

Санкт-Петербургский Политехнический Университет Петра Великого
Институт компьютерных наук и технологий
Кафедра компьютерных систем и программных технологий

Защита информации

Отчет по лабораторной работе №1

Исследование сетевого трафика

Работу выполнил:
Раскин Андрей
Группа: 43501/3
Преподаватель:
Новопашенный Андрей Гелиевич

Санкт-Петербург
2017

1 Цель работы

Закрепление навыков работы в программе Wireshark и знаний о некоторых сетевых протоколах.

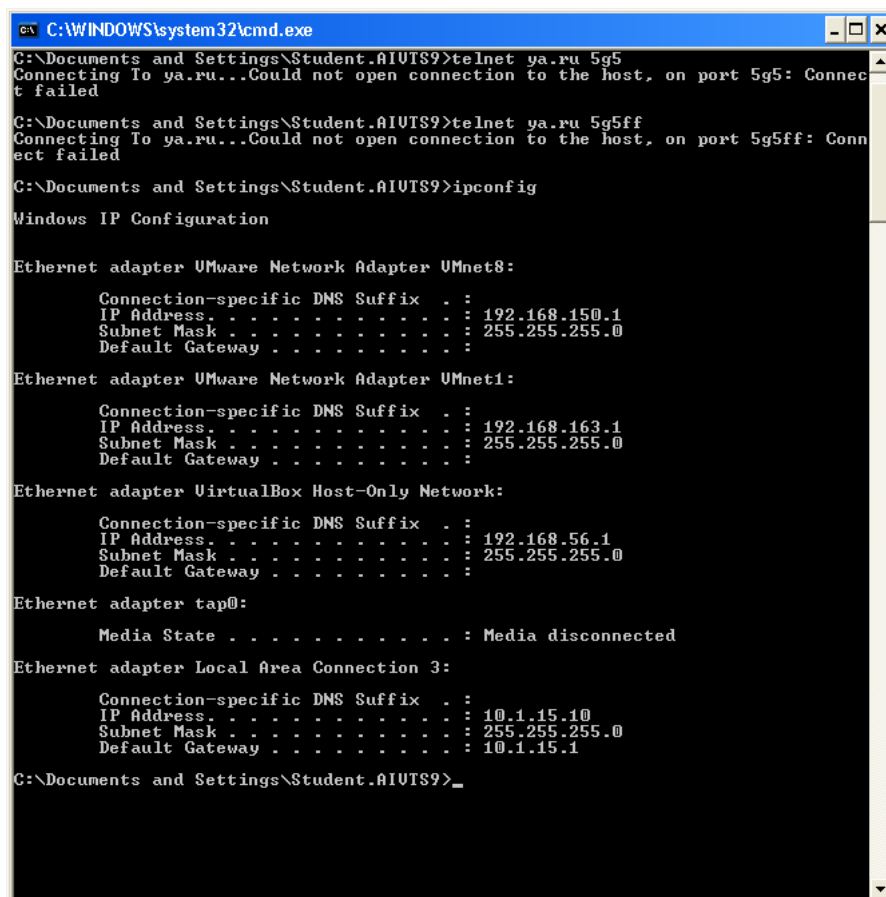
2 Программа работы

При помощи анализатора сетевого трафика Wireshark продемонстрировать в сети:

1. Работу утилиты ping
2. Работу утилиты tracert
3. Работу ICMP-протокола в следующих ситуациях:
 - Отправка фрагментированного ping'a,
 - Получение ошибки 3.1 (Destination host unreachable)
4. Работу ARP-протокола (запрос и ответ);
5. Работу протокола TCP в следующих ситуациях:
 - Установка соединения,
 - Разрыв соединения,
 - Попытка соединения на отсутствующий порт

3 Конфигурация компьютера в сети

1



```

C:\WINDOWS\system32\cmd.exe
C:\Documents and Settings\Student.AIUTS9>telnet ya.ru 5g5
Connecting To ya.ru...Could not open connection to the host, on port 5g5: Connection failed

C:\Documents and Settings\Student.AIUTS9>telnet ya.ru 5g5ff
Connecting To ya.ru...Could not open connection to the host, on port 5g5ff: Connection failed

C:\Documents and Settings\Student.AIUTS9>ipconfig

Windows IP Configuration

Ethernet adapter VMware Network Adapter VMnet8:

    Connection-specific DNS Suffix  . : 
    IP Address. . . . .               : 192.168.150.1
    Subnet Mask . . . . .             : 255.255.255.0
    Default Gateway . . . . .         : 

Ethernet adapter VMware Network Adapter VMnet1:

    Connection-specific DNS Suffix  . : 
    IP Address. . . . .               : 192.168.163.1
    Subnet Mask . . . . .             : 255.255.255.0
    Default Gateway . . . . .         : 

Ethernet adapter VirtualBox Host-Only Network:

    Connection-specific DNS Suffix  . : 
    IP Address. . . . .               : 192.168.56.1
    Subnet Mask . . . . .             : 255.255.255.0
    Default Gateway . . . . .         : 

Ethernet adapter tap0:

    Media State . . . . .              : Media disconnected

Ethernet adapter Local Area Connection 3:

    Connection-specific DNS Suffix  . : 
    IP Address. . . . .               : 10.1.15.10
    Subnet Mask . . . . .             : 255.255.255.0
    Default Gateway . . . . .         : 10.1.15.1

C:\Documents and Settings\Student.AIUTS9>_

```

Рис. 1: Конфигурация сети

4 Ход работы

4.1 Работы утилиты ping

Ping — утилита для проверки целостности и качества соединений в сетях на основе TCP/IP, а также обиходное наименование самого запроса. Утилита отправляет запросы (ICMP Echo-Request) протокола ICMP указанному узлу сети и фиксирует поступающие ответы (ICMP Echo-Reply). По умолчанию производится 4 попытки отправки запроса.

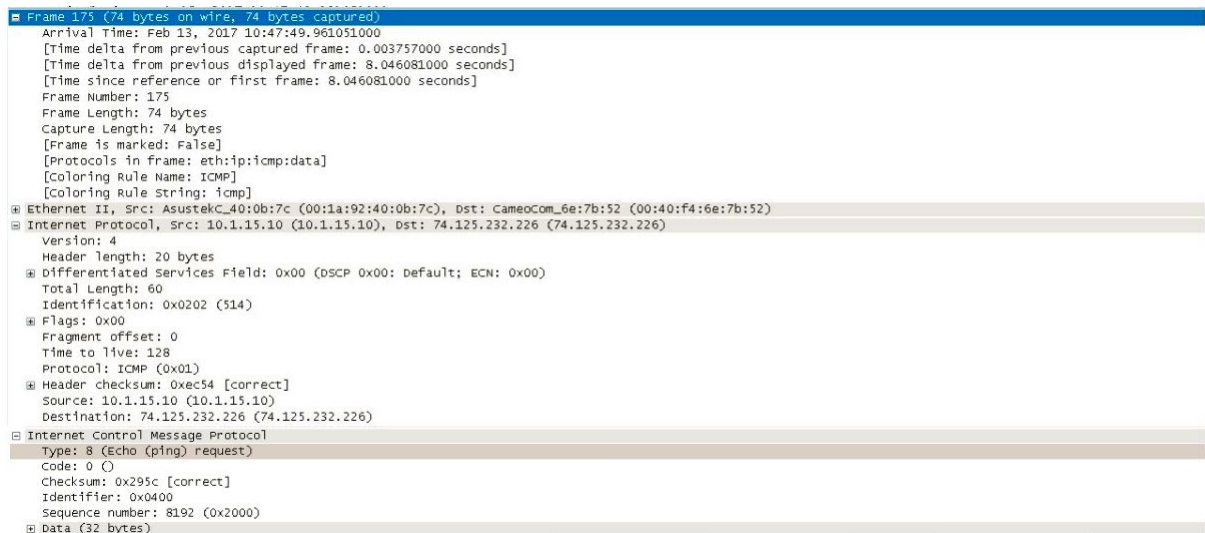


Рис. 2: ICMP эхо запрос

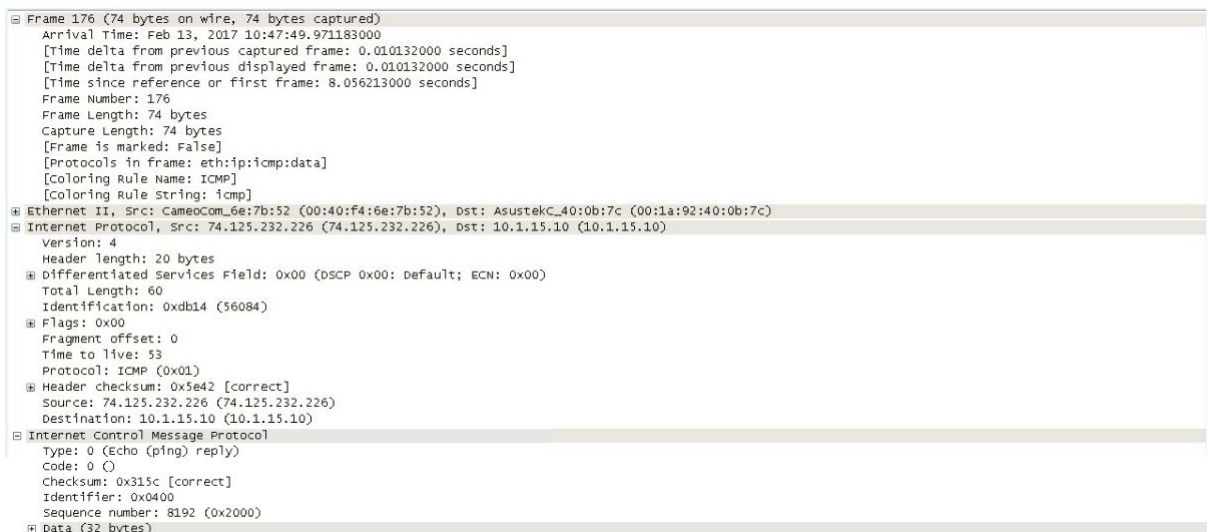


Рис. 3: ICMP эхо ответ

Как видно, в поле Destination указан IP-адрес google.com, поле Source показывает IP-адрес текущего компьютера. Тип сообщения равный 8 означает эхо-запрос, а тип 0 означает эхо-ответ.

4.2 Работа утилиты tracer

В основе работы данной утилиты лежит протокол icmp. Команда TRACERT определяет путь до точки назначения с помощью отправки в точку назначения эхо-сообщений протокола Control Message Protocol (ICMP) с постоянным увеличением значений срока жизни (Time to Live, TTL). Попробуем пронаблюдать трассировку маршрута пакетов до узла spbstu.ru при помощи протокола ICMP и утилиты tracer.

Первый пакет трассировки маршрута отправляется с TTL равным 1. Это значит, что на первом же маршрутизаторе пакет будет уничтожен и нам придет сообщение об ошибке.

```
C:\Users\Georgiy>tracert spbstu.ru

Трассировка маршрута к spbstu.ru [195.209.230.198]
с максимальным числом прыжков 30:

  1    13 ms    5 ms    6 ms  192.168.1.1
  2    6 ms    6 ms   14 ms  192.168.25.2
  3    6 ms    4 ms    7 ms  192.168.25.2
  4    3 ms    3 ms   30 ms  192.168.25.2
  5    7 ms    6 ms    4 ms  192.168.25.2
  6    3 ms    3 ms    4 ms  192.168.25.2
  7   78 ms   123 ms   5 ms  192.168.25.2
  8    4 ms    4 ms    6 ms  192.168.25.2
  9    4 ms    3 ms    4 ms  192.168.25.2
```

Рис. 4: Результат трассировки маршрута в консоли

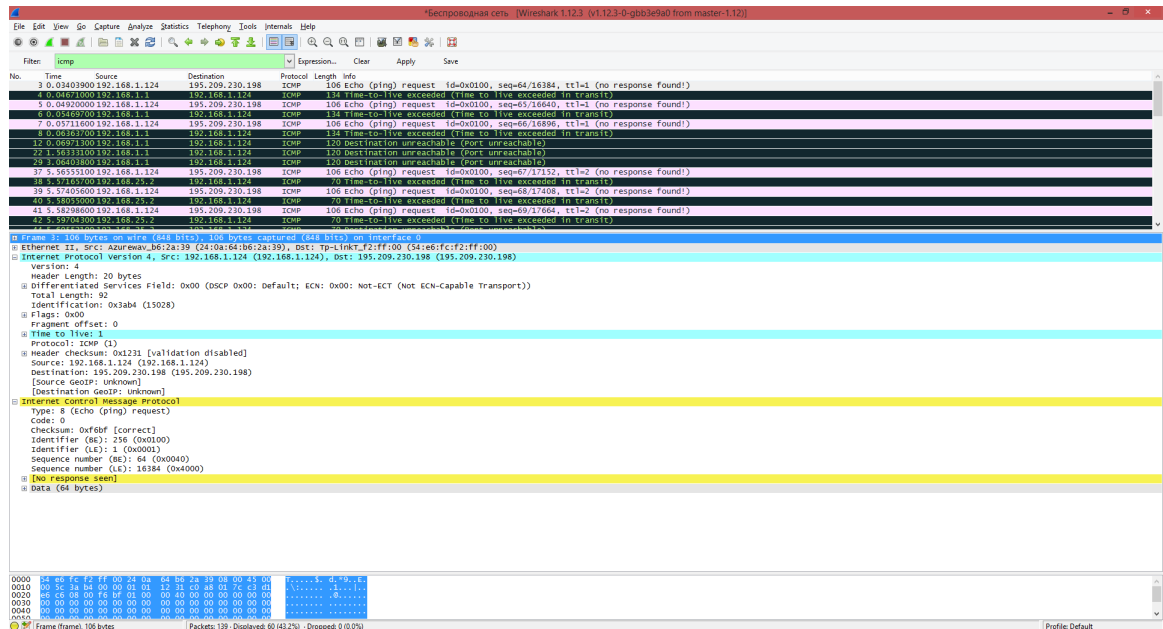


Рис. 5: Первый пакет трассировки маршрута

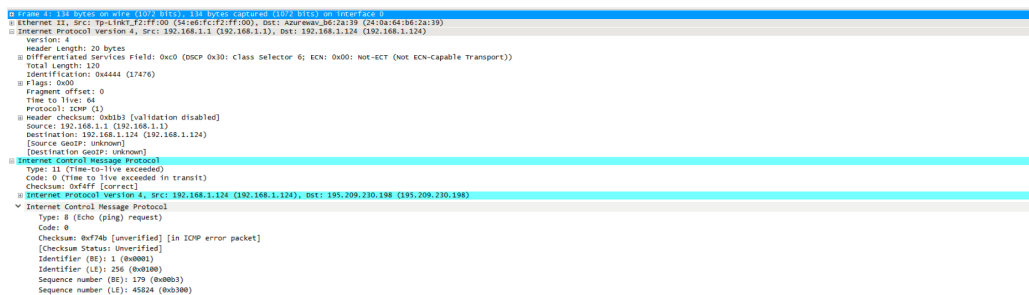


Рис. 6: Ответ на первый пакет трассировки

В сообщении об ошибке указан тип ICMP-пакета – 11.0, что означает, что время жизни пакета истекло. Сообщение пришло от маршрутизатора сети, который имеет адрес 192.168.1.1. Аналогично продолжается трассировка маршрута дальше с постепенным инкрементом параметра TTL.

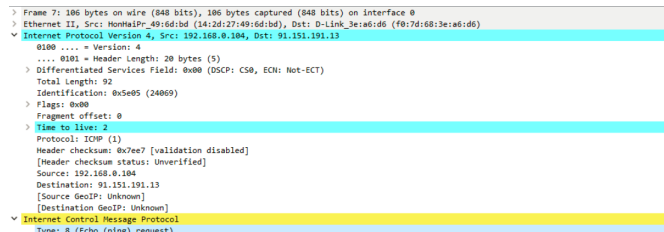


Рис. 7: Второй пакет трассировки

Таким образом составляется примерный маршрут прохождения IP-пакета до узла с адресом spbstu.ru.

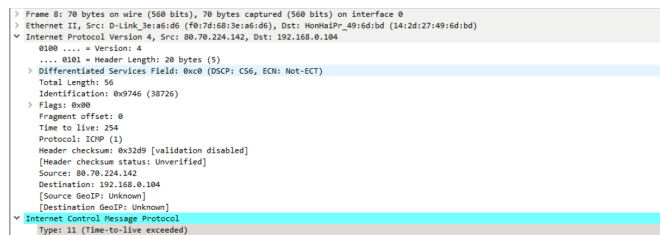


Рис. 8: Ответ на второй пакет трассировки

4.3 Протокол ICMP

4.3.1 Фрагментированный ping

Попробуем отослать фрагментированный ping-запрос. Данный вид запроса использует ICMP-протокол. Для фрагментации пакета необходимо указать его размер, превышающий MTU (maximum transmission unit) - максимальный размер полезного блока данных одного пакета, который может быть передан протоколом без фрагментации. Для протокола Ethernet обычно это чуть больше 1500 байт. Для фрагментации пакета на 3 части укажем размер – 4000 байт.

25	3.00459300	192.168.1.3	8.8.8.8	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=0, ID=4e44) [Reassembled i
26	3.00461900	192.168.1.3	8.8.8.8	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=4e44) [Reasemble
27	3.00462600	192.168.1.3	8.8.8.8	ICMP	682	Echo (ping) request id=0x0100, seq=28/7168, ttl=128 (reply in 32)

Рис. 9: Сегментирование пакетов

Как видно, первый пакет был отправлен по протоколу ICMP, следующие фрагментированные пакеты передавались по протоколу IPv4 уже без заголовка ICMP.

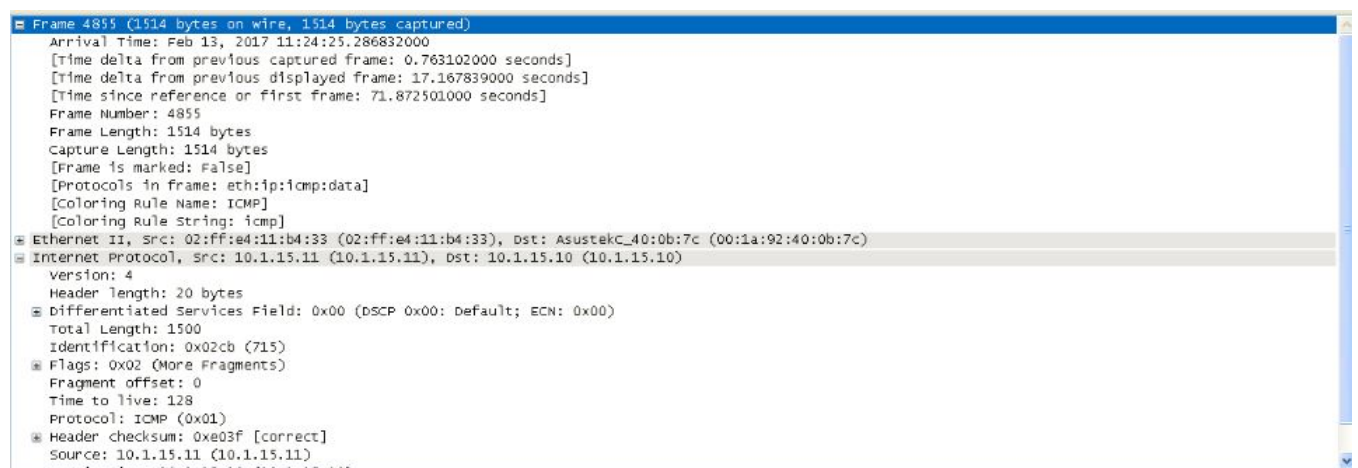


Рис. 10: Первый фрагмент пакета ping-запроса

О фрагментированности пакета свидетельствуют флаги пакета IP (0x01 – имеются еще фрагменты). О том, что это первый пакет из фрагментированных, свидетельствует нулевое смещение фрагмента. При этом во всех трех IP пакетах содержится ICMP-пакет с одним и тем же идентификатором.

Видим, что смещение в данном случае уже ненулевое. Последний пакет выглядит следующим образом:

Здесь флаг, присутствующий в предыдущих пакетах, не установлен, что свидетельствует о том, что фрагмент последний.

4.3.2 Несуществующий хост

Попробуем пронаблюдать ошибку типа 3.1 (целевой узел недостижим). Для этого отправим ping-запрос на адрес, которого не существует. В пакете можно наблюдать типичный ping-запрос (ICMP-пакет типа 8.0).

А вот ответом на указанный выше запрос будет ICMP-пакет типа 3.1, свидетельствующий об ошибке «целевой узел недостижим». При этом, в ответе, в качестве данных пакета отправляется заголовок того пакета, на который пришел ответ.

```

+ Frame 26: 1514 bytes on wire (12112 bits), 1514 bytes captured (12112 bits) on interface 0
+ Ethernet II, Src: IntelCor_9d:6b:3d (4c:eb:42:9d:6b:3d), Dst: D-LinkIn_7c:58:30 (c8:be:19:7c:58:30)
+ Internet Protocol Version 4, Src: 192.168.1.3 (192.168.1.3), Dst: 8.8.8.8 (8.8.8.8)
  Version: 4
  Header Length: 20 bytes
+ Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))
  Total Length: 1500
  Identification: 0x4e44 (20036)
+ Flags: 0x01 (More Fragments)
  Fragment offset: 1480
  Time to live: 128
  Protocol: ICMP (1)
+ Header checksum: 0xf468 [validation disabled]
  Source: 192.168.1.3 (192.168.1.3)
  Destination: 8.8.8.8 (8.8.8.8)
  [Source GeoIP: Unknown]
  [Destination GeoIP: Unknown]
  Reassembled IPv4 in frame: 27
+ Data (1480 bytes)

```

Рис. 11: Второй фрагмент запроса

```

+ Frame 27: 682 bytes on wire (5456 bits), 682 bytes captured (5456 bits) on interface 0
+ Ethernet II, Src: IntelCor_9d:6b:3d (4c:eb:42:9d:6b:3d), Dst: D-LinkIn_7c:58:30 (c8:be:19:7c:58:30)
+ Internet Protocol Version 4, Src: 192.168.1.3 (192.168.1.3), Dst: 8.8.8.8 (8.8.8.8)
  Version: 4
  Header Length: 20 bytes
+ Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))
  Total Length: 668
  Identification: 0x4e44 (20036)
+ Flags: 0x00
  Fragment offset: 2960
  Time to live: 128
  Protocol: ICMP (1)
+ Header checksum: 0x16f0 [validation disabled]
  Source: 192.168.1.3 (192.168.1.3)
  Destination: 8.8.8.8 (8.8.8.8)
  [Source GeoIP: Unknown]
  [Destination GeoIP: Unknown]
+ [3 IPv4 Fragments (3608 bytes): #25(1480), #26(1480), #27(648)]
+ Internet Control Message Protocol
  Type: 8 (Echo (ping) request)
  Code: 0
  Checksum: 0xc0a7 [correct]
  Identifier (BE): 256 (0x0100)
  Identifier (LE): 1 (0x0001)
  Sequence number (BE): 28 (0x001c)
  Sequence number (LE): 7168 (0x1c00)
  \[Response frame: 32\]
+ Data (3600 bytes)

```

Рис. 12: Второй фрагмент запроса

```

+ Ethernet II, Src: AsustekC_40:0b:7c (00:1a:92:40:0b:7c), Dst: CameoCom_6e:7b:52 (00:40:f4:6e:7b:52)
+ Internet Protocol, Src: 10.1.15.10 (10.1.15.10), Dst: 10.1.15.1 (10.1.15.1)
  Version: 4
  Header length: 20 bytes
+ Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
  Total Length: 148
  Identification: 0x19ef (6639)
+ Flags: 0x00
  Fragment offset: 1480
  Time to live: 128
  Protocol: ICMP (0x01)
+ Header checksum: 0xedb4 [correct]
  Source: 10.1.15.10 (10.1.15.10)
  Destination: 10.1.15.1 (10.1.15.1)
+ [IP Fragments (1608 bytes): #5032(1480), #5033(128)]
+ Internet Control Message Protocol
  Type: 8 (Echo (ping) request)
  Code: 0 ()
  Checksum: 0x2150 [correct]
  Identifier: 0x0600
  Sequence number: 7168 (0x1c00)
+ Data (1600 bytes)
0000  00 40 f4 6e 7b 52 00 1a 92 40 0b 7c 08 00 45 00  .@.n{R.. .@.|...E.
0010  00 94 19 ef 00 b9 80 01 ed b4 0a 01 0f 0a 0a 01  .....
0020  0f 01 61 e2 63 64 65 66 67 68 69 6a 6b 6c 6d 6e  ..abcdef ghijklmn
0030  6f 70 71 72 73 74 75 76 77 78 79 7a 7b 7c 7d 7e  ..opqrstuvw xy z[deEa
Frame (162 bytes) Reassembled IPv4 (1608 bytes)

```

Рис. 13: Ping-запрос

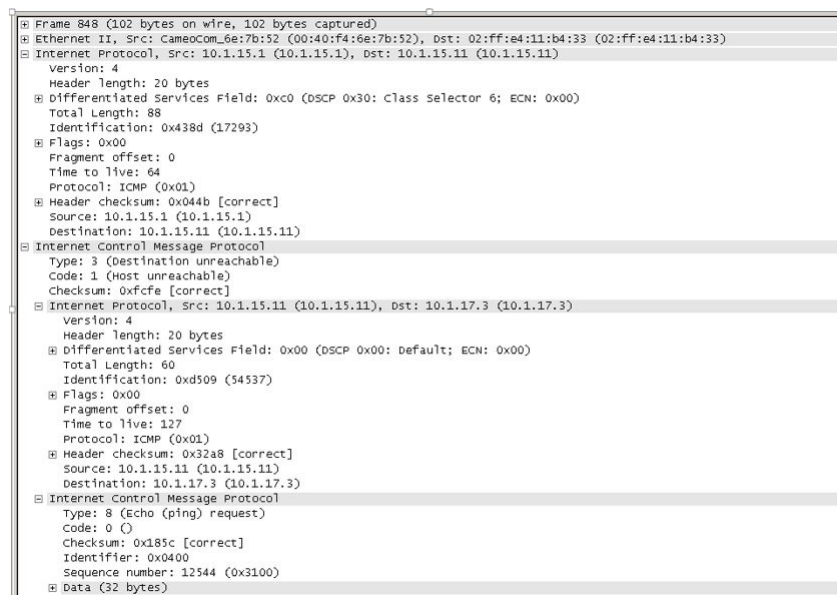


Рис. 14: ICMP-ответ

4.4 ARP протокол

В пакете указывается его тип (поле Opcode) – запрос, а так же целевой IP-адрес для которого запрашивается MAC-адрес. MAC-адрес цели при этом обнулен.

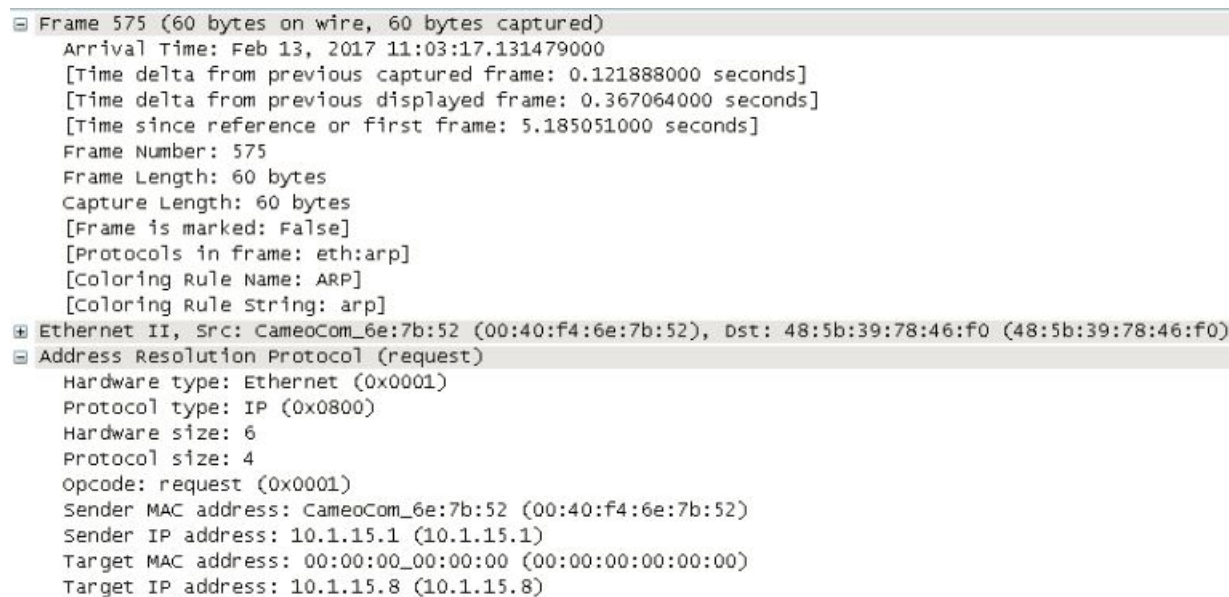


Рис. 15: ARP запрос

ARP-ответ отсылается уже на тот адрес, с которого исходил ARP-запрос. В пакете указывается его тип (поле Opcode) – ответ, а так же заполненный MAC-адрес цели.

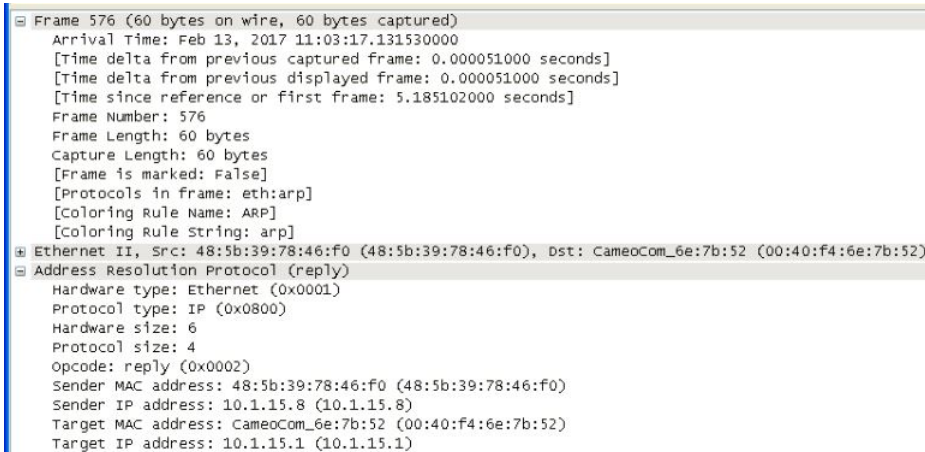


Рис. 16: ARP ответ

4.5 TCP-протокол

4.5.1 Установление соединения

Эта операция происходит следующим образом: Клиент, посылает серверу сегмент с номером последовательности и флагом SYN.

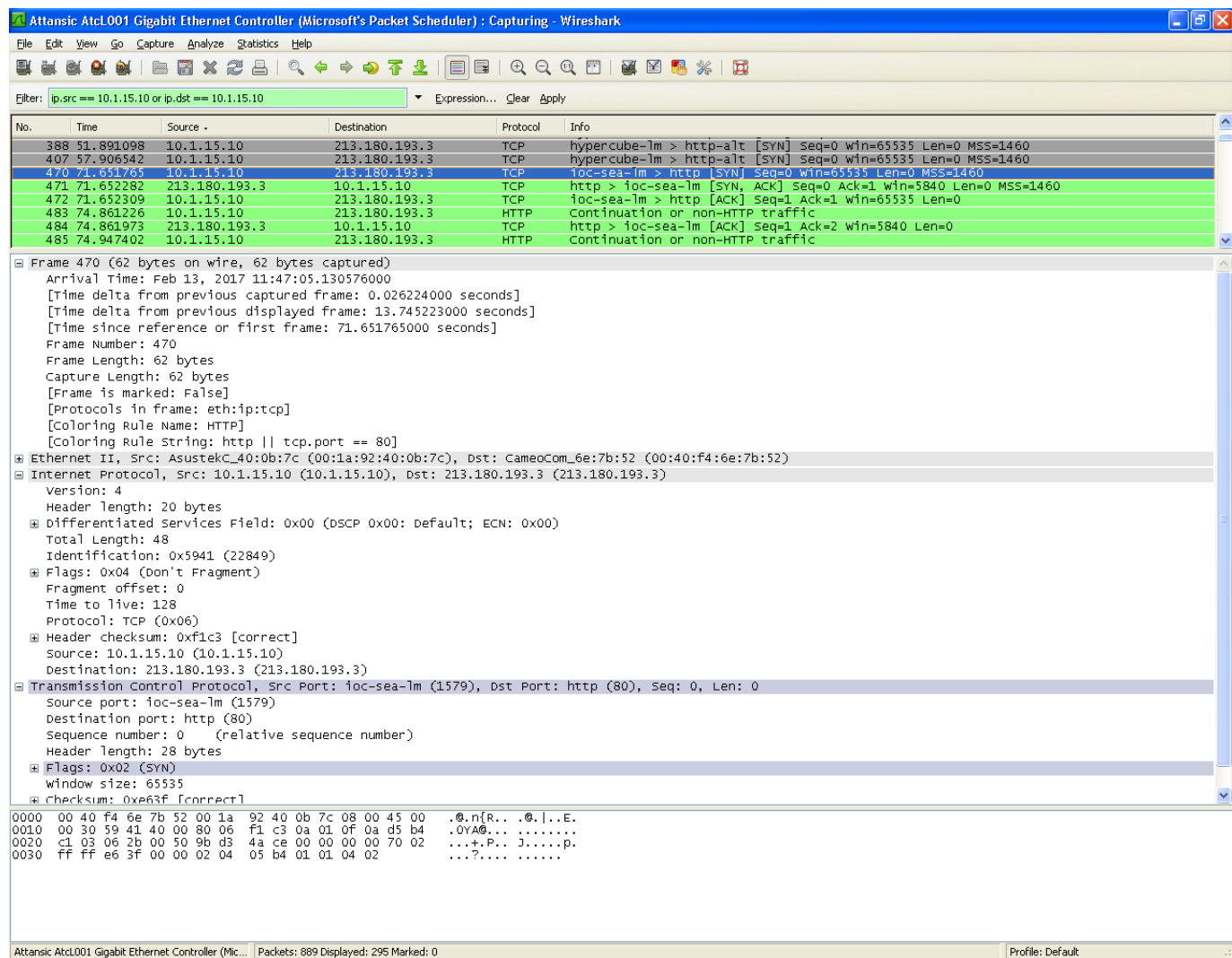


Рис. 17: TCP запрос на установление соединения SYN

В заголовке TCP-пакета можно увидеть следующие поля:

1. Sequence Number - порядковый номер: 32 бита Порядковый номер первого октета данных в сегменте при отсутствии флага SYN. Если в сегменте присутствует бит SYN, поле номера содержит значение начального порядкового номера (ISN), а первый октет данных имеет номер ISN+1.
2. Acknowledgment Number - номер подтверждения: 32 бита Если бит ACK установлен, это поле содержит значение следующего порядкового номера, который отправитель сегмента ожидает получить. После организации соединения это значение передается всегда.

Сервер получает сегмент, запоминает номер последовательности и посылает клиенту сегмент с номером последовательности и флагами SYN и ACK. Если клиент получает сегмент с флагом SYN,

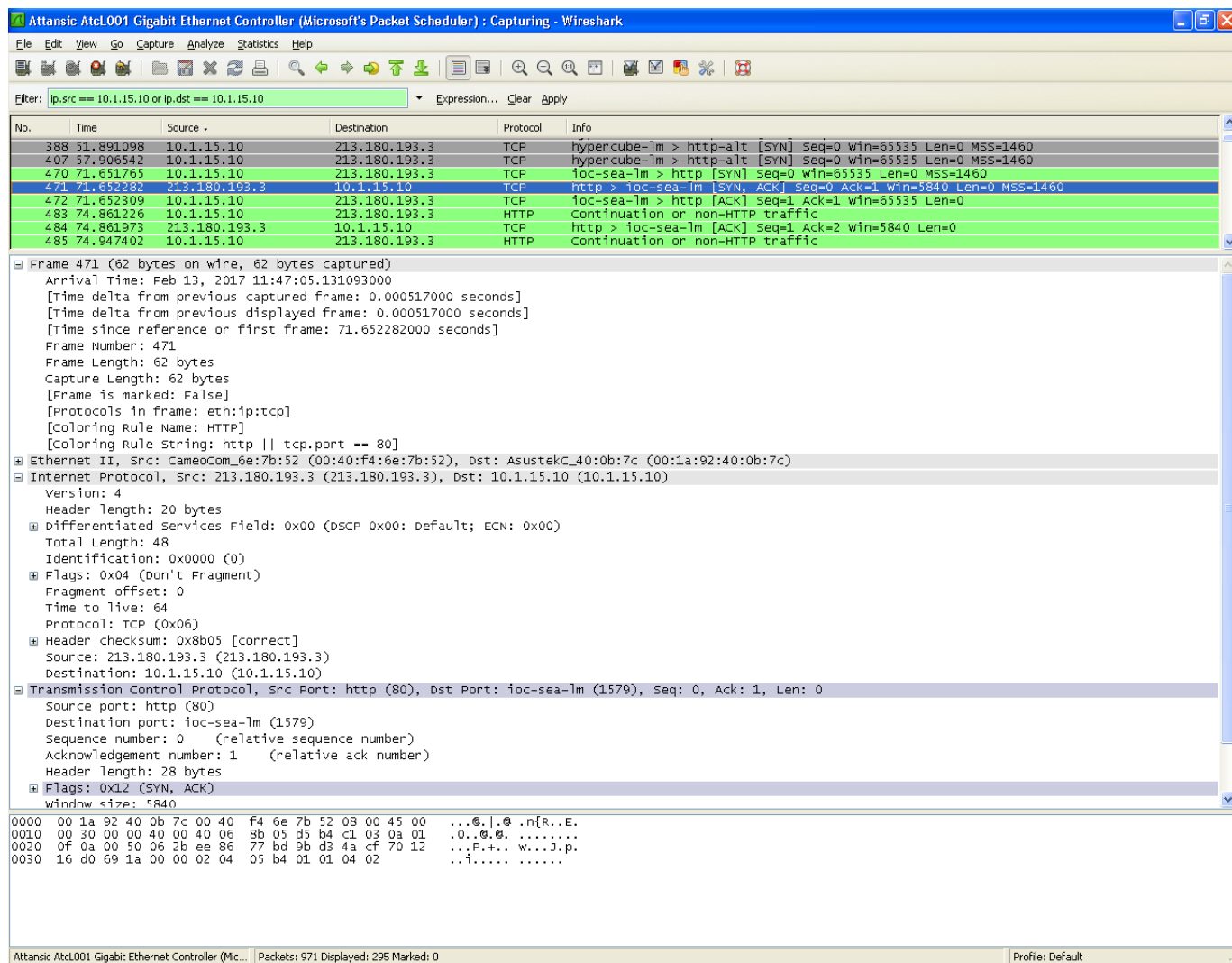


Рис. 18: Ответ сервера на установление TCP-соединения

то он запоминает номер последовательности и посылает сегмент с флагом ACK.

4.5.2 Разрыв соединения

При разрыве соединения сервер отправляет клиенту пакет с установленным флагом RST.

File

Edit

View

Go

Capture

Analyze

Statistics

Telephony

Tools

Internals

Help

Рис. 19: Подтверждение от клиента о получении ответа

FileEditViewGoCaptureAnalyzeStatisticsTelephonyToolsInternalsHelp

Filtertcp and not http and (tcp.flags.reset == 1)

Expression...ClearApplySave

No.

Time

Source

Destination

Protocol

Length

Info

39610.8598000191.238.224.150192.168.1.124TCP54220422350 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0

6190563.934595000192.168.1.12423.77.198.8TCP5422424-443 [RST, ACK] Seq=913 Ack=5913 Win=0 Len=0

11280948.135387000173.194.33.191192.168.1.124TCP54443-22462 [RST] Seq=2 Win=0 Len=0

11597987.99145000087.245.198.35192.168.1.124TCP5480-22457 [RST] Seq=798 Win=0 Len=0

131581280.495933000192.168.1.1192.168.1.124TCP541910-22482 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0

131621281.000311000192.168.1.1192.168.1.124TCP541910-22482 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0

131791281.548938000192.168.1.1192.168.1.124TCP541910-22482 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0

131961281.587947000192.168.1.1192.168.1.124TCP541910-22484 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0

132631288.102745000192.168.1.1192.168.1.124TCP541910-22484 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0

Frame 39: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface 0

Ethernet II, Src: Tp-LinkT_f2:ff:00 (54:e6:fc:f2:ff:00), Dst: Azurewav_b6:2a:39 (24:0a:64:b6:2a:39)

Internet Protocol Version 4, Src: 191.238.224.150 (191.238.224.150), Dst: 192.168.1.124 (192.168.1.124)

Version: 4

Header Length: 20 bytes

Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-capable Transport))

Total Length: 40

Identification: 0x7135 (28981)

Flags: 0x02 (Don't Fragment)

Fragment offset: 0

Time to live: 108

Protocol: TCP (6)

Header checksum: 0x3af1 [validation disabled]

Source: 191.238.224.150 (191.238.224.150)

Destination: 192.168.1.124 (192.168.1.124)

[Source GeoIP: unknown]

[Destination GeoIP: unknown]

Transmission Control Protocol, Src Port: 80 (80), Dst Port: 22350 (22350), Seq: 1, Ack: 1, Len: 0

Source Port: 80 (80)

Destination Port: 22350 (22350)

[Stream index: 2]

[TCP Segment Len: 0]

Sequence number: 1 (relative sequence number)

Acknowledgment number: 1 (relative ack number)

Header Length: 20 bytes

... 0000 0001 0100 = Flags: 0x014 (RST, ACK)

000. = Reserved: Not set

...0 = Nonce: Not set

.... 0... = Congestion window Reduced (CWR): Not set

.... .0... = ECN-Echo: Not set

.... .0. = Urgent: Not set

.... .1. = Acknowledgment: Set

.... .1. = Push: Not set

.... .1. = Reset: Set

[Expert Info (warn/Sequence): Connection reset (RST)]

.... .0. = Syn: Not set

.... .0. = Fin: Not set

Window size value: 0

[Calculated window size: 0]

[Window size scaling factor: -1 (unknown)]

000024 0a 64 b6 2a 39 54 e6 fc f2 ff 00 08 00 45 00\$.d.*9T.....E.

001000 28 71 35 40 00 8c 06 3a f1 bf ee e0 96 c0 a8.q58.1.....E.

002001 7c 00 50 57 4e e8 b3 80 8e 4f 0d 57 5b 50 14}.PWN...O.W[P.

003000 00 e5 dd 00 00.....

Беспроводная сеть: <live capture in progress...> Packets: 20148 - Displayed: 9 (0,0%)

Рис. 20: Пример пакета с флагом RST

4.5.3 Завершение соединения

При завершении соединения происходит обмен пакетами с флагами FIN и ACK. Сервер посылает

```

+ Frame 13238: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface 0
+ Ethernet II, Src: D-LinkIn_7c:58:30 (c8:be:19:7c:58:30), Dst: IntelCor_9d:6b:3d (4c:eb:42:9d:6b:3d)
+ Internet Protocol Version 4, Src: 185.26.97.188 (185.26.97.188), Dst: 192.168.1.3 (192.168.1.3)
- Transmission Control Protocol, Src Port: 80 (80), Dst Port: 51254 (51254), Seq: 1199, Ack: 1695, Len: 0
  Source Port: 80 (80)
  Destination Port: 51254 (51254)
  [Stream index: 258]
  [TCP Segment Len: 0]
  Sequence number: 1199 (relative sequence number)
  Acknowledgment number: 1695 (relative ack number)
  Header Length: 20 bytes
- .... 0000 0001 0001 = Flags: 0x011 (FIN, ACK)
  000. .... = Reserved: Not set
  ...0 .... = Nonce: Not set
  .... 0... = Congestion window Reduced (CWR): Not set
  .... .0.. = ECN-Echo: Not set
  .... ..0. = Urgent: Not set
  .... ...1 = Acknowledgment: Set
  .... .... 0... = Push: Not set
  .... ..... 0.. = Reset: Not set
  .... .... ..0. = Syn: Not set
+ .... .... ....1 = Fin: Set
  window size value: 1125
  [Calculated window size: 18000]
  [window size scaling factor: 16]
+ Checksum: 0xf23b [validation disabled]
  urgent pointer: 0
```

Рис. 21: Передача пакета с флагами FIN и ACK от сервера клиенту

клиенту пакет с установленными флагами ACK, FIN. Сервер переходит из состояния ESTABLISHED в состояние FIN-WAIT-1.

```

+ Frame 13239: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface 0
+ Ethernet II, Src: IntelCor_9d:6b:3d (4c:eb:42:9d:6b:3d), Dst: D-LinkIn_7c:58:30 (c8:be:19:7c:58:30)
+ Internet Protocol Version 4, Src: 192.168.1.3 (192.168.1.3), Dst: 185.26.97.188 (185.26.97.188)
- Transmission Control Protocol, Src Port: 51254 (51254), Dst Port: 80 (80), Seq: 1695, Ack: 1200, Len: 0
  Source Port: 51254 (51254)
  Destination Port: 80 (80)
  [Stream index: 258]
  [TCP Segment Len: 0]
  Sequence number: 1695 (relative sequence number)
  Acknowledgment number: 1200 (relative ack number)
  Header Length: 20 bytes
- .... 0000 0001 0000 = Flags: 0x010 (ACK)
  000. .... = Reserved: Not set
  ...0 .... = Nonce: Not set
  .... 0... = Congestion window Reduced (CWR): Not set
  .... .0.. = ECN-Echo: Not set
  .... ..0. = Urgent: Not set
  .... ...1 = Acknowledgment: Set
  .... .... 0... = Push: Not set
  .... ..... 0.. = Reset: Not set
  .... .... ..0. = Syn: Not set
  .... .... ...0 = Fin: Not set
  window size value: 7892
  [Calculated window size: 31568]
  [window size scaling factor: 4]
+ Checksum: 0xd7cc [validation disabled]
  urgent pointer: 0
+ [SEQ/ACK analysis]
```

Рис. 22: Подтверждение получения пакета

⊞	Frame 13240: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface 0
⊞	Ethernet II, Src: IntelCor_9d:6b:3d (4c:eb:42:9d:6b:3d), Dst: D-LinkIn_7c:58:30 (c8:be:19:7c:58:30)
⊞	Internet Protocol Version 4, Src: 192.168.1.3 (192.168.1.3), Dst: 185.26.97.188 (185.26.97.188)
⊞	Transmission Control Protocol, Src Port: 51254 (51254), Dst Port: 80 (80), Seq: 1695, Ack: 1200, Len: 0
	Source Port: 51254 (51254)
	Destination Port: 80 (80)
	[Stream index: 258]
	[TCP Segment Len: 0]
	Sequence number: 1695 (relative sequence number)
	Acknowledgment number: 1200 (relative ack number)
	Header Length: 20 bytes
⊞	... 0000 0001 0001 = Flags: 0x011 (FIN, ACK)
	000. = Reserved: Not set
	...0 = Nonce: Not set
 0... = Congestion Window Reduced (CWR): Not set
0.. = ECN-Echo: Not set
0. = Urgent: Not set
1 = Acknowledgment: Set
 0.. = Push: Not set
 0.. = Reset: Not set
0. = Syn: Not set
⊞1 = Fin: Set
	window size value: 7892
	[calculated window size: 31568]
	[window size scaling factor: 4]
⊞	checksum: 0xd7cb [validation disabled]
	urgent pointer: 0

Рис. 23: Подтверждение завершения соединения от клиента серверу

Клиент посылает второй пакет с флагами FIN, ACK, после его отсылки клиент переходит в состояние LAST-ACK, а сервер в состояние TIME-WAIT.

4.5.4 Установка соединения с отсутствующим портом

При попытке подключения к отсутствующему порту, не приходит ACK и RST, поэтому клиент находится в подвешенном состоянии и ожидает ответа.

No.	Time	Source	Destination	Protocol	Info
22	4.975089	10.1.15.12	46.255.138.1	TCP	ff-sm > 65533 [SYN] Seq=0 Win=65535 Len=0 MSS=1460
36	7.257085	10.1.15.12	46.255.138.1	TCP	ff-sm > 65533 [SYN] Seq=0 Win=65535 Len=0 MSS=1460
64	13.273460	10.1.15.12	46.255.138.1	TCP	ff-sm > 65533 [SYN] Seq=0 Win=65535 Len=0 MSS=1460

⊞	Frame 22 (62 bytes on wire, 62 bytes captured)
⊞	Ethernet II, Src: 02:ff:46:ce:2a:d0 (02:ff:46:ce:2a:d0), Dst: CamoCom_6e:7b:52 (00:40:f4:6e:7b:52)
⊞	Internet Protocol, Src: 10.1.15.12 (10.1.15.12), Dst: 46.255.138.1 (46.255.138.1)
⊞	Transmission Control Protocol, Src Port: ff-sm (1091), Dst Port: 65533 (65533), Seq: 0, Len: 0
	Source port: ff-sm (1091)
	Destination port: 65533 (65533)
	Sequence number: 0 (relative sequence number)
	Header length: 28 bytes
⊞	Flags: 0x02 (SYN)
	0... = Congestion Window Reduced (CWR): Not set
	..0. = ECN-Echo: Not set
	...0. = Urgent: Not set
 0... = Acknowledgment: Not set
 0... = Push: Not set
0.. = Reset: Not set
1. = Syn: Set
0 = Fin: Not set
	Window size: 65535
⊞	Checksum: 0xa507 [correct]
⊞	Options: (8 bytes)

0000	00 40 f4 6e 7b 52 02 ff	46 ce 2a d0 08 00 45 00	.0.nR..F..E..
0010	00 30 09 78 40 00 80 06	1f 43 0a 01 0f 0c 2e ff	.0.xb...C.....
0020	8a 01 04 43 ff fd 1f ff	e7 c8 00 00 00 00 70 02	...C.....p..
0030	ff ff a5 07 00 00 02 04	05 b4 01 01 04 02

Рис. 24: Попытка tcp - соединения на sakh.com:65533

5 Выводы

В ходе работы были получены навыки работы в программе Wireshark и закрепились знания о сетевых протоколах ARP, ICMP, TCP. Были рассмотрены:

1. работу утилит ping и tracer;
2. работа ARP-протокола;
3. работа протокола ICMP, включая такие типовые случаи, как: отправка фрагментированного пакета, возникновение ошибки 3.1, трассировка маршрута;
4. установка, разрыв и завершение TCP соединения;