

Университет ИТМО

Факультет программной инженерии и компьютерной техники

Лабораторная работа №4

**«Анализ трафика компьютерных сетей с помощью утилиты
Wireshark»**

по дисциплине “Компьютерные сети”

Выполнили:

Студенты группы Р3334

Баянов Р. Д.

Преподаватель:

Алиев Т. И.

Санкт-Петербург

2025 г.

Содержание

| | |
|---|-----------|
| Задание | 3 |
| Вариант | 4 |
| Анализ трафика утилиты ping | 5 |
| Анализа трафика утилиты tracert (traceroute) | 9 |
| Анализ HTTP-трафика | 13 |
| Анализ ARP-трафика..... | 16 |
| Вывод | 19 |

Задание

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 лабораторной работы будут представлены пункты 4.1, 4.2, 4.3, 4.5.

Сайт для анализа трафика – brd.ru

Анализ трафика утилиты ping

В командной строке поочередно с увеличением размера будем отправлять пакеты через утилиту ping на сайт *brd.ru*. Формат команды:

ping -l <размер пакета> -n <кол-во пакетов> brd.ru

Опция -n нужна для того, чтобы отправлять один пакет, так как сама по себе утилита по умолчанию отправляет 4 пакета.

Применим данную команду несколько раз для случаев, когда размер у пакета будет 100, 200, 500, 1000, 2000, 5000, 10000 байт.

Для начала опишем структуру пакета. Утилита управляет ICMP запросами и ICMP ответами. Структура:

1. Канальный уровень – Ethernet 2

Заголовок содержит:

- Destination MAC address – MAC адрес получателя.
- Source MAC address – MAC-адрес отправителя.
- Type – поле типа протокола.

2. Сетевой уровень – IP-заголовок

Заголовок содержит:

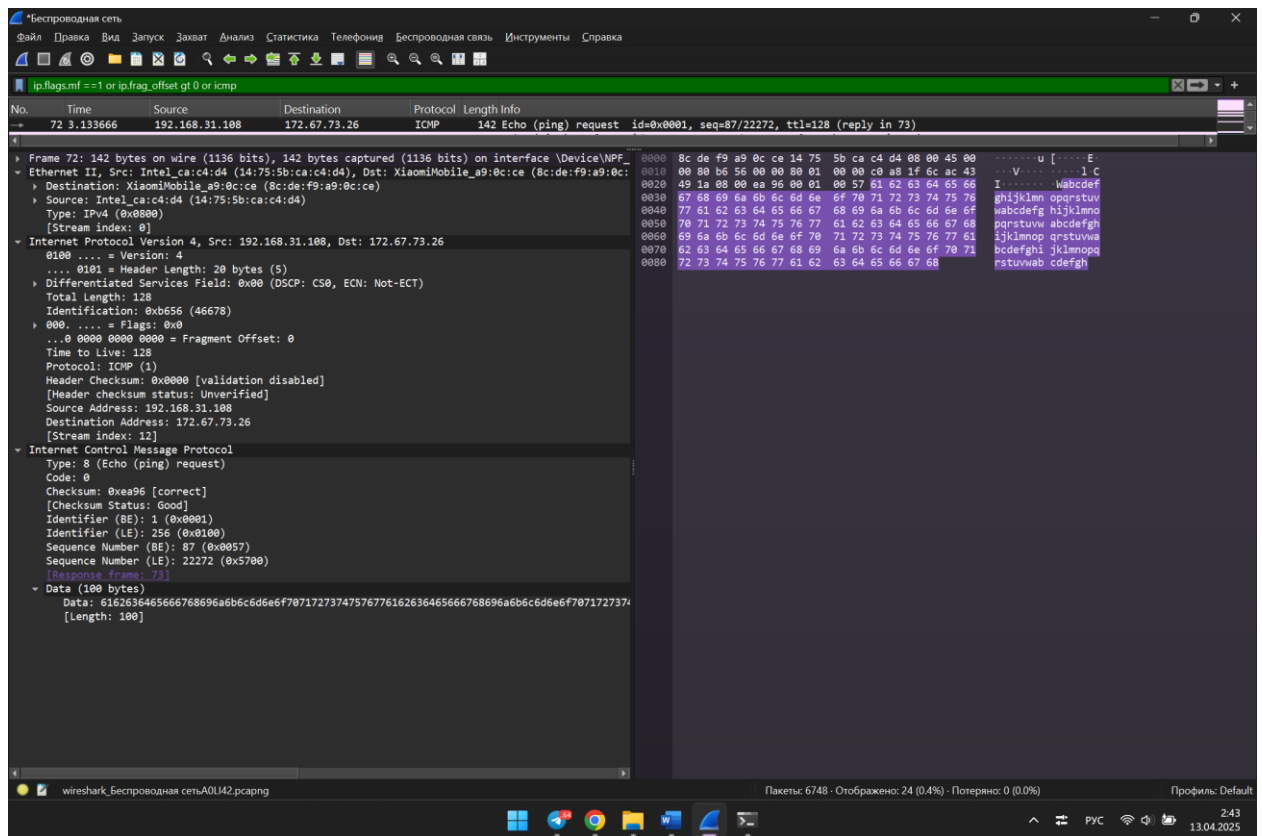
- Version
- Header Length
- Identification – идентификатор фрагмента
- Protocol – тип вложенного протокола
- Flags – указывается DF и MF
- TTL – ограничение на кол-во хопов
- Fragment offset – смещение фрагмента (если пакет был фрагментирован)
- Header Checksum – контрольная сумма заголовка
- Source IP address
- Destination IP address

3. Сетевой протокол ICMP

- Type – request или reply
- Checksum – контрольная сумма ICMP-пакета
- Identifier – уникальный ID запроса
- Seq number – номер последовательности запроса

4. Поле данных (Payload)

Примерно так выглядит структура пакета ICMP:



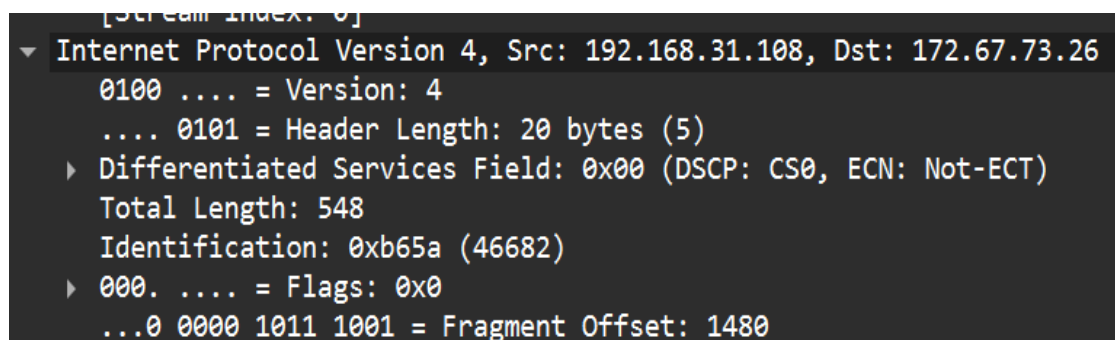
Ответы на вопросы:

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

Фрагментация происходит, какой размер IP-пакета превышает MTU (maximum transmission unit) (обычно 1480 байт для Ethernet).

Признаком фрагментации служат:

- Флаг MF (More Fragments) в IP-заголовке
- Поле Fragment Offset (смещение фрагмента)



```

▼ Internet Protocol Version 4, Src: 192.168.31.108, Dst: 172.67.73.26
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  ► Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
  Total Length: 1500
  Identification: 0xb65a (46682)
  ▼ 001. .... = Flags: 0x1, More fragments
    0... .... = Reserved bit: Not set
    .0.. .... = Don't fragment: Not set
    ..1. .... = More fragments: Set
    ...0 0000 0000 0000 = Fragment Offset: 0

```

Заметим, что когда пакет фрагментируется, то часть данных отправляется вместе с ICMP заголовком, а остальные фрагменты чисто по протоколу IP.

- Какая информация указывает, является ли фрагмент пакета последним или промежуточным?
 - MF = 1 – промежуточный фрагмент.
 - MF = 0 – последний фрагмент.

```

.0.. .... = Don't fragment: Not set
..1. .... = More fragments: Set

```

```

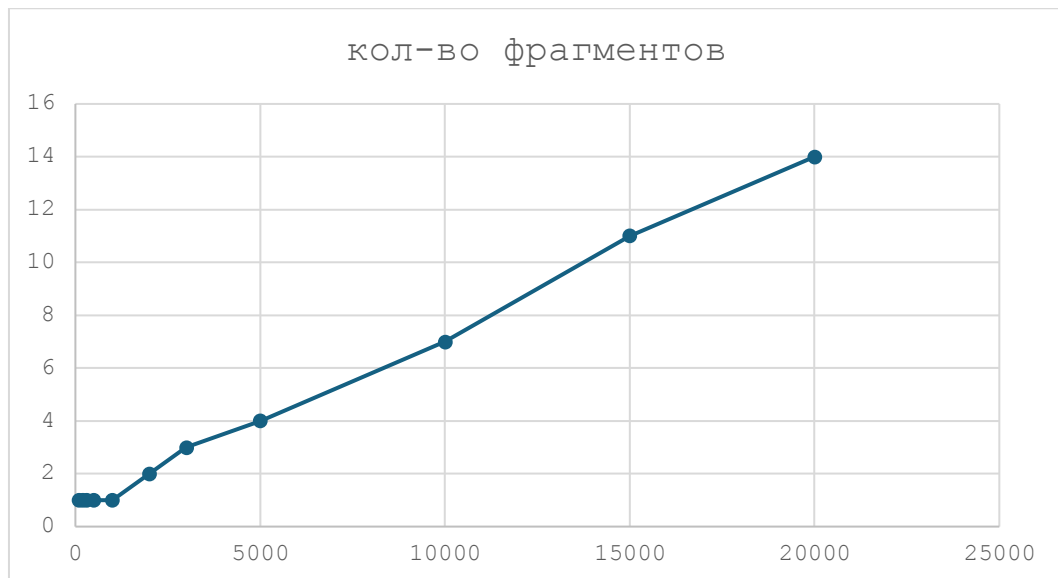
.0.. .... = Don't fragment: Not set
..0. .... = More fragments: Not set

```

- Чему равно количество фрагментов при передаче ping-пакетов? Учитывая, что один фрагмент по MTU равен примерно 1480 байт. То кол-во фрагментов будет равно *размер пакета / 1480* и округлить до верхнего целого числа.

| No. | Time | Source | Destination | Protocol | Length | Info |
|------|-----------|----------------|----------------|----------|--------|---|
| 72 | 3.133666 | 192.168.31.108 | 172.67.73.26 | ICMP | 142 | Echo (ping) request id=0x0001, seq=87/22272, ttl=128 (reply in 73) |
| 73 | 3.148458 | 172.67.73.26 | 192.168.31.108 | ICMP | 142 | Echo (ping) reply id=0x0001, seq=87/22272, ttl=55 (request in 72) |
| 137 | 12.379758 | 192.168.31.108 | 172.67.73.26 | ICMP | 242 | Echo (ping) request id=0x0001, seq=88/22528, ttl=128 (reply in 140) |
| 140 | 12.398336 | 172.67.73.26 | 192.168.31.108 | ICMP | 242 | Echo (ping) reply id=0x0001, seq=88/22528, ttl=55 (request in 137) |
| 170 | 16.805644 | 192.168.31.108 | 172.67.73.26 | ICMP | 542 | Echo (ping) request id=0x0001, seq=89/22784, ttl=128 (reply in 171) |
| 171 | 16.820949 | 172.67.73.26 | 192.168.31.108 | ICMP | 542 | Echo (ping) reply id=0x0001, seq=89/22784, ttl=55 (request in 170) |
| 206 | 21.312734 | 192.168.31.108 | 172.67.73.26 | ICMP | 1042 | Echo (ping) request id=0x0001, seq=90/23040, ttl=128 (reply in 207) |
| 207 | 21.332380 | 172.67.73.26 | 192.168.31.108 | ICMP | 1042 | Echo (ping) reply id=0x0001, seq=90/23040, ttl=55 (request in 206) |
| 274 | 24.791332 | 192.168.31.108 | 172.67.73.26 | IPv4 | 1514 | Fragmented IP protocol (proto=ICMP 1, off=0, ID=b65b) [Reassembled in #275] |
| 275 | 24.791332 | 192.168.31.108 | 172.67.73.26 | ICMP | 562 | Echo (ping) request id=0x0001, seq=91/23296, ttl=128 (reply in 276) |
| 277 | 24.818418 | 172.67.73.26 | 192.168.31.108 | IPv4 | 1514 | Fragmented IP protocol (proto=ICMP 1, off=0, ID=2cad) [Reassembled in #278] |
| 278 | 24.818418 | 172.67.73.26 | 192.168.31.108 | ICMP | 562 | Echo (ping) reply id=0x0001, seq=91/23296, ttl=55 (request in 275) |
| 396 | 27.572940 | 192.168.31.108 | 172.67.73.26 | IPv4 | 1514 | Fragmented IP protocol (proto=ICMP 1, off=0, ID=b65b) [Reassembled in #399] |
| 397 | 27.572940 | 192.168.31.108 | 172.67.73.26 | IPv4 | 1514 | Fragmented IP protocol (proto=ICMP 1, off=1480, ID=b65b) [Reassembled in #399] |
| 398 | 27.572940 | 192.168.31.108 | 172.67.73.26 | IPv4 | 1514 | Fragmented IP protocol (proto=ICMP 1, off=2960, ID=b65b) [Reassembled in #399] |
| 399 | 27.572940 | 192.168.31.108 | 172.67.73.26 | ICMP | 602 | Echo (ping) request id=0x0001, seq=92/23552, ttl=128 (no response found) |
| 1234 | 37.321956 | 192.168.31.108 | 172.67.73.26 | IPv4 | 1514 | Fragmented IP protocol (proto=ICMP 1, off=0, ID=b65b) [Reassembled in #1240] |
| 1235 | 37.321956 | 192.168.31.108 | 172.67.73.26 | IPv4 | 1514 | Fragmented IP protocol (proto=ICMP 1, off=1480, ID=b65c) [Reassembled in #1240] |
| 1236 | 37.321956 | 192.168.31.108 | 172.67.73.26 | IPv4 | 1514 | Fragmented IP protocol (proto=ICMP 1, off=2960, ID=b65c) [Reassembled in #1240] |
| 1237 | 37.321956 | 192.168.31.108 | 172.67.73.26 | IPv4 | 1514 | Fragmented IP protocol (proto=ICMP 1, off=4440, ID=b65c) [Reassembled in #1240] |
| 1238 | 37.321956 | 192.168.31.108 | 172.67.73.26 | IPv4 | 1514 | Fragmented IP protocol (proto=ICMP 1, off=5920, ID=b65c) [Reassembled in #1240] |
| 1239 | 37.321956 | 192.168.31.108 | 172.67.73.26 | IPv4 | 1514 | Fragmented IP protocol (proto=ICMP 1, off=7400, ID=b65c) [Reassembled in #1240] |
| 1240 | 37.321956 | 192.168.31.108 | 172.67.73.26 | ICMP | 1162 | Echo (ping) request id=0x0001, seq=93/23808, ttl=128 (no response found) |

- График: размер пакета – кол-во фрагментов.



5. Как изменить поле TTL с помощью утилиты ping?

Изменить это поле можно командой: `ping -l 3000 -n 1 -i 5 brd.ru`

```
Time to Live: 5
```

```
Protocol: ICMP (1)
```

6. Что содержится в поле данных ping-пакета?

- Заголовок ICMP
- Идентификатор
- Номер последовательности
- Содержимое

Анализа трафика утилиты `tracert` (`traceroute`)

Введём в командную строку команду `tracert brd.ru`

Данная утилита также пользуется протоколом ICMP, поэтому разбирать его структуру мы не будем. Но, помимо этого, утилита `tracert` отправляет DNS-пакеты. DNS – это протокол, который переводит доменные имена в IP-адреса, которые понятны компьютерам. С помощью ключа `-d` можно сделать так, чтобы DNS пакеты отправлялись уже после построения маршрута, так как они не несут в себе важный функционал.

Вот структура DNS пакета:

```
Domain Name System (query)
  Transaction ID: 0xef58
  Flags: 0x0100 Standard query
    0... .. = Response: Message is a query
    .000 0... .. = Opcode: Standard query (0)
    .... ..0. .... = Truncated: Message is not truncated
    .... ..1 .... = Recursion desired: Do query recursively
    .... ..0.. .... = Z: reserved (0)
    .... ..0 .... = Non-authenticated data: Unacceptable
  Questions: 1
  Answer RRs: 0
  Authority RRs: 0
  Additional RRs: 0
  Queries
    ledger.bt.co: type A, class IN
      Name: ledger.bt.co
      [Name Length: 12]
      [Label Count: 3]
      Type: A (1) (Host Address)
      Class: IN (0x0001)
[Response In: 969]
```

Заголовок размером 12 байт содержит:

- ID – уникальный ID-запроса
- Flags – ошибки, авторитетность, тип запроса/ответа.
- QDCOUNT – кол-во запросов
- ANCOUNT – кол-во ответов
- INSCOUNT – кол-во записей авторитетных серверов
- ARCOUNT – кол-во дополнительных записей

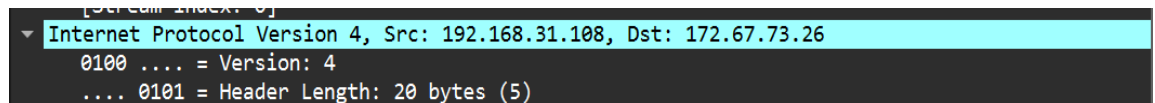
Раздел вопросов, который содержит доменное имя, которое мы запрашиваем.

Раздел ответов, который содержит IP-адрес в ответ на запрос.

Ответы на вопросы:

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

Заголовок IP обычно составляет 20 байт для IPv4.



Поле данных – это содержимое, инкапсулированное в IP-пакете, ICMP-пакета.

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

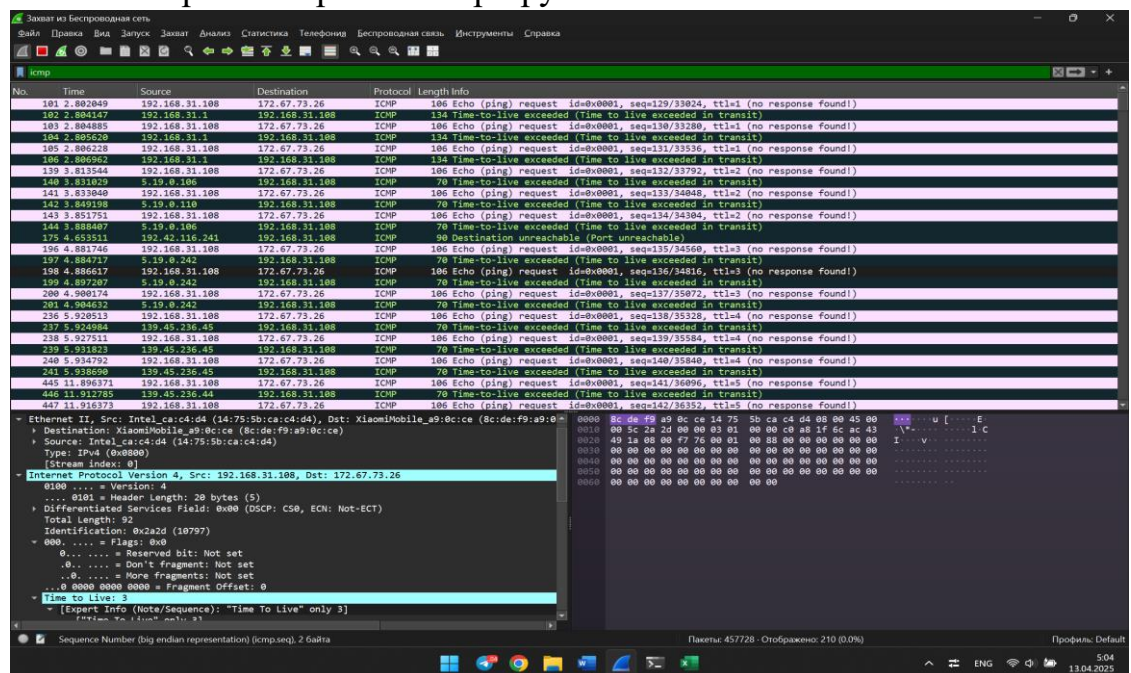
Следовательно, данные 72 байта. IP-заголовок 20 байт.

2. Как и почему изменяется поле TTL в следующих ICMP-пакетах tracer? Утилита tracer отправляет ICMP-пакеты с увеличивающимся TTL, начиная с 1.

Каждый маршрутизатор уменьшает TTL на 1. Когда TTL становится 0 – маршрутизатор отбрасывает пакет и отправляет обратно ICMP Time Exceeded.

Это позволяет tracer определить каждый узел на пути.

TTL изменяется поэтапно, чтобы каждый узел по очереди откликнулся, и таким образом строится маршрут.



3. Чем отличаются ICMP-пакеты, генерируемые tracer, от ICMP-пакетов ping?

ping всегда шлёт ICMP Echo Request и ждёт Echo Reply.

Tracer использует ICMP Echo Request с разным TTL и анализирует:

- ICMP Time Exceeded от промежуточных маршрутизаторов.
- ICMP Echo Reply от конечного узла.

То есть ping проверяет доступность узла, а traceroute строит маршрут до него.

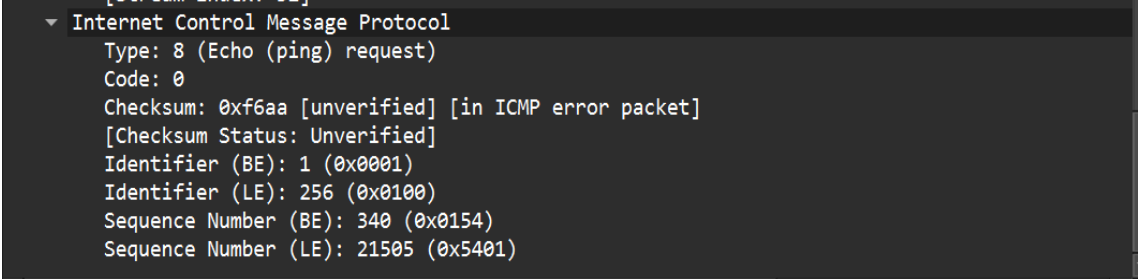
4. Чем отличаются ICMP reply от ICMP error и зачем нужны оба?

ICMP reply – отклик от целевого хоста, подтверждающий, что он доступен.

ICMP error – приходит от маршрутизаторов, когда TTL истекает. Эти пакеты нужны для определения маршрута.

Оба типа позволяют traceroute:

- Узнать IP каждого промежуточного маршрутизатора (через error).
- Подтвердить достижение конечного узла (через reply).



```

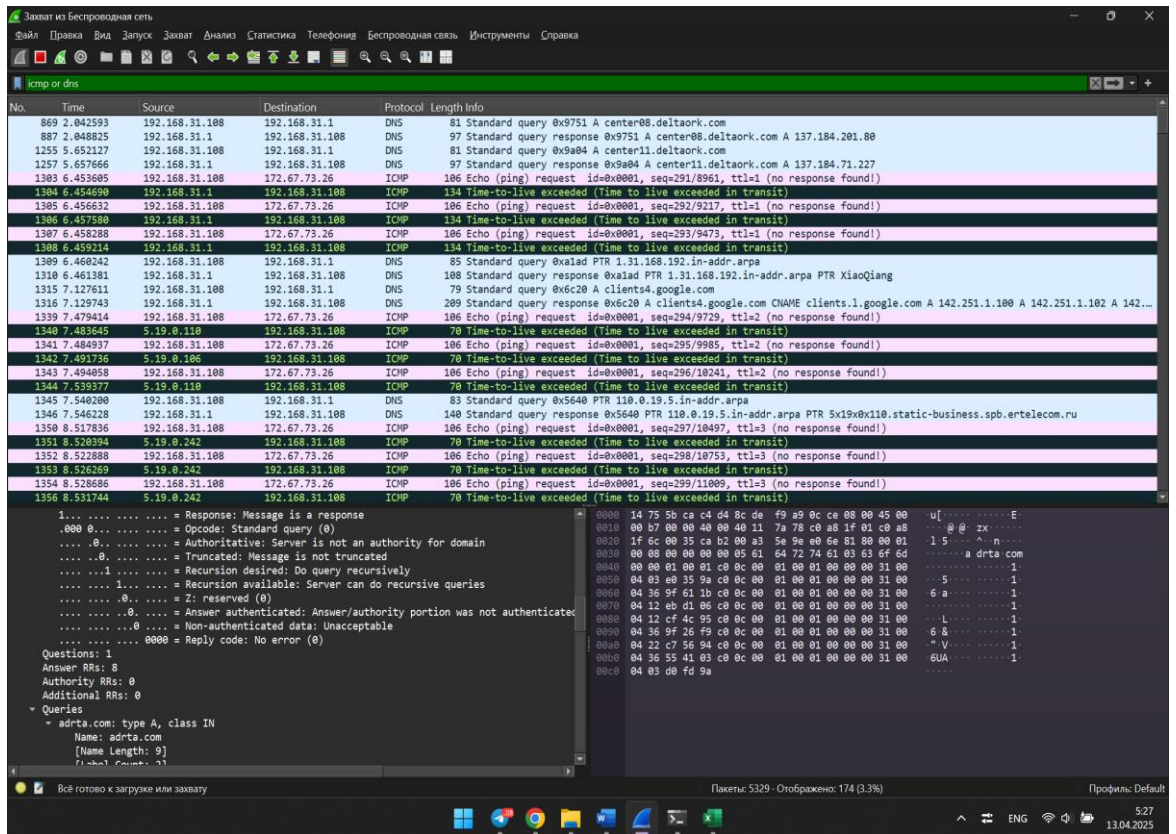
▼ Internet Control Message Protocol
  Type: 8 (Echo (ping) request)
  Code: 0
  Checksum: 0xf6aa [unverified] [in ICMP error packet]
  [Checksum Status: Unverified]
  Identifier (BE): 1 (0x0001)
  Identifier (LE): 256 (0x0100)
  Sequence Number (BE): 340 (0x0154)
  Sequence Number (LE): 21505 (0x5401)
  
```

5. Что изменится в работе traceroute, если убрать ключ -d? Какой трафик будет генерироваться дополнительно?

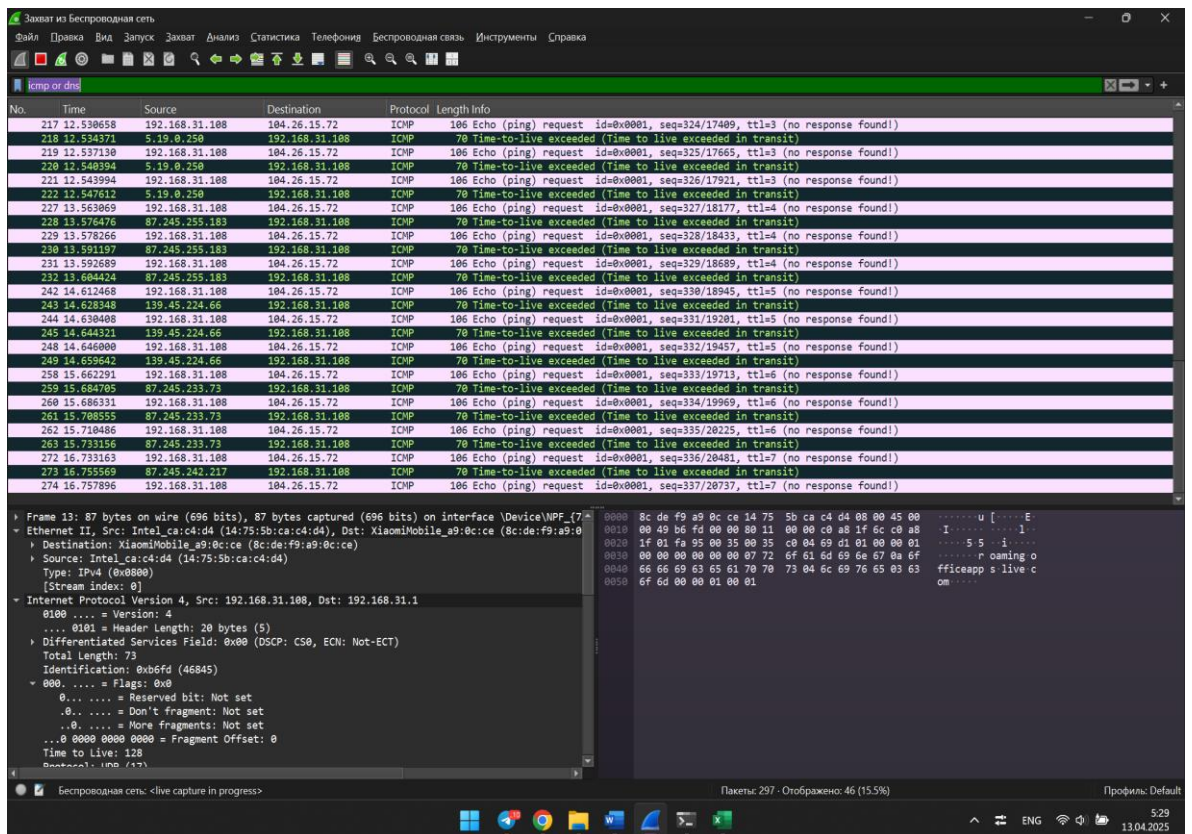
Ключ -d отключает обратное разрешение IP-адресов в доменные имена.

Без -d traceroute будет пытаться разрешить IP-адреса в имена хостов (через DNS). Это приведёт к дополнительному DNS-трафику, так как каждый IP будет запрашиваться у DNS-сервера.

Вот окно Wireshark без ключа -d:

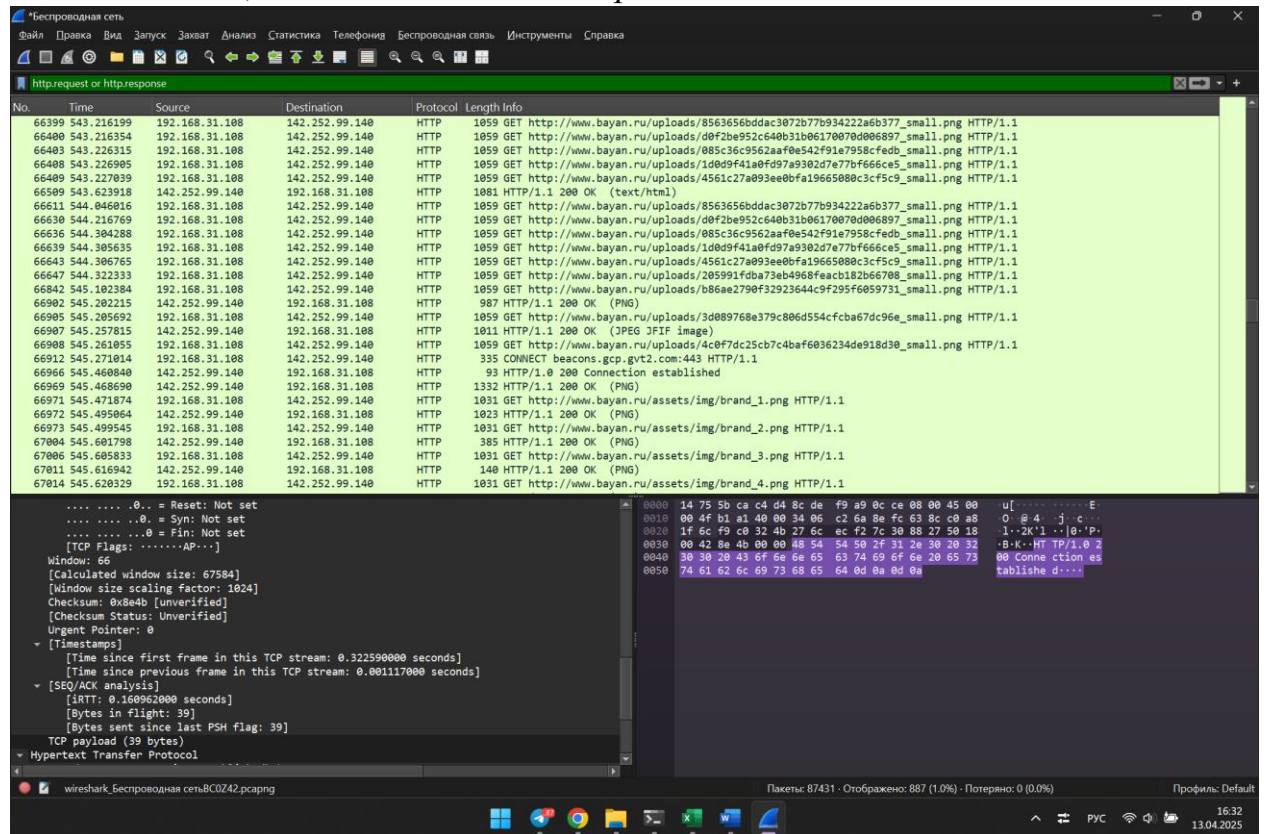


Вот окно wireshark с ключом -d:

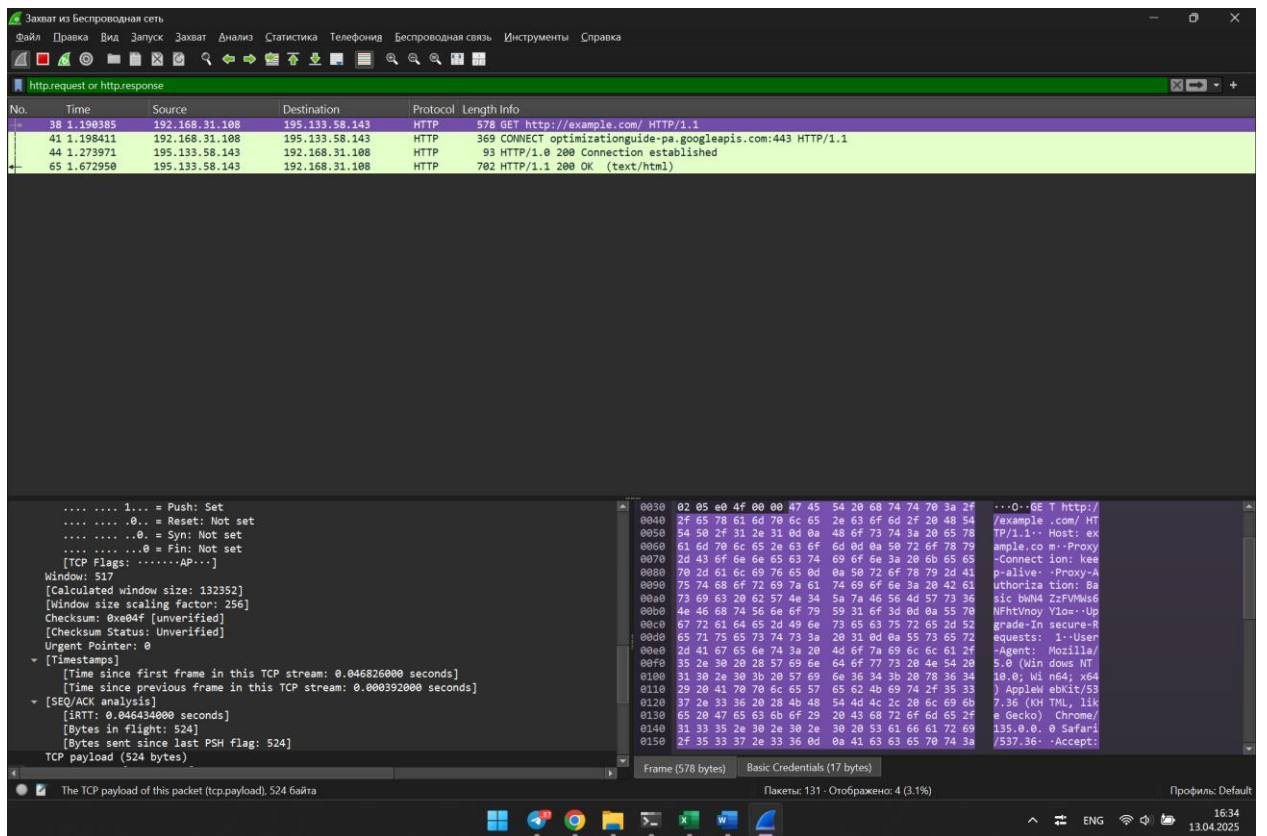


Анализ HTTP-трафика

Запустим анализ в Wireshark и перейдём на сайт brd.ru. К сожалению, сайт, который подходит нам по варианту, не обладает возможностью принимать условные GET-запросы. Сколько раз не обновляй мы не можем получить ответ 304. Поэтому воспользуемся сайтом, который точно обладает такой возможностью, а именно сайтом *example.com*.

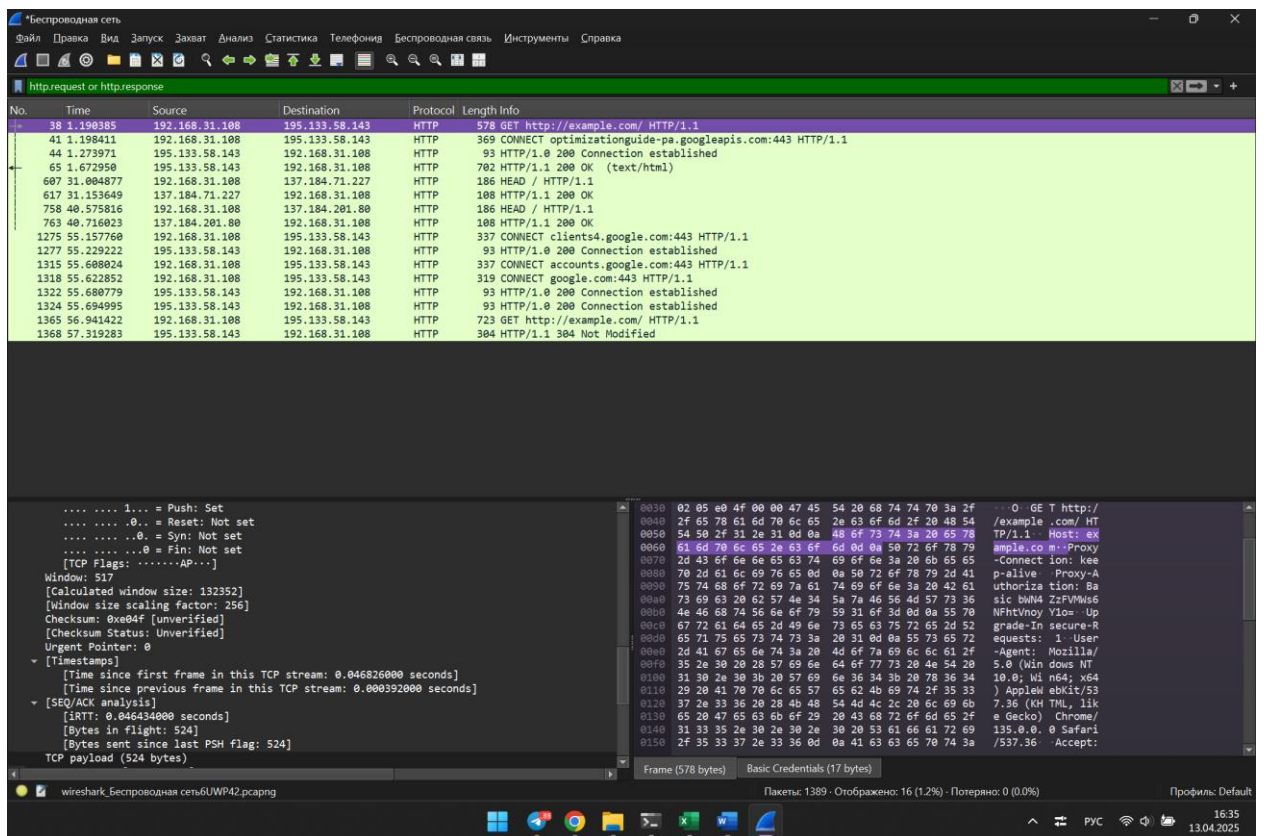


Сначала просто зайдём на сайт *example.com*.



Заметим, что всё отработало как надо, мы получаем ответ 200.

Теперь попробуем обновить страницу и посмотрим, что будет.



```
Content-Type: text/html\r\n
```

```
Last-Modified: Mon, 13 Jan 2025 20:11:20 GMT\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
```

Заметим, что мы получаем совсем другую ситуацию. Здесь у нас получилось отправить условный GET-запрос. И мы получаем ответ 304 от сервера. Это можно понять по появившимся полям Last-Modified и If-Modified-Since.

Анализ ARP-трафика

Для начала очистим ARP-таблицу с помощью команды:

netsh interface ip delete arpcache

```
C:\Users\RavvCheck1>netsh interface ip delete arpcache
OK.
```

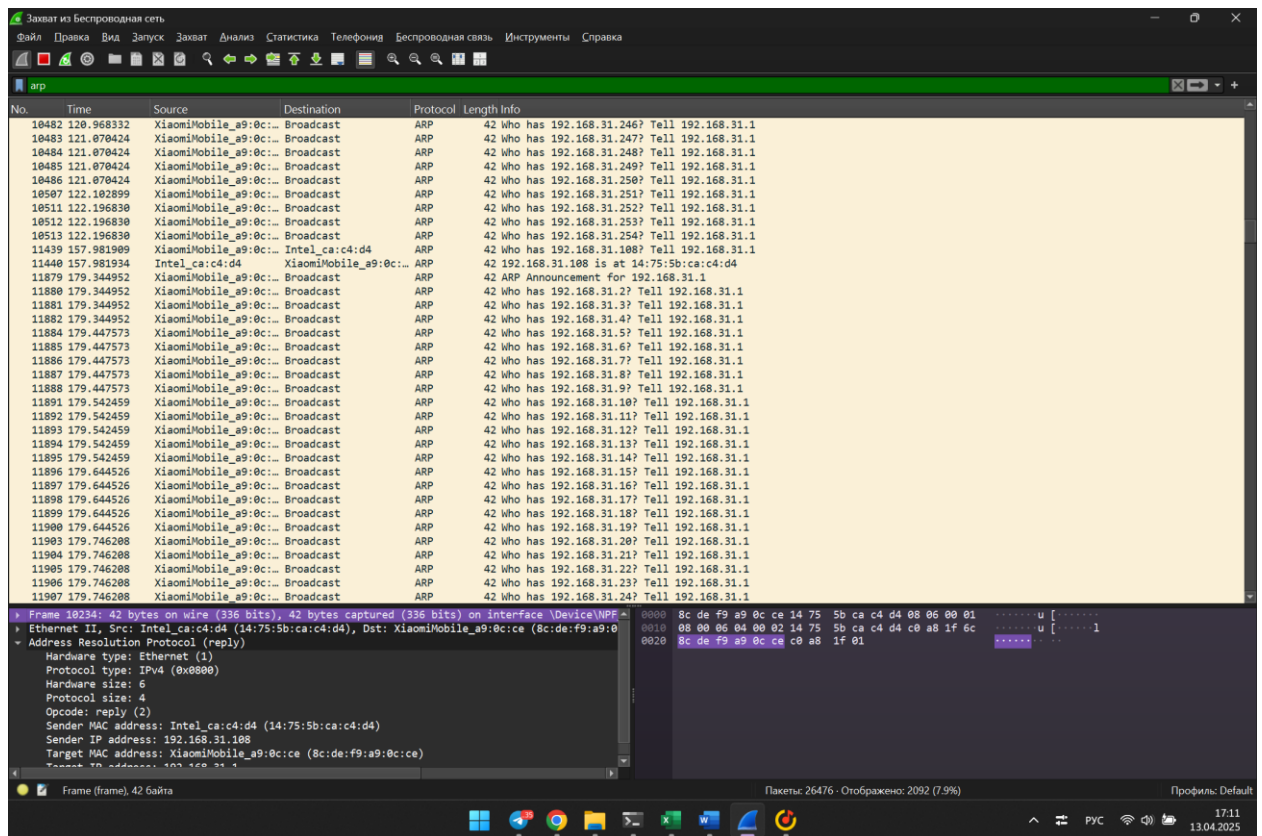
Получим вот такую arp-таблицу:

```
Интерфейс: 192.168.112.1 --- 0x30
  адрес в Интернете      Физический адрес      Тип
224.0.0.22              01-00-5e-00-00-16      статический
224.0.0.251             01-00-5e-00-00-fb      статический
```

После удаления кэша браузера отправимся на сайт brd.ru и увидим новую запись, в arp-таблице.

```
Интерфейс: 192.168.31.108 --- 0ха
  адрес в Интернете      Физический адрес      Тип
192.168.31.1            8c-de-f9-a9-0c-ce      динамический
```

Заметим, что это вообще не похоже на IP адрес сайта, на который мы перешли. А всё, потому что MAC-адреса используются только в локальной сети. Мы не сможем увидеть ARP-запрос, который узнаёт MAC-адрес нашего сайта, так как его и вовсе нет. Но мы видим IP-адрес 192.168.31.1. Вероятнее всего это IP нашего маршрутизатора, который как раз таки и взялся в дальнейшем уже за поиск того сайта, на который мы перешли.



Ответы на вопросы:

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

В ARP-пакетах мы увидим два типа MAC-адресов:

- MAC-адрес отправителя запроса – адрес нашего компьютера. Он используется в поле Sender MAC-address
- MAC-адрес искомого устройства:

В ARP-запросе (who-has) поле Target MAC Address будет заполнено нулями, потому что он ещё известен.

В ARP-ответе (is-at) это будет MAC-адрес шлюза/маршрутизатора, провайдера или другого узла локальной сети, связанного с IP, на который отправляется запрос.

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

```
▼ Ethernet II, Src: XiaomiMobile_a9:0c:ce (8c:de:f9:a9:0c:ce), Dst: Intel_ca:c4:d4 (14:75:5b:ca:c4:d4)
  ▸ Destination: Intel_ca:c4:d4 (14:75:5b:ca:c4:d4)
  ▸ Source: XiaomiMobile_a9:0c:ce (8c:de:f9:a9:0c:ce)
    Type: IPv4 (0x0800)
    [Stream index: 1]
```

HTTP работает поверх TCP/IP и Ethernet. В Ethernet-заголовке каждого HTTP-пакета указывается:

- MAC-адрес источника – это MAC-адрес твоего компьютера
- MAC-адрес назначения – это обычно MAC-адрес ближайшего маршрутизатора/шлюза, через который трафик пойдёт в Интернет.

MAC-адреса веб-сайта, на который мы заходим, мы не увидим, потому что MAC-адреса используются только внутри локальной сети.

3. Для чего ARP-запроса содержит IP-адрес источника?

ARP-запроса содержит IP-адрес источника, чтобы:

- Получатель запроса (тот, чей IP адрес запрашивается) мог записать в свою ARP-таблицу соответствие, и тем самым сократить количество ARP-запросов в будущем.
- Получатель понимал, кто запрашивает – это нужно для формирования ARP-запроса-ответа.

IP-адрес источника нужен для обратной связи и корректного построения локальной маршрутизации.

Вывод

Выполнив данную лабораторную работу, я с помощью программы wireshark проанализировал передачу пакетов по сети. Мне удалось описать структуры DNS, ICMP, IP, ARP и HTTP протоколов. Выяснил, что передача по сети на самом деле очень сложный механизм, который включает в себя взаимодействие огромного количества протоколов и интерфейсов.