

Национальный исследовательский университет ИТМО  
Факультет программной инженерии и компьютерной техники

**Учебно-исследовательская работа №3**  
**по дисциплине Сети ЭВМ и телекоммуникации**  
**Анализ трафика компьютерных сетей утилитой Wireshark**

Студент: Саржевский Иван  
Группа: Р3302

г. Санкт-Петербург  
2020 г.

# Содержание

<b>1</b>	<b>Цель</b>	<b>2</b>
<b>2</b>	<b>Анализ трафика утилиты ping</b>	<b>2</b>
2.1	Фрейм . . . . .	2
2.2	Ethernet II . . . . .	2
2.3	IPv4 . . . . .	3
2.4	Internet Control Message Protocol (ICMP) . . . . .	4
2.5	Ответы на вопросы . . . . .	5

# 1 Цель

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

## 2 Анализ трафика утилиты ping

Для анализа трафика, создаваемого утилитой ping был выбран сайт **www.ias.ru**.

```
Frame 5: 1042 bytes on wire (8336 bits), 1042 bytes captured (8336 bits) on interface wlp4s0, id 0
Ethernet II, Src: Chongqin_64:e6:c5 (c0:b5:d7:64:e6:c5), Dst: Tp-LinkT_3d:06:ae (cc:32:e5:3d:06:ae)
Internet Protocol Version 4, Src: 192.168.0.105, Dst: 81.195.71.197
Internet Control Message Protocol
```

Рис. 1: Заголовки протоколов для команды ping.

На рисунке 1 изображены заголовки различных протоколов, используемых при передаче запроса.

### 2.1 Фрейм

```
Interface id: 0 (wlp4s0)
Encapsulation type: Ethernet (1)
Arrival Time: Apr 12, 2020 22:23:51.094101026 MSK
[Time shift for this packet: 0.000000000 seconds]
Epoch Time: 1586719431.094101026 seconds
[Time delta from previous captured frame: 0.000253948 seconds]
[Time delta from previous displayed frame: 0.000000000 seconds]
[Time since reference or first frame: 0.036999160 seconds]
Frame Number: 5
Frame Length: 1042 bytes (8336 bits)
Capture Length: 1042 bytes (8336 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]
```

Рис. 2: Информация о фрейме команды ping.

Структура, представленная на рисунке 2, описывает метаданные Wireshark для этого запроса - его порядковый номер среди всех записанных, время прибытия, размер, протокол и цвет выделения в интерфейсе.

### 2.2 Ethernet II

Ethernet II - протокол канального уровня, т.е. описывает передачу данных в рамках локальной сети. Типичная структура кадра Ethernet II представлена в таблице 1.

Таблица 1: Структура кадра Ethernet II.

Кадр Ethernet II (от 64-х до 1528-ти байт)				
MAC-заголовок (14 байт)			Данные (от 46-ти до 1500 байт)	–
MAC получателя (6 байт)	MAC отправителя (6 байт)	Тип протокола (2 байта)	Данные	CRC (4 байта)

В данном случае получателем выступает роутер, а отправителем - рабочая машина, их MAC-адреса записаны в кадр, тип протокола - IPv4, что можно увидеть на рисунке 3.

```

Destination: Tp-LinkT_3d:06:ae (cc:32:e5:3d:06:ae)
Address: Tp-LinkT_3d:06:ae (cc:32:e5:3d:06:ae)
....0. .... = LG bit: Globally unique address (factory default)
....0. .... = IG bit: Individual address (unicast)
Source: Chongqin_64:e6:c5 (c0:b5:d7:64:e6:c5)
Address: Chongqin_64:e6:c5 (c0:b5:d7:64:e6:c5)
....0. .... = LG bit: Globally unique address (factory default)
....0. .... = IG bit: Individual address (unicast)
Type: IPv4 (0x0800)

```

Рис. 3: Кадр Ethernet II для ping.

## 2.3 IPv4

IPv4 - протокол сетевого уровня. Подробные сведения полей, которые включены в заголовок протокола, приведены на рисунке 4. Туда включены IP-адреса отправителя и получателя, длина заголовка и сообщения, флаги указывающие на наличие фрагментации данных, промежуточности данного пакета и т. д.

Offsets	Octet	0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Version				IHL				DSCP						ECN		Total Length															
4	32	Identification																Flags			Fragment Offset												
8	64	Time To Live								Protocol								Header Checksum															
12	96	Source IP Address																															
16	128	Destination IP Address																															
20	160	Options (if IHL > 5)																															
24	192																																
28	224																																
32	256																																

Рис. 4: Структура заголовка IPv4.

Данные, переданные с использованием протокола IPv4 для команды ping можно увидеть на рисунке 5.

```

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: 1028
Identification: 0x8a07 (35335)
Flags: 0x4000, Don't fragment
  0... .... = Reserved bit: Not set
  .1... .... = Don't fragment: Set
  ..0. .... = More fragments: Not set
...0 0000 0000 0000 = Fragment offset: 0
Time to live: 64
Protocol: ICMP (1)
Header checksum: 0x5258 [validation disabled]
[Header checksum status: Unverified]
Source: 192.168.0.105
Destination: 81.195.71.197
[Destination GeoIP: RU]

```

Рис. 5: Данные пакета IPv4 для команды ping.

## 2.4 Internet Control Message Protocol (ICMP)

Offsets	Octet	0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Type								Code								Checksum															
4	32	Rest of Header																															

Рис. 6: Структура заголовка ICMP.

Данный протокол сетевого уровня используется для передачи служебных сообщений - кода ошибки в случае исключительной ситуации, кода запрашиваемой операции и кода подтверждения в случае удачной передачи. Подробная структура заголовка ICMP приведена на рисунке 6.

```

Type: 8 (Echo (ping) request)
Code: 0
Checksum: 0xb113 [correct]
[Checksum Status: Good]
Identifier (BE): 4975 (0x136f)
Identifier (LE): 28435 (0x6f13)
Sequence number (BE): 1 (0x0001)
Sequence number (LE): 256 (0x0100)
[Response frame: 6]
Timestamp from icmp data: Apr 12, 2020 22:23:51.000000000 MSK
[Timestamp from icmp data (relative): 0.094101026 seconds]
Data (992 bytes)

```

Рис. 7: Данные ICMP для команды ping.

Для команды ping структура ICMP представлена на рисунке 7.

Структура ответов имеет схожую структуру, отличаться они будут типом ICMP, сменой адресов получателя и отправителя, timestamp'ами.

## 2.5 Ответы на вопросы

1. Имеет ли место фрагментация исходного пакета, какое поле на это указывает?  
– Да. Но только в том случае, если размер пакета превышает **Maximum Transmission Unit (MTU)**, равный для протокола **Ethernet II** 1500 байт. Информация о наличии фрагментации содержится во флаге в заголовке **IPv4**.
2. Какая информация указывает, является ли фрагмент пакета последним или промежуточным?  
– Флаг **More Fragments** в заголовке **IPv4**.
3. Чему равно количество фрагментов при передаче ping-пакетов?  
– MTU равен 1500 байт, пакет включает в себя **IPv4**-заголовок (20 байт), **ICMP**-заголовок (8 байт), и, непосредственно, данные. Это означает, что количество фрагментов равно  $\lceil (s + 20 + 8) / 1500 \rceil$ , где  $s$  - аргумент **-s** команды **ping**. Зависимость количества фрагментов от размера пакета приведена в таблице 2.

Таблица 2: Количество фрагментов при разных размерах пакета

Размер пакета	100	500	1000	1500	2000	3000	5000	10000
Кол-во фраг.	1	1	1	2	2	3	4	7

4. Построить график, в котором на оси абсцисс находится размер\_пакета, а по оси ординат – количество фрагментов, на которое был разделён каждый **ping**-пакет.  
– см. рисунок 8.

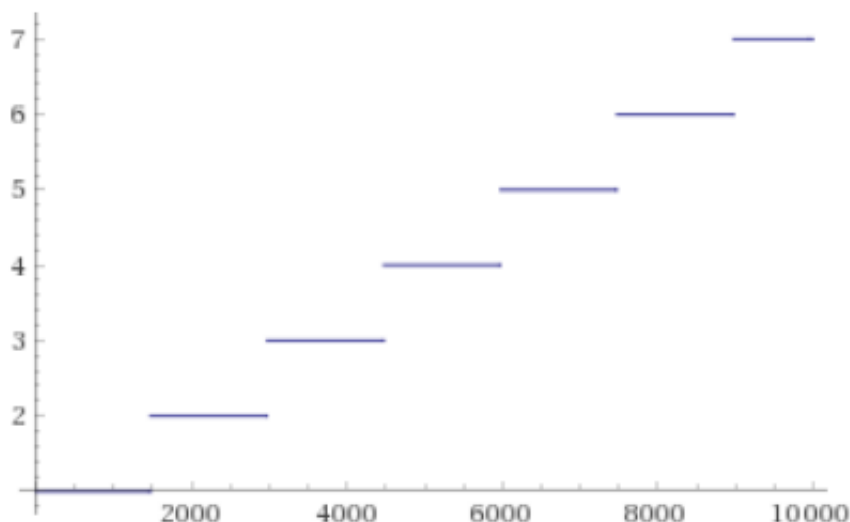


Рис. 8: Зависимость количества фрагментов от размера пакета

5. Как изменить поле TTL с помощью утилиты **ping**?

- Linux : **ping -t ttl\_value**

- Windows : `ping -i ttl_value`

6. Что содержится в поле данных `ping`-пакета?

– В поле данных содержится текущий timestamp, а затем циклически повторяющиеся биты от 00 до FF.

```
0030 00 00 5d 12 04 00 00 00 00 00 10 11 12 13 14 15
0040 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25
0050 26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35
0060 36 37 38 39 3a 3b 3c 3d 3e 3f 40 41 42 43 44 45
. . .
0110 e6 e7 e8 e9 ea eb ec ed ee ef f0 f1 f2 f3 f4 f5
0120 f6 f7 f8 f9 fa fb fc fd fe FF 00 01 02 03 04 05
```