

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«Санкт-Петербургский национальный исследовательский университет
информационных технологий, механики и оптики»

ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И КОМПЬЮТЕРНЫХ
ТЕХНОЛОГИЙ

Лабораторная работа №4
«Локальные сети»
по дисциплине
«Компьютерные сети»

Выполнил:
Векшин А. И. Р3316
Преподаватель:
Тропченко А.А.

| | |
|------------------------------------|-----------|
| Анализ трафика утилиты ping | 3 |
| Запрос ICMP | 4 |
| Ответ ICMP | 5 |
| Анализ утилиты tracert | 8 |
| Анализ HTTP-трафика | 11 |
| Вывод | 14 |

Анализ трафика утилиты ping

Команда: ping vekshin.ru -t

Команда-фильтр: ip.src == 5.101.152.33 or ip.dst == 5.101.152.33

Структура ICMP-запроса

Канальный уровень – Ethernet 2

Заголовок содержит:

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

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

Сетевой протокол ICMP

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

Поле данных (Payload)

Запрос ICMP

▼ Frame 91703: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface \Device\NPF_{E02C3A3B-DFDF-4C6D-B6BE-BABD8FC0981A}

Section number: 1

> Interface id: 0 (\Device\NPF_{E02C3A3B-DFDF-4C6D-B6BE-BABD8FC0981A})

Encapsulation type: Ethernet (1)

Arrival Time: May 23, 2025 11:47:31.212145000 RTZ 2 (зима)

UTC Arrival Time: May 23, 2025 08:47:31.212145000 UTC

Epoch Arrival Time: 1747990051.212145000

[Time shift for this packet: 0.000000000 seconds]

[Time delta from previous captured frame: 0.495961000 seconds]

[Time delta from previous displayed frame: 1.001133000 seconds]

[Time since reference or first frame: 413.901153000 seconds]

Frame Number: 91703

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: Intel_14:ff:ff (0c:dd:24:14:ff:ff), Dst: TendaTechnol_23:57:50 (b8:3a:08:23:57:50)

> Destination: TendaTechnol_23:57:50 (b8:3a:08:23:57:50)

> Source: Intel_14:ff:ff (0c:dd:24:14:ff:ff)

Type: IPv4 (0x0800)

▼ Internet Protocol Version 4, Src: 192.168.0.195, Dst: 5.101.152.33

0100 = Version: 4

.... 0101 = Header Length: 20 bytes (5)

> Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)

Total Length: 60

Identification: 0x7e1f (32287)

> 000. = Flags: 0x0

...0 0000 0000 0000 = Fragment Offset: 0

Time to Live: 128

Protocol: ICMP (1)

Header Checksum: 0x5db0 [validation disabled]

[Header checksum status: Unverified]

Source Address: 192.168.0.195

Destination Address: 5.101.152.33

▼ Internet Control Message Protocol

Type: 8 (Echo (ping) request)

Code: 0

Checksum: 0x4cfb [correct]

[Checksum Status: Good]

Identifier (BE): 1 (0x0001)

Identifier (LE): 256 (0x0100)

Sequence Number (BE): 96 (0x0060)

Sequence Number (LE): 24576 (0x6000)

[\[Response frame: 91704\]](#)

> Data (32 bytes)

0000 b8 3a 08 23 57 50 0c dd 24 14 ff ff 08 00 45 00 ··:·#WP·· \$·.....E·

0010 00 3c 7e 1f 00 00 80 01 5d b0 c0 a8 00 c3 05 65 ·<~.....].....e

0020 98 21 08 00 4c fb 00 01 00 60 61 62 63 64 65 66 ·!·..L... ·`abcdef

0030 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76 ghijklmn opqrstuv

0040 77 61 62 63 64 65 66 67 68 69 wabcdefg hi

Ответ ICMP

```

▼ Frame 91704: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface \Device\NPF_{E02C3A3B-DFDF-4C6D-B6BE-BABD8FC0981A}
  Section number: 1
  > Interface id: 0 (\Device\NPF_{E02C3A3B-DFDF-4C6D-B6BE-BABD8FC0981A})
  Encapsulation type: Ethernet (1)
  Arrival Time: May 23, 2025 11:47:31.232489000 RTZ 2 (зима)
  UTC Arrival Time: May 23, 2025 08:47:31.232489000 UTC
  Epoch Arrival Time: 1747990051.232489000
  [Time shift for this packet: 0.000000000 seconds]
  [Time delta from previous captured frame: 0.020344000 seconds]
  [Time delta from previous displayed frame: 0.020344000 seconds]
  [Time since reference or first frame: 413.921497000 seconds]
  Frame Number: 91704
  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: TendaTechnol_23:57:50 (b8:3a:08:23:57:50), Dst: Intel_14:ff:ff (0c:dd:24:14:ff:ff)
  > Destination: Intel_14:ff:ff (0c:dd:24:14:ff:ff)
  > Source: TendaTechnol_23:57:50 (b8:3a:08:23:57:50)
  Type: IPv4 (0x0800)
▼ Internet Protocol Version 4, Src: 5.101.152.33, Dst: 192.168.0.195
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
  Total Length: 60
  Identification: 0x1abc (6844)
  > 000. .... = Flags: 0x0
  ...0 0000 0000 0000 = Fragment Offset: 0
  Time to Live: 57
  Protocol: ICMP (1)
  Header Checksum: 0x0814 [validation disabled]
  [Header checksum status: Unverified]
  Source Address: 5.101.152.33
  Destination Address: 192.168.0.195
▼ Internet Control Message Protocol
  Type: 0 (Echo (ping) reply)
  Code: 0
  Checksum: 0x54fb [correct]
  [Checksum Status: Good]
  Identifier (BE): 1 (0x0001)
  Identifier (LE): 256 (0x0100)
  Sequence Number (BE): 96 (0x0060)
  Sequence Number (LE): 24576 (0x6000)
  [Request frame: 91703]
  [Response time: 20,344 ms]
  > Data (32 bytes)
```

| | | |
|------|---|-------------------|
| 0000 | 0c dd 24 14 ff ff b8 3a 08 23 57 50 08 00 45 00 | --\$---: --NP--E- |
| 0010 | 00 3c 1a bc 00 00 39 01 08 14 05 65 98 21 c0 a8 | -<---9- ---e!.. |
| 0020 | 00 c3 00 00 54 fb 00 01 00 60 61 62 63 64 65 66 | ----T---`abcdef |
| 0030 | 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76 | ghijklmn opqrstuv |
| 0040 | 77 61 62 63 64 65 66 67 68 69 | wabcdefg hi |

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: 192.168.0.100, Dst: 5.101.152.33
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
▼ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
  0000 00.. = Differentiated Services Codepoint: Default (0)
  .... ..00 = Explicit Congestion Notification: Not ECN-Capable Transport (0)
Total Length: 548
Identification: 0xa1b0 (41392)
▼ Flags: 0x00
  0... .... = Reserved bit: Not set
  .0.. .... = Don't fragment: Not set
  ..0. .... = More fragments: Not set
  ...0 0101 1100 1000 = Fragment Offset: 1480
Time to Live: 128
Protocol: ICMP (1)
Header Checksum: 0x0000 [validation disabled]
[Header checksum status: Unverified]
Source Address: 192.168.0.100
Destination Address: 5.101.152.33

```

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

| | | | | | | |
|------|-----------|---------------|---------------|------|---|---|
| 2929 | 56.084685 | 5.101.152.33 | 192.168.0.100 | ICMP | 562 Echo (ping) reply | id=0x0001, seq=45/11520, ttl=57 (request in 2927) |
| 3264 | 67.074349 | 192.168.0.100 | 5.101.152.33 | IPv4 | 1514 Fragmented IP protocol (proto=ICMP 1, off=0, ID=a1b1) | [Reassembled in #3267] |
| 3265 | 67.074349 | 192.168.0.100 | 5.101.152.33 | IPv4 | 1514 Fragmented IP protocol (proto=ICMP 1, off=1480, ID=a1b1) | [Reassembled in #3267] |
| 3266 | 67.074349 | 192.168.0.100 | 5.101.152.33 | IPv4 | 1514 Fragmented IP protocol (proto=ICMP 1, off=2960, ID=a1b1) | [Reassembled in #3267] |

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

Флаг MF = 1 (промежуточный фрагмент)

```

▼ Flags: 0x20, More fragments
  0... .... = Reserved bit: Not set
  .0.. .... = Don't fragment: Not set
  ..1. .... = More fragments: Set
  ...0 0101 1100 1000 = Fragment Offset: 1480

```

Флаг MF = 0 (последний фрагмент)

```

▼ Flags: 0x00
  0... .... = Reserved bit: Not set
  .0.. .... = Don't fragment: Not set
  ..0. .... = More fragments: Not set

```

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

Размер пакета / 1480 и округлить до верхнего целого числа.

| | | | | | | |
|------|-----------|---------------|---------------|------|---|---|
| 1862 | 33.689834 | 192.168.0.100 | 5.101.152.33 | ICMP | 142 Echo (ping) request | id=0x0001, seq=41/10496, ttl=128 (reply in 1863) |
| 1863 | 33.711208 | 5.101.152.33 | 192.168.0.100 | ICMP | 142 Echo (ping) reply | id=0x0001, seq=41/10496, ttl=57 (request in 1862) |
| 2238 | 41.065961 | 192.168.0.100 | 5.101.152.33 | ICMP | 242 Echo (ping) request | id=0x0001, seq=42/10752, ttl=128 (reply in 2239) |
| 2239 | 41.085447 | 5.101.152.33 | 192.168.0.100 | ICMP | 242 Echo (ping) reply | id=0x0001, seq=42/10752, ttl=57 (request in 2238) |
| 2453 | 47.398534 | 192.168.0.100 | 5.101.152.33 | ICMP | 542 Echo (ping) request | id=0x0001, seq=43/11008, ttl=128 (reply in 2454) |
| 2454 | 47.418345 | 5.101.152.33 | 192.168.0.100 | ICMP | 542 Echo (ping) reply | id=0x0001, seq=43/11008, ttl=57 (request in 2453) |
| 2768 | 51.065026 | 192.168.0.100 | 5.101.152.33 | ICMP | 1042 Echo (ping) request | id=0x0001, seq=44/11264, ttl=128 (reply in 2770) |
| 2770 | 51.084730 | 5.101.152.33 | 192.168.0.100 | ICMP | 1042 Echo (ping) reply | id=0x0001, seq=44/11264, ttl=57 (request in 2768) |
| 2926 | 56.064854 | 192.168.0.100 | 5.101.152.33 | IPv4 | 1514 Fragmented IP protocol (proto=ICMP 1, off=0, ID=a1b0) | [Reassembled in #2927] |
| 2927 | 56.064854 | 192.168.0.100 | 5.101.152.33 | ICMP | 562 Echo (ping) request | id=0x0001, seq=45/11520, ttl=128 (reply in 2929) |
| 2928 | 56.084685 | 5.101.152.33 | 192.168.0.100 | IPv4 | 1514 Fragmented IP protocol (proto=ICMP 1, off=0, ID=4bc8) | [Reassembled in #2929] |
| 2929 | 56.084685 | 5.101.152.33 | 192.168.0.100 | ICMP | 562 Echo (ping) reply | id=0x0001, seq=45/11520, ttl=57 (request in 2927) |
| 3264 | 67.074349 | 192.168.0.100 | 5.101.152.33 | IPv4 | 1514 Fragmented IP protocol (proto=ICMP 1, off=0, ID=a1b1) | [Reassembled in #3267] |
| 3265 | 67.074349 | 192.168.0.100 | 5.101.152.33 | IPv4 | 1514 Fragmented IP protocol (proto=ICMP 1, off=1480, ID=a1b1) | [Reassembled in #3267] |
| 3266 | 67.074349 | 192.168.0.100 | 5.101.152.33 | IPv4 | 1514 Fragmented IP protocol (proto=ICMP 1, off=2960, ID=a1b1) | [Reassembled in #3267] |
| 3267 | 67.074349 | 192.168.0.100 | 5.101.152.33 | ICMP | 602 Echo (ping) request | id=0x0001, seq=46/11776, ttl=128 (reply in 3271) |
| 3268 | 67.094334 | 5.101.152.33 | 192.168.0.100 | IPv4 | 1514 Fragmented IP protocol (proto=ICMP 1, off=0, ID=4d4c) | [Reassembled in #3271] |
| 3269 | 67.094378 | 5.101.152.33 | 192.168.0.100 | IPv4 | 1514 Fragmented IP protocol (proto=ICMP 1, off=1480, ID=4d4c) | [Reassembled in #3271] |
| 3270 | 67.094378 | 5.101.152.33 | 192.168.0.100 | IPv4 | 1514 Fragmented IP protocol (proto=ICMP 1, off=2960, ID=4d4c) | [Reassembled in #3271] |
| 3271 | 67.094378 | 5.101.152.33 | 192.168.0.100 | ICMP | 602 Echo (ping) reply | id=0x0001, seq=46/11776, ttl=57 (request in 3267) |
| 3752 | 75.295752 | 192.168.0.100 | 5.101.152.33 | IPv4 | 1514 Fragmented IP protocol (proto=ICMP 1, off=0, ID=a1b2) | [Reassembled in #3758] |
| 3753 | 75.295752 | 192.168.0.100 | 5.101.152.33 | IPv4 | 1514 Fragmented IP protocol (proto=ICMP 1, off=1480, ID=a1b2) | [Reassembled in #3758] |
| 3754 | 75.295752 | 192.168.0.100 | 5.101.152.33 | IPv4 | 1514 Fragmented IP protocol (proto=ICMP 1, off=2960, ID=a1b2) | [Reassembled in #3758] |
| 3755 | 75.295752 | 192.168.0.100 | 5.101.152.33 | IPv4 | 1514 Fragmented IP protocol (proto=ICMP 1, off=4440, ID=a1b2) | [Reassembled in #3758] |
| 3756 | 75.295752 | 192.168.0.100 | 5.101.152.33 | IPv4 | 1514 Fragmented IP protocol (proto=ICMP 1, off=5920, ID=a1b2) | [Reassembled in #3758] |
| 3757 | 75.295752 | 192.168.0.100 | 5.101.152.33 | IPv4 | 1514 Fragmented IP protocol (proto=ICMP 1, off=7400, ID=a1b2) | [Reassembled in #3758] |
| 3758 | 75.295752 | 192.168.0.100 | 5.101.152.33 | ICMP | 1162 Echo (ping) request | id=0x0001, seq=47/12032, ttl=128 (no response found!) |

4. График: размер пакета – кол-во фрагментов.



5. Как изменить поле TTL с помощью утилиты ping?

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

Time to Live: 128

Protocol: ICMP (1)

Header Checksum: 0x0000 [validation disabled]

6. Что содержится в поле данных ping-пакета?

Заголовок ICMP

Идентификатор

Номер последовательности

Содержимое

Анализ утилиты tracert

Команда: tracert vekshin.ru

Фильтр: dns

```
C:\Users\Арсений>tracert vekshin.ru

Tracing route to vekshin.ru [5.101.152.33]
over a maximum of 30 hops:

  1      1 ms      1 ms      2 ms  192.168.0.1
  2     14 ms     28 ms     14 ms  5x19x0x106.static-business.spb.ertelecom.ru [5.19.0.106]
  3      4 ms      3 ms      3 ms  5x19x0x250.static-business.spb.ertelecom.ru [5.19.0.250]
  4      4 ms      3 ms      3 ms  bbr03.spb.ertelecom.ru [188.234.152.203]
  5      3 ms      3 ms      3 ms  188-234-140-21.ertelecom.ru [188.234.140.21]
  6      *        *        *    Request timed out.
  7     12 ms     12 ms     12 ms  80.64.101.183.rascom.as20764.net [80.64.101.183]
  8     20 ms     21 ms     20 ms  10.255.200.37
  9     24 ms     43 ms     20 ms  m2.dale.beget.com [5.101.152.33]

Trace complete.
```

```
C:\Users\Арсений>tracert -d vekshin.ru

Tracing route to vekshin.ru [5.101.152.33]
over a maximum of 30 hops:

  1      1 ms      2 ms      2 ms  192.168.0.1
  2      5 ms      3 ms     25 ms  5.19.0.110
  3      3 ms      3 ms      3 ms  5.19.0.250
  4      3 ms      3 ms      3 ms  188.234.152.203
  5      3 ms      2 ms      3 ms  188.234.140.21
  6      *        *        *    Request timed out.
  7     11 ms     11 ms     12 ms  80.64.101.183
  8     20 ms     20 ms     20 ms  10.255.200.37
  9     20 ms     20 ms     21 ms  5.101.152.33
```


Структура dns-пакета

```

Domain Name System (query)
  Transaction ID: 0x4e11
  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
    vekshin.ru: type A, class IN
      Name: vekshin.ru
      [Name Length: 10]
      [Label Count: 2]
      Type: A (Host Address) (1)
      Class: IN (0x0001)
\[Response In: 13645\]
```

Ответы на вопросы:

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

```

Internet Protocol Version 4, Src: 192.168.0.100, Dst: 192.168.0.1
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
  Total Length: 56
  Identification: 0xf1ea (61930)
```

2. Как и почему изменяется поле TTL в следующих ICMP-пакетах tracer? Утилита tracer посылает 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 отправляет ICMP echo request, а ожидает ICMP echo reply

tracer использует ICMP Echo Request с разным TTL и анализирует:

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

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 tracer:

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

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

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

Анализ HTTP-трафика

Сайт по варианту отклоняет входящие GET-запросы. Для демонстрации будет использован сайт example.com

Команда-фильтр: http.request or http.response

Первый вход на сайт

| | | | | | | |
|---|------|-----------|---------------|---------------|------|-------------------------------|
| → | 3440 | 40.288746 | 192.168.0.100 | 23.192.228.80 | HTTP | 535 GET / HTTP/1.1 |
| ← | 3445 | 40.481516 | 23.192.228.80 | 192.168.0.100 | HTTP | 304 HTTP/1.1 304 Not Modified |
| | 3522 | 45.128244 | 192.168.0.1 | 192.168.0.100 | HTTP | 60 HTTP/1.1 200 OK |

Обновим страницу

| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|------|-----------|----------------|----------------|--------|---|
| → | 3440 | 40.288746 | 192.168.0.100 | 23.192.228.80 | HTTP | 535 GET / HTTP/1.1 |
| ← | 3445 | 40.481516 | 23.192.228.80 | 192.168.0.100 | HTTP | 304 HTTP/1.1 304 Not Modified |
| | 3522 | 45.128244 | 192.168.0.1 | 192.168.0.100 | HTTP | 60 HTTP/1.1 200 OK |
| | 3531 | 45.143712 | 192.168.0.1 | 192.168.0.100 | HTTP | 60 HTTP/1.1 200 OK |
| → | 4917 | 79.079188 | 192.168.0.100 | 23.192.228.80 | HTTP | 535 GET / HTTP/1.1 |
| | 4919 | 79.271996 | 23.192.228.80 | 192.168.0.100 | HTTP | 304 HTTP/1.1 304 Not Modified |
| | 4920 | 79.294701 | 192.168.0.100 | 23.192.228.80 | HTTP | 430 GET /favicon.ico HTTP/1.1 |
| | 4935 | 79.541204 | 23.192.228.80 | 192.168.0.100 | HTTP | 242 HTTP/1.1 404 Not Found (text/html) |
| | 5272 | 88.713459 | 192.168.0.100 | 151.101.38.172 | HTTP | 341 GET /msdownload/update/v3/static/trustedr/en/disallow |
| | 5274 | 88.756521 | 151.101.38.172 | 192.168.0.100 | HTTP | 253 HTTP/1.1 304 Not Modified |
| | 5305 | 89.815839 | 192.168.0.100 | 151.101.38.172 | HTTP | 341 GET /msdownload/update/v3/static/trustedr/en/disallow |
| | 5307 | 89.859165 | 151.101.38.172 | 192.168.0.100 | HTTP | 255 HTTP/1.1 304 Not Modified |

Тело запроса Get

```
▼ Hypertext Transfer Protocol
  > GET /msdownload/update/v3/static/trustedr/en/disallowedcertstl.cab?ff2938d5fe737038 HTTP/1.1\r\n
    Connection: Keep-Alive\r\n
    Accept: */*\r\n
    If-Modified-Since: Thu, 05 Dec 2024 19:42:09 GMT\r\n
    If-None-Match: "06cfcc54d47db1:0"\r\n
    User-Agent: Microsoft-CryptoAPI/10.0\r\n
    Host: ctldl.windowsupdate.com\r\n
    \r\n
    [Full request URI: http://ctldl.windowsupdate.com/msdownload/update/v3/static/trustedr/en/disallow
    [HTTP request 1/1]
    [Response in frame: 5274]
```

Тело ответа с кодом 304

```
> Transmission Control Protocol, Src Port: 80, Dst Port: 60927, Seq: 1, Ack: 288, Len: 199
▼ Hypertext Transfer Protocol
  > HTTP/1.1 304 Not Modified\r\n
    Connection: keep-alive\r\n
    Date: Thu, 29 May 2025 22:13:29 GMT\r\n
    Via: 1.1 varnish\r\n
    X-Varnish: 50710064\r\n
    Cache-Control: public,max-age=900\r\n
    ETag: "06cfcc54d47db1:0"\r\n
    Age: 92\r\n
    \r\n
    [HTTP response 1/1]
    [Time since request: 0.043062000 seconds]
    [Request in frame: 5272]
    [Request URI: http://ctldl.windowsupdate.com/msdownload/update/v3/static/trustedr/en/disallowedc
```

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

Фильтр: arp

Для начала очистим ARP-таблицу

```
C:\Windows\system32>netsh interface ip delete arpcache
OK.

C:\Windows\system32>arp -a

Интерфейс: 192.168.56.1 --- 0x7
    адрес в Интернете      Физический адрес      Тип
    224.0.0.22              01-00-5e-00-00-16     статический

Интерфейс: 192.168.0.100 --- 0xe
    адрес в Интернете      Физический адрес      Тип
    192.168.0.1             b8-3a-08-23-57-50     динамический
    224.0.0.2               01-00-5e-00-00-02     статический
    224.0.0.22              01-00-5e-00-00-16     статический
    224.0.0.251             01-00-5e-00-00-fb     статический
    224.0.0.252             01-00-5e-00-00-fc     статический
    239.255.102.18          01-00-5e-7f-66-12     статический

Интерфейс: 172.26.96.1 --- 0x4d
    адрес в Интернете      Физический адрес      Тип
    224.0.0.22              01-00-5e-00-00-16     статический
```

Зайдем на сайт и проверим новую запись в таблице

```
C:\Windows\system32>arp -a

Интерфейс: 192.168.56.1 --- 0x7
    адрес в Интернете      Физический адрес      Тип
    224.0.0.22              01-00-5e-00-00-16     статический

Интерфейс: 192.168.0.100 --- 0xe
    адрес в Интернете      Физический адрес      Тип
    192.168.0.1             b8-3a-08-23-57-50     динамический
    224.0.0.2               01-00-5e-00-00-02     статический
    224.0.0.22              01-00-5e-00-00-16     статический
    224.0.0.251             01-00-5e-00-00-fb     статический
    224.0.0.252             01-00-5e-00-00-fc     статический
    239.255.102.18          01-00-5e-7f-66-12     статический
    → 239.255.255.250       01-00-5e-7f-ff-fa     статический

Интерфейс: 172.26.96.1 --- 0x4d
    адрес в Интернете      Физический адрес      Тип
    224.0.0.22              01-00-5e-00-00-16     статический
```

Логи Wireshark

| No. | Time | Source | Destination | Protocol | Length | Info |
|------|------------|-------------------|-------------------|----------|--------|---|
| 28 | 4.807245 | TendaTec_23:57... | Broadcast | ARP | 60 | Who has 192.168.0.100? Tell 192.168.0.1 |
| 29 | 4.807260 | Giga-Byt_da:47... | TendaTec_23:57... | ARP | 42 | 192.168.0.100 is at 18:c0:4d:da:47:dd |
| 4866 | 124.465637 | TendaTec_23:57... | Broadcast | ARP | 60 | Who has 192.168.0.100? Tell 192.168.0.1 |
| 4867 | 124.465653 | Giga-Byt_da:47... | TendaTec_23:57... | ARP | 42 | 192.168.0.100 is at 18:c0:4d:da:47:dd |
| 6565 | 154.519275 | Giga-Byt_da:47... | Broadcast | ARP | 42 | Who has 192.168.0.1? Tell 192.168.0.100 |
| 6566 | 154.520030 | TendaTec_23:57... | Giga-Byt_da:47... | ARP | 60 | 192.168.0.1 is at b8:3a:08:23:57:50 |
| 9731 | 274.090479 | TendaTec_23:57... | Broadcast | ARP | 60 | Who has 192.168.0.100? Tell 192.168.0.1 |
| 9732 | 274.090488 | Giga-Byt_da:47... | TendaTec_23:57... | ARP | 42 | 192.168.0.100 is at 18:c0:4d:da:47:dd |

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: Giga-Byt_da:47:dd (18:c0:4d:da:47:dd),
 > Destination: TendaTec_23:57:50 (b8:3a:08:23:57:50)
 > Source: Giga-Byt_da:47:dd (18:c0:4d:da:47:dd)
 Type: ARP (0x0806)

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

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