достаточно большим, чтобы достичь конечного узла → узел отправит ICMP Echo Reply.

Поле TTL изменяется по линейному закону $(1, 2, 3, \dots)$ для идентификации каждого «прыжка» в маршруте.

4.3 Чем отличаются ICMP-пакеты, генерируемые утилитой tracert, от ICMP-пакетов, генерируемых утилитой ping?

- **ping** отправляет пакеты с фиксированным TTL (по умолчанию 128 в Windows) и ожидает только *Echo Reply* от конечного узла.
- **tracert** меняет TTL на каждом шаге, получая от промежуточных роутеров *ICMP Time Exceeded* (Type 11), а от конечного узла — *Echo Reply* (Type 0).

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

ICMP error (Time Exceeded) — тип 11, код 0. Отправляется роутером, на котором истёк TTL. Позволяет `tracert` узнать IP этого хопа и измерить время до него.

ICMP reply (Echo Reply) — тип 0. Отправляется конечным узлом в ответ на Echo Request, когда TTL достаточен, чтобы достичь цели. Завершает трассировку и даёт время до конечной точки.

Оба типа необходимы для поэтапного картографирования маршрута: первые выдают адреса промежуточных узлов, вторые — подтверждают достижение цели.

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

Без `-d` `tracert` для каждого полученного IP выполняет обратный DNS-запрос (PTR) с целью получить *имя* хоста вместо чистого IP. Это приводит к генерации одного DNS-запроса и одного DNS-ответа на каждом хопе, что:

- Увеличивает общее время трассировки (задержка на разрешение имён).
- Добавляет DNS-трафик (UDP/TCP порт 53) между вашим компьютером и DNS-сервером.

5 Анализ HTTP-трафика

При выполнении анализа HTTP-трафика мы захватили два набора пакетов при обращении к `http://info.kuchizu.ru`:

1. Первичный запрос — обычный `GET`.
2. Вторичный запрос — условный (*Conditional GET*) с заголовком `If-Modified-Since`.

5.1 Первичное посещение — обычный GET

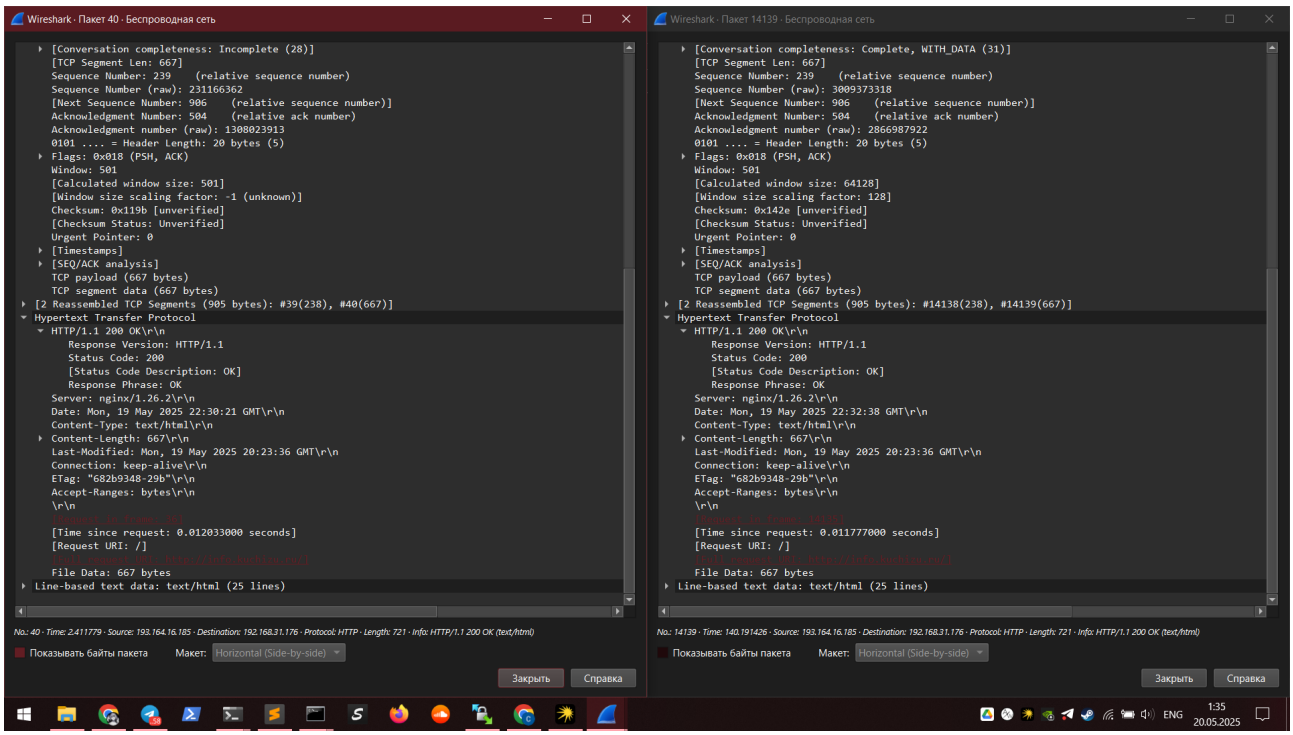

Рис. 4: Первичный GET-запрос ответ 200 OK с телом

При первом обращении браузер отправляет запрос без условий кэширования:

```
GET / HTTP/1.1
Host: info.kuchizu.ru
User-Agent: Mozilla/5.0 (...)
Accept: text/html,application/xhtml+xml,...
Connection: keep-alive
```

Сервер отвечает полным содержимым:

```
HTTP/1.1 200 OK
Content-Type: text/html
Content-Length: 667
Last-Modified: Mon, 19 May 2025 20:23:36 GMT
ETag: "682b9348-29b"
...
<html>... (полный HTML-документ) ...</html>
```

5.2 Повторное посещение — условный GET

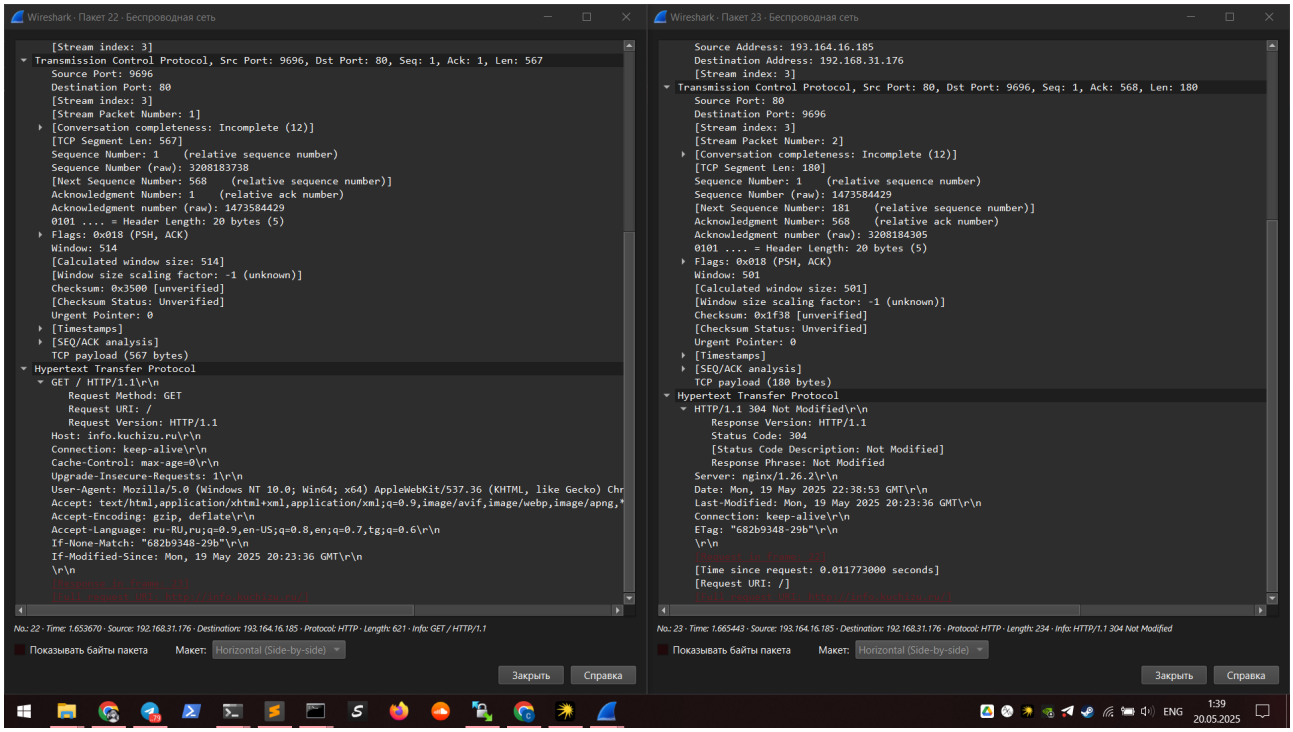

Рис. 5: Условный GET-запрос с If-Modified-Since и ответ 304 Not Modified

При обновлении страницы браузер автоматически добавляет условие по дате последнего изменения:

```
GET / HTTP/1.1
Host: info.kuchizu.ru
If-Modified-Since: Mon, 19 May 2025 20:23:36 GMT
If-None-Match: "682b9348-29b"
Connection: keep-alive
```

Поскольку ресурс не изменился, сервер возвращает код `304 Not Modified` без тела:

```
HTTP/1.1 304 Not Modified
Date: Tue, 19 May 2025 22:38:53 GMT
Server: nginx/1.26.2
ETag: "682b9348-29b"
```

5.3 Вывод

При первичном GET-запросе HTTP передаёт полный HTML-документ (ответ 200 OK). При условном GET, если ресурс не изменился, передаётся только заголовок с кодом `304 Not Modified`, что экономит трафик и ускоряет загрузку страницы.

6 Анализ ARP-трафика

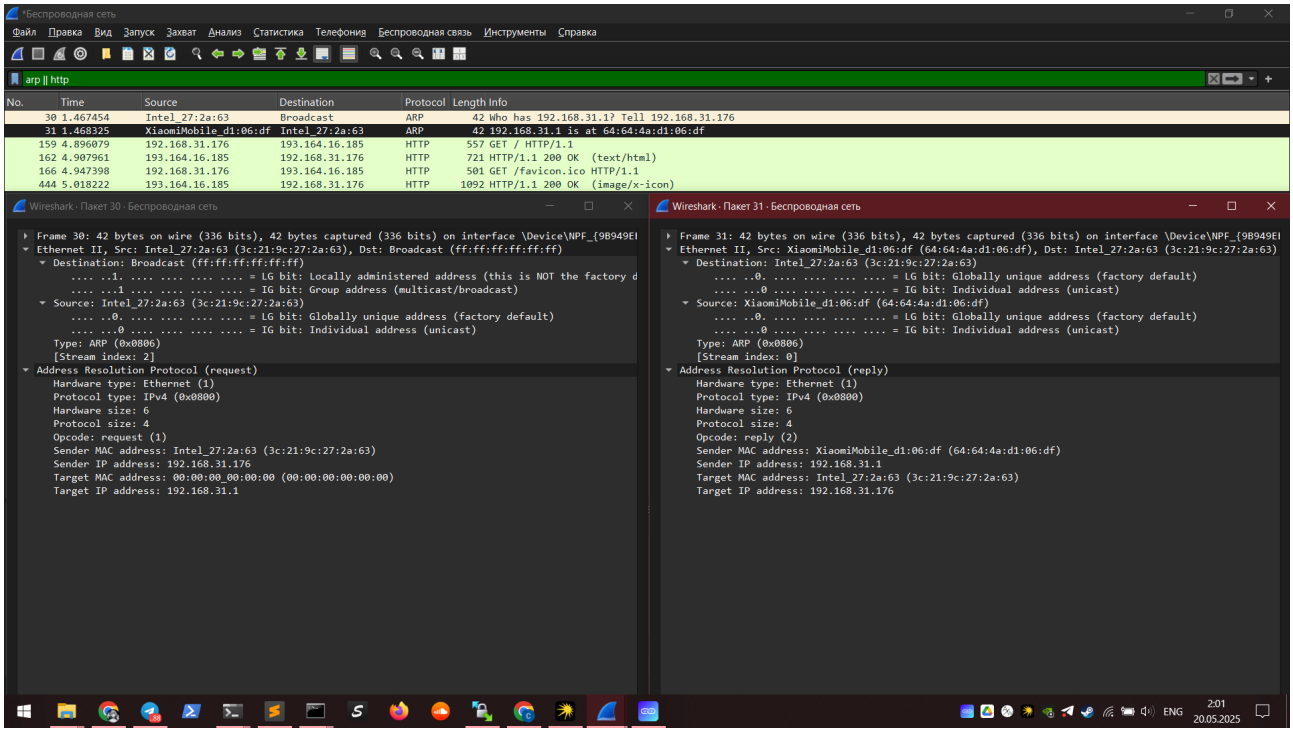

Рис. 6: Обмен ARP-запросом и ответом между клиентом и шлюзом

6.1 Какие MAC-адреса присутствуют в захваченных пакетах ARP-протокола? Что они означают? Какие устройства они идентифицируют?

В захваченном трафике присутствуют следующие ARP-пакеты:

- ARP-запрос: устройство с MAC-адресом `Intel_27:2a:63` (3c:21:9c:27:2a:63) отправило широковещательный запрос:
  `Who has 192.168.31.1? Tell 192.168.31.176`.
- ARP-ответ: устройство с MAC-адресом `XiaomiMobile_d1:06:df` (64:64:4a:d1:06:df) ответило:
  `192.168.31.1 is at 64:64:4a:d1:06:df`.

Это означает, что:

- MAC-адрес `Intel_27:2a:63` принадлежит сетевому интерфейсу компьютера пользователя.
- MAC-адрес `64:64:4a:d1:06:df` принадлежит маршрутизатору (шлюзу), который имеет IP-адрес 192.168.31.1.

6.2 Какие MAC-адреса присутствуют в захваченных HTTP-пакетах и что означают эти адреса? Какие устройства они идентифицируют?

В Ethernet-заголовках HTTP-пакетов видно:

- MAC-адрес отправителя — `Intel_27:2a:63` — это MAC-адрес компьютера пользователя.
- MAC-адрес получателя — `64:64:4a:d1:06:df` — это MAC-адрес маршрутизатора, через который идёт соединение в Интернет.

Эти адреса используются для физической доставки IP-пакетов от клиента до шлюза внутри локальной сети.

6.3 Для чего ARP-запрос содержит IP-адрес источника?

Поле с IP-адресом источника (`Sender IP address`) в ARP-запросе необходимо, чтобы получатель ARP-ответа знал:

- кому отправить ARP-ответ (на какой IP и MAC);
- и при необходимости — мог добавить MAC-адрес источника в свою собственную ARP-таблицу.

Это обеспечивает двустороннюю связь и позволяет узлам автоматически обучаться MAC-адресам своих соседей по сети.

7 Вывод

В ходе лабораторной работы были проанализированы различные типы сетевого трафика с использованием утилиты Wireshark. Получены и исследованы пакеты ARP, ICMP, HTTP и Traceroute, выявлены особенности их структуры и поведения. Рассмотрены механизмы фрагментации IP-пакетов, условных HTTP-запросов, маршрутизации и разрешения адресов. Практический анализ позволил глубже понять работу сетевых протоколов и взаимодействие между уровнями сетевой модели при обмене данными между клиентом и сервером.