

Университет ИТМО

**Сети ЭВМ и телекоммуникации**  
**Лабораторная работа №3**

Выполнила: Калугина Марина  
Группа: Р3302

г. Санкт-Петербург

2020 г.

# Содержание

<b>Содержание</b>	<b>2</b>
<b>Цель работы</b>	<b>6</b>
<b>Ход работы</b>	<b>6</b>
Утилита ping	6
Задание	6
Описание структуры	6
Frame 19	6
Ethernet II	7
Internet Protocol Version 4 (IPv4):	7
Internet Control Message Protocol	8
Анализ трафика утилиты ping	9
Ответы на вопросы	10
Имеет ли место фрагментация исходного пакета, какое поле на это указывает?	10
Какая информация указывает, является ли фрагмент пакета последним или промежуточным?	10
Чему равно количество фрагментов при передаче ping-пакетов?	11
Построить график, в котором на оси абсцисс находится размер_пакета, а по оси ординат – количество фрагментов, на которое был разделён каждый ping-пакет	11
Как изменить поле TTL с помощью утилиты ping?	11
Что содержится в поле данных ping-пакета?	11
Анализ трафика утилиты tracert (traceroute)	12
Задание	12
Описание структуры	12
UDP-запрос	12
Frame 33	12
Ethernet II	13
Internet Protocol Version 4 (IPv4)	13
User Datagram Protocol (UDP)	14
Data	15
ICMP-ответ	15
Frame 37	15
Ethernet II	16
Internet Protocol Version 4 (IPv4)	16
Internet Control Message Protocol	17
Ответы на вопросы	17
Сколько байт содержится в заголовке IP? Сколько байт содержится в поле данных?	17

Как и почему изменяется поле TTL в следующих друг за другом ICMP-пакетах tracer? Для ответа на этот вопрос нужно проследить изменение TTL при передаче по маршруту, состоящему из более чем двух хопов.	18
Чем отличаются ICMP-пакеты, генерируемые утилитой tracer, от ICMP-пакетов, генерируемых утилитой ping (см. предыдущее задание).	18
Чем отличаются полученные пакеты «ICMP reply» от «ICMP error» и зачем нужны оба этих типа ответов?	18
Что изменится в работе tracer, если убрать ключ «-d»? Какой дополнительный трафик при этом будет генерироваться?	19
Анализ HTTP-трафика	19
Задание	19
Transmission Control Protocol (TCP)	20
Hypertext Transfer Protocol (HTTP)	21
Первичный запрос	22
Первичный ответ	22
Повторный запрос	23
Повторный ответ	24
Анализ DNS-трафика	24
Задание	24
Структура DNS-пакета	25
Ответы на вопросы	26
Почему адрес, на который отправлен DNS-запрос, не совпадает с адресом посещаемого сайта?	26
Какие бывают типы DNS-запросов?	27
В какой ситуации нужно выполнять независимые DNS-запросы для получения содержащихся на сайте изображений?	27
Анализ ARP-трафика	27
Задание	27
Описание	27
Address Resolution Protocol (ARP)	28
Структура запроса	28
Структура ответа	28
Ответы на вопросы	29
Какие MAC-адреса присутствуют в захваченных пакетах ARP-протокола? Что означают эти адреса? Какие устройства они идентифицируют?	29
Какие MAC-адреса присутствуют в захваченных HTTP-пакетах и что означают эти адреса? Что означают эти адреса? Какие устройства они идентифицируют?	29
Для чего ARP-запрос содержит IP-адрес источника?	30
Анализ трафика утилиты nslookup	30
Задание	30
Структура	30
DNS-запрос	30

DNS-ответ	31
Запросы и ответы типа NS	32
Ответы на вопросы	32
Чем различается трасса трафика в п.2 и п.4, указанных выше?	32
Что содержится в поле «Answers» DNS-ответа?	33
Каковы имена серверов, возвращающих авторитативный (authoritative) отклик?	33
Анализ FTP-трафика	33
Задание	33
Структура	33
FTP	33
FTP-DATA	35
Ответы на вопросы	36
Сколько байт данных содержится в пакете FTP-DATA?	36
Как выбирается порт транспортного уровня, который используется для передачи FTP-пакетов?	36
Чем отличаются пакеты FTP от FTP-DATA?	36
Анализ DHCP-трафика	36
Задание	36
Структура	36
Realise. Сброс текущего ip	37
Discover. Обнаружение DHCP	38
Offert. Предложение DHCP	38
Request. Запрос DHCP	40
ACK. Подтверждение DHCP	41
Временная диаграмма	42
Используемые порты	42
Ответы на вопросы	43
Чем различаются пакеты «DHCP Discover» и «DHCP Request»?	43
Как и почему менялись MAC- и IP-адреса источника и назначения в переданных DHCP-пакетах.	43
Каков IP-адрес DHCP-сервера?	43
Что произойдет, если очистить использованный фильтр “bootp”?	44
Анализ Skype-трафика	44
Задание	44
Структура документа	44
Текст	44
Secure Sockets Layer (SSL)	44
Аудио	44
Видео	45
Ответы на вопросы	46
Чем различаются пакета разных видов Skype-трафика (текст, аудио, видео)?	46

Какой Wireshark-фильтр следует использовать для независимой идентификации Skype-трафика разных видов (текст, аудио, видео)?

47

## Цель работы

Изучить структуру протокольных блоков данных, анализируя реальный трафик на компьютере студента с помощью бесплатно распространяемой утилиты Wireshark.

## Ход работы

### Утилита ping

#### Задание

Необходимо отследить и проанализировать трафик, создаваемый утилитой ping.

Для выполнения этого задания был выбран сайт kmm.org

Утилита ping запускалась со следующими значениями размеров пакетов: 100 500 1000 1500 2000 3000 4000 5000 7000 10000

#### Описание структуры

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

```
▶ Frame 19: 142 bytes on wire (1136 bits), 142 bytes captured (1136 bits) on interface 0
▶ Ethernet II, Src: LcfcHefe_2d:0a:c7 (54:e1:ad:2d:0a:c7), Dst: Tp-LinkT_1c:a4:84 (98:da:c4:1c:a4:84)
▶ Internet Protocol Version 4, Src: 192.168.0.105, Dst: 184.175.101.76
▶ Internet Control Message Protocol
```

Рис. 1

Frame 19

Информация о фрейме. На рисунке 2 представлена более подробная информация.

```

▶ Interface id: 0 (enp2s0)
Encapsulation type: Ethernet (1)
Arrival Time: Apr 1, 2020 21:13:12.277522754 MSK
[Time shift for this packet: 0.000000000 seconds]
Epoch Time: 1585764792.277522754 seconds
[Time delta from previous captured frame: 0.856569553 seconds]
[Time delta from previous displayed frame: 0.856569553 seconds]
[Time since reference or first frame: 3.003537951 seconds]
Frame Number: 19
Frame Length: 142 bytes (1136 bits)
Capture Length: 142 bytes (1136 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

## Ethernet II

Протокол канального уровня. Предназначен для передачи данных между физическими устройствами. Содержит адреса источника и назначения (mac-адреса устройств).

На рисунке 3 показано как выглядит формат кадра для Ethernet II

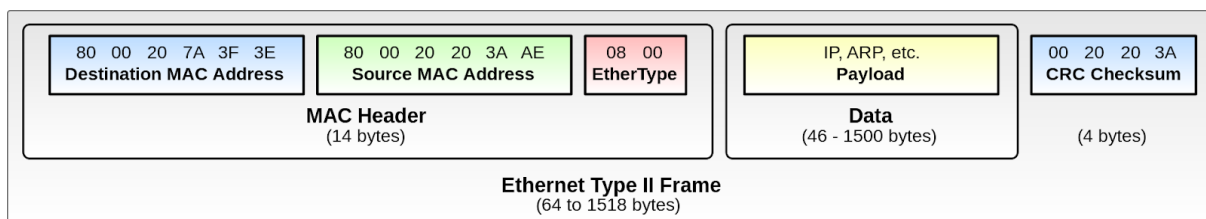


Рис.3

На рисунке 4 показана информация содержащаяся в этом протоколе

```

▼ Destination: Tp-LinkT_1c:a4:84 (98:da:c4:1c:a4:84)
  Address: Tp-LinkT_1c:a4:84 (98:da:c4:1c:a4:84)
  ....0. .... = LG bit: Globally unique address (factory default)
  ....0 .... = IG bit: Individual address (unicast)
▼ Source: LcfcHefe_2d:0a:c7 (54:e1:ad:2d:0a:c7)
  Address: LcfcHefe_2d:0a:c7 (54:e1:ad:2d:0a:c7)
  ....0. .... = LG bit: Globally unique address (factory default)
  ....0 .... = IG bit: Individual address (unicast)
Type: IPv4 (0x0800)

```

Рис.4

## Internet Protocol Version 4 (IPv4):

Протокол сетевого уровня.

В первом байте храниться информация о версии ip-протокола и о размере хедера. Далее идет информация о флагах: информация о фрагментации данных, информация о том последний или промежуточный фрагмент рассматривается и др. Также здесь храниться информация об ip-адреса источника и приемника.

На рисунке 5 показана структура IPv4 протокола.

IPv4 Header Format																																	
Отступ	Октет	0								1								2								3							
Октет	Бит	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
0	0	Версия				Размер заголовка				Differentiated Services Code Point				Explicit Congestion Notification				Размер пакета (полный)															
4	32	Идентификатор																Флаги		Смещение фрагмента													
8	64	Время жизни								Протокол								Контрольная сумма заголовка															
12	96	IP-адрес источника																															
16	128	IP-адрес назначения																															
20	160	Опции (если размер заголовка > 5)																															
20 или 24+	160 или 192+	Данные																															

Рис.5

На рисунке 6 показаны те данные протокола IPv4, которые были получены.

```

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: 128
Identification: 0x8b18 (35608)
▼ 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: 0xd057 [validation disabled]
[Header checksum status: Unverified]
Source: 192.168.0.105
Destination: 184.175.101.76

```

Рис. 6

## Internet Control Message Protocol

Сетевой протокол. В основном используется для передачи сообщений об ошибках и других исключительных ситуациях, возникших при передаче данных, например, запрашиваемая услуга недоступна, или хост, или маршрутизатор не отвечают. Здесь хранится информация о контрольной сумме, порядковом номере, даваемом утилитой ping, информации о времени и сами данные.

На рисунке 7 показана структура ICMP-пакета



IP Datagram				
	Bit 0 — 7	Bit 8 — 15	Bit 16 — 23	Bit 24 — 31
IP Header (20 bytes)	Version/IHL	Type of service	Length	
	Identification		flags and offset	
	Time To Live (TTL)	Protocol	Checksum	
	Source IP address			
	Destination IP address			
ICMP Header (8 bytes)	Type of message	Code	Checksum	
	Header Data			
ICMP Payload (optional)	Payload Data			

Рис.7

На рисунке 8 представлена подробная информация, которая была передана этим протоколом.

Type: 8 (Echo (ping) request)
Code: 0
Checksum: 0x3c34 [correct]
[Checksum Status: Good]
Identifier (BE): 14527 (0x38bf)
Identifier (LE): 48952 (0xbf38)
Sequence number (BE): 11 (0x000b)
Sequence number (LE): 2816 (0x0b00)
<a href="#">[Response frame: 26]</a>
Timestamp from icmp data: Apr 1, 2020 21:13:12.000000000 MSK
[Timestamp from icmp data (relative): 0.277522754 seconds]
► Data (92 bytes)

Рис.8

Передача **ответа** имеет схожую структуру. Отличие будет происходить в данных, установленных флагах и т.п.

## Анализ трафика утилиты ping

В таблице 1 представлены результаты анализа трафика утилиты ping

Таблица 1

Размер пакета	100	500	1000	1500	2000	3000	4000	5000	7000	10000
Фрагментация	Нет	Нет	Нет	Да	Да	Да	Да	Да	Да	Да
Кол-во фрагментов	1	1	1	2	2	3	3	4	5	7

## Ответы на вопросы

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

Да, фрагментация исходного пакета имеет место, при размере пакета большем максимального размера (MTU). Для протокола Ethernet II MTU=1500 байт.

На рисунке 9 показан формат кадра Ethernet II

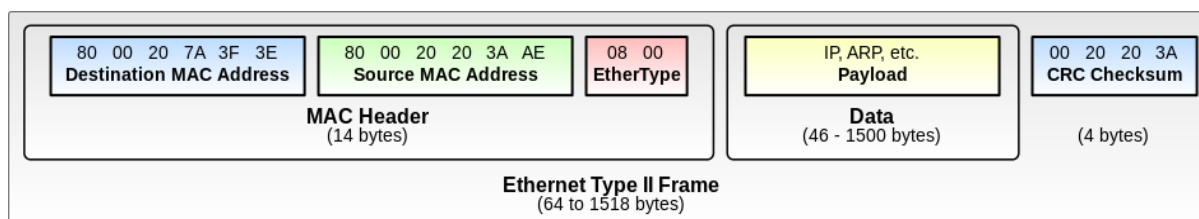


Рис.9.

Протокол IP содержит флаг, указывающий на фрагментацию пакета в третьем бите (см. рис. 10)

```

▼ 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
  
```

Рис.10.

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

Бит More fragments поля Flags заголовка IPv4 указывает, является этот фрагмент последним (значение бита 0) или промежуточным (значение 1). (см. рис. 10)

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

Максимальный размер фрагмента в нашем случае равен 1500 байт, куда входит IPv4 заголовок (20 байт). Остальной объем занимает ICMP пакет. Который включает в себя заголовок (8 байт (Header + Header data) и payload. Таким образом можно рассчитать количество фрагментов ping пакета:  $N = (\text{payload} + 8) / 1480$ . Где payload — значение, указываемое аргументом в утилите ping.

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

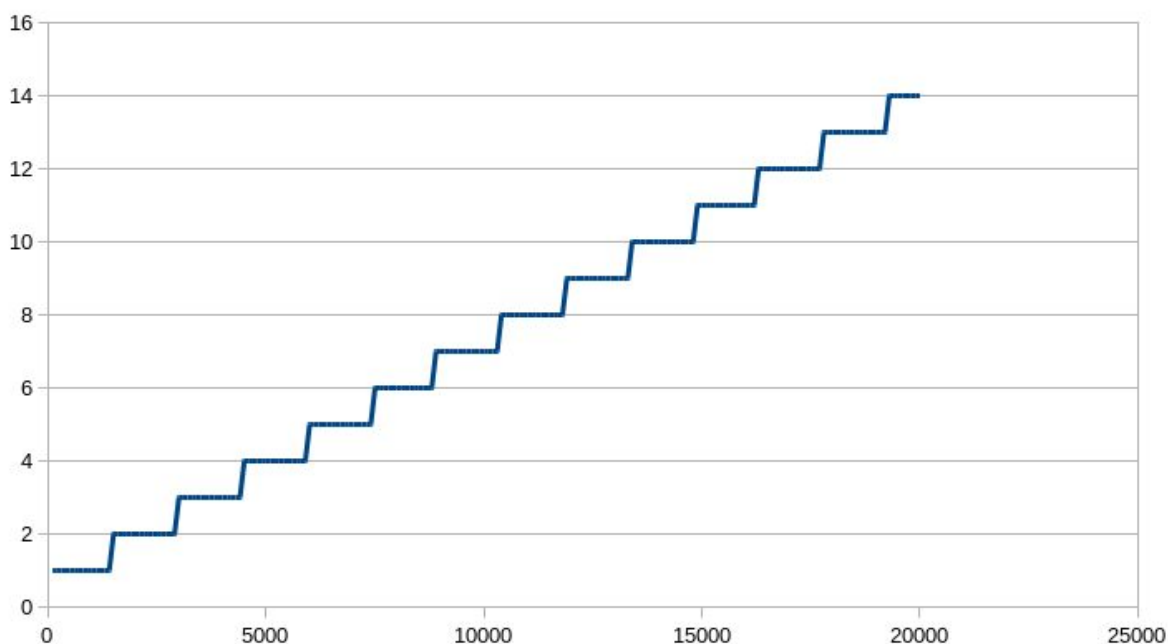


Рис. 11.

На рисунке 11 показано как число фрагментов зависит от размера пакета. Здесь каждые 1480 байт число фрагментов увеличивается на 1, исходя из описания ответа на вопрос 3.

5. Как изменить поле TTL с помощью утилиты ping?

Для Linux — `ping -t new_ttl host`

Для Windows — `ping -i new_ttl host`

6. Что содержится в поле данных ping-пакета?

ICMP содержит Header Data и Payload Data. В Header Data, при ping запросах, содержится идентификатор запроса и порядковый номер. В Payload Data ping пакета

содержится время отправки пакета и циклическое повторение последовательности байтов 00...ff. На рисунке 12 показано содержимое поля данных ring пакета.

98	da	c4	1c	a4	84	54	e1	ad	2d	0a	c7	08	00	45	00
00	80	8b	18	40	00	40	01	d0	57	c0	a8	00	69	b8	af
65	4c	08	00	3c	34	38	bf	00	0b	b8	d9	84	5e	00	00
00	00	de	3b	04	00	00	00	00	00	10	11	12	13	14	15
16	17	18	19	1a	1b	1c	1d	1e	1f	20	21	22	23	24	25
26	27	28	29	2a	2b	2c	2d	2e	2f	30	31	32	33	34	35
36	37	38	39	3a	3b	3c	3d	3e	3f	40	41	42	43	44	45
46	47	48	49	4a	4b	4c	4d	4e	4f	50	51	52	53	54	55
56	57	58	59	5a	5b	5c	5d	5e	5f	60	61	62	63		

Рис. 12

# Анализ трафика утилиты tracert (traceroute)

## Задание

Необходимо отследить и проанализировать трафик, создаваемый утилитой tracert (или traceroute в Linux). Для выполнения лабораторной работы был использован сайт kmm.org

## Описание структуры

Утилита traceroute отправляет udp-запросы, увеличивая с каждым разом ttl (пачкой из 3-х штук). И получает ICMP-ответ.

## UDP-запрос

На рисунке 13 представлено, как выглядит запрос.

```
▶ Frame 33: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0
▶ Ethernet II, Src: LcfcHefe_2d:0a:c7 (54:e1:ad:2d:0a:c7), Dst: Tp-LinkT_1c:a4:84 (98:da:c4:1c:a4:84)
▶ Internet Protocol Version 4, Src: 192.168.0.105, Dst: 184.175.101.76
▶ User Datagram Protocol, Src Port: 47081, Dst Port: 33449
▶ Data (32 bytes)
```

Рис. 13

Frame 33

Информация о пакете. (см. рис 14)

```
▶ Interface id: 0 (enp2s0)
  Encapsulation type: Ethernet (1)
  Arrival Time: Apr  1, 2020 23:39:00.395898628 MSK
  [Time shift for this packet: 0.000000000 seconds]
  Epoch Time: 1585773540.395898628 seconds
  [Time delta from previous captured frame: 0.000027658 seconds]
  [Time delta from previous displayed frame: 0.000027658 seconds]
  [Time since reference or first frame: 5.052476456 seconds]
  Frame Number: 33
  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:udp:data]
  [Coloring Rule Name: UDP]
  [Coloring Rule String: udp]
```

Рис. 14

## Ethernet II

Протокол канального уровня. Предназначен для передачи данных между физическими устройствами. Содержит адреса источника и назначения (mac-адреса устройств).

На рисунке 15 показано как выглядит формат кадра для Ethernet II

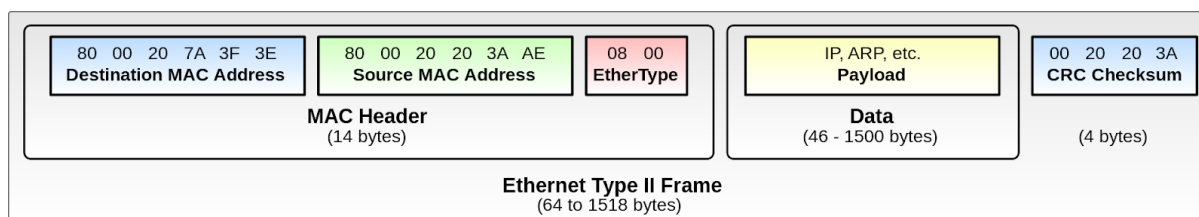


Рис.15

На рисунке 16 показана информация содержащаяся в этом протоколе

```
▼ Destination: Tp-LinkT_1c:a4:84 (98:da:c4:1c:a4:84)
  Address: Tp-LinkT_1c:a4:84 (98:da:c4:1c:a4:84)
  .... ..0. .... = LG bit: Globally unique address (factory default)
  .... ..0. .... = IG bit: Individual address (unicast)
▼ Source: LcfcHefe_2d:0a:c7 (54:e1:ad:2d:0a:c7)
  Address: LcfcHefe_2d:0a:c7 (54:e1:ad:2d:0a:c7)
  .... ..0. .... = LG bit: Globally unique address (factory default)
  .... ..0. .... = IG bit: Individual address (unicast)
Type: IPv4 (0x0800)
```

Рис.16

## Internet Protocol Version 4 (IPv4)

Протокол сетевого уровня.

В первом байте храниться информация о версии ip-протокола и о размере хедера. Далее идет информация о флагах: информация о фрагментации данных, информация о том последний или промежуточный фрагмент рассматривается и др. Также здесь храниться информация об ip-адреса источника и приемника.

На рисунке 17 показана структура IPv4 протокола.

IPv4 Header Format																																				
Отступ	Октет	0								1								2								3										
Октет	Бит	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7			
0	0	Версия				Размер заголовка				Differentiated Services Code Point				Explicit Congestion Notification				Размер пакета (полный)																		
4	32	Идентификатор																Флаги				Смещение фрагмента														
8	64	Время жизни								Протокол								Контрольная сумма заголовка																		
12	96	IP-адрес источника																																		
16	128	IP-адрес назначения																																		
20	160	Опции (если размер заголовка > 5)																																		
20 или 24+	160 или 192+	Данные																																		

Рис.17

На рисунке 18 показаны те данные протокола IPv4, которые были получены.

```

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: 60
Identification: 0x2d92 (11666)
▼ Flags: 0x0000
  0... .... = Reserved bit: Not set
  .0... .... = Don't fragment: Not set
  ..0. .... = More fragments: Not set
  ...0 0000 0000 0000 = Fragment offset: 0
Time to live: 6
Protocol: UDP (17)
Header checksum: 0xa812 [validation disabled]
[Header checksum status: Unverified]
Source: 192.168.0.105
Destination: 184.175.101.76

```

Рис.18

## User Datagram Protocol (UDP)

Протокол транспортного уровня.

Здесь в первой паре байтов хранится порт отправителя, во второй паре - порт получателя. В следующих байтах хранится длина сообщения и контрольная сумма и сами данные (см. рис 19).

Биты	0 - 15	16 - 31
0-31	Порт отправителя (Source port)	Порт получателя (Destination port)
32-63	Длина датаграммы (Length)	Контрольная сумма (Checksum)
64-...	Данные (Data)	



Рис.19

На рисунке 20 показано, как эта структура выглядит для данной лабораторной работы.

```
Source Port: 47081
► Destination Port: 33449
Length: 40
Checksum: 0xf0f8 [unverified]
[Checksum Status: Unverified]
[Stream index: 16]
```

Рис. 20

Data

Полученные данные

ICMP-ответ

На рисунке 21 показано как выглядит ICMP-ответ:

```
Frame 37: 70 bytes on wire (560 bits), 70 bytes captured (560 bits) on interface 0
Ethernet II, Src: Tp-LinkT_1c:a4:84 (98:da:c4:1c:a4:84), Dst: LcfcHefe_2d:0a:c7 (54:e1:ad:2d:0a:c7)
Internet Protocol Version 4, Src: 93.191.60.217, Dst: 192.168.0.105
Internet Control Message Protocol
```

Рис. 21

*Более подробное описание ICMP находится в описании структуры трафика утилиты `ping`.*

Frame 37

Информация о пакете. (см. рис. 22)



```

▶ Interface id: 0 (enp2s0)
Encapsulation type: Ethernet (1)
Arrival Time: Apr  1, 2020 23:39:00.396773742 MSK
[Time shift for this packet: 0.000000000 seconds]
Epoch Time: 1585773540.396773742 seconds
[Time delta from previous captured frame: 0.000169715 seconds]
[Time delta from previous displayed frame: 0.000169715 seconds]
[Time since reference or first frame: 5.053351570 seconds]
Frame Number: 37
Frame Length: 70 bytes (560 bits)
Capture Length: 70 bytes (560 bits)
[Frame is marked: False]
[Frame is ignored: False]
[Protocols in frame: eth:ethertype:ip:icmp:ip:udp]
[Coloring Rule Name: ICMP errors]
[Coloring Rule String: icmp.type eq 3 || icmp.type eq 4 || icmp.type eq 5

```

Рис.22.

## Ethernet II

Протокол канального уровня. Предназначен для передачи данных между физическими устройствами. Содержит адреса источника и назначения (mac-адреса устройств).

На рисунке 23 показана информация содержащаяся в этом протоколе

```

▼ Destination: LcfcHefe_2d:0a:c7 (54:e1:ad:2d:0a:c7)
  Address: LcfcHefe_2d:0a:c7 (54:e1:ad:2d:0a:c7)
    ....0. .... = LG bit: Globally unique address (factory default)
    ....0. .... = IG bit: Individual address (unicast)
▼ Source: Tp-LinkT_1c:a4:84 (98:da:c4:1c:a4:84)
  Address: Tp-LinkT_1c:a4:84 (98:da:c4:1c:a4:84)
    ....0. .... = LG bit: Globally unique address (factory default)
    ....0. .... = IG bit: Individual address (unicast)
Type: IPv4 (0x0800)

```

Рис. 23

## Internet Protocol Version 4 (IPv4)

Протокол сетевого уровня.

В первом байте храниться информация о версии ip-протокола и о размере хедера. Далее идет информация о флагах: информация о фрагментации данных, информация о том последний или промежуточный фрагмент рассматривается и др. Также здесь храниться информация об ip-адреса источника и приемника.

На рисунке 24 показаны те данные протокола IPv4, которые были получены.

```

0100 .... = Version: 4
.... 0101 = Header Length: 20 bytes (5)
▼ Differentiated Services Field: 0xc0 (DSCP: CS6, ECN: Not-ECT)
  1100 00.. = Differentiated Services Codepoint: Class Selector 6 (48)
  .... ..00 = Explicit Congestion Notification: Not ECN-Capable Transport (0)
Total Length: 56
Identification: 0x4945 (18757)
▼ Flags: 0x0000
  0... .... = Reserved bit: Not set
  .0.. .... = Don't fragment: Not set
  ..0. .... = More fragments: Not set
  ...0 0000 0000 0000 = Fragment offset: 0
Time to live: 254
Protocol: ICMP (1)
Header checksum: 0x1716 [validation disabled]
[Header checksum status: Unverified]
Source: 93.191.60.217
Destination: 192.168.0.105

```

Рис. 24

## Internet Control Message Protocol

Сетевой протокол. В основном используется для передачи сообщений об ошибках и других исключительных ситуациях, возникших при передаче данных, например, запрашиваемая услуга недоступна, или хост, или маршрутизатор не отвечают. Здесь хранится информация о контрольной сумме, порядковом номере, даваемом утилитой ping, информации о времени и сами данные.

На рисунке 25 представлена подробная информация, которая была передана этим протоколом. Здесь в теле ответа пришел заголовок и первые 64 бита оригинальных данных. Так как рассматриваемый запрос (с ttl = 6) не дошел до конечной точки, был возвращен первые 64 бита пакета, на котором произошла ошибка.

```

[Checksum Status: Good]
► Internet Protocol Version 4, Src: 192.168.0.105, Dst: 184.175.101.76
► User Datagram Protocol, Src Port: 41547, Dst Port: 33437

```

Рис.25

## Ответы на вопросы

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

Заголовок IPv4 содержит 20 байт. Поле данных содержит 40 байт, из них: заголовок UDP 8 байт, данные 32 байта. (см. рис. 18)

2. Как и почему изменяется поле TTL в следующих друг за другом ICMP-пакетах tracer? Для ответа на этот вопрос нужно проследить изменение TTL при передаче по маршруту, состоящему из более чем двух хопов.

Утилита traceroute посылает по 3 пакета с одинаковым TTL начиная с 1 и увеличивая это значение на 1 для каждого трёх последующих пакетов. Это необходимо для того, чтобы каждый узел сети до пункта назначения посылал ответ на пакет, у которого истекло время жизни, тем самым идентифицируя себя.

3. Чем отличаются ICMP-пакеты, генерируемые утилитой tracer, от ICMP-пакетов, генерируемых утилитой ping (см. предыдущее задание).

Для выполнения работы использовалась утилита traceroute для Linux, которая использует UDP, вместо ICMP. Исходя из найденной в интернете информации, ICMP-пакеты, генерируемые утилитой tracer, от ICMP-пакетов, генерируемых утилитой ping отличаются лишь значением TTL.

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

ICMP error пакеты приходят от узлов сети, сообщающих о том, что время жизни пакета истекло. Содержат заголовок IP и первые 64 байта датаграммы (см. рис. 26). На рисунке 27, полученных в ходе лабораторной работы данных, можно увидеть как раз такой вид пакета.

Time Exceeded Message

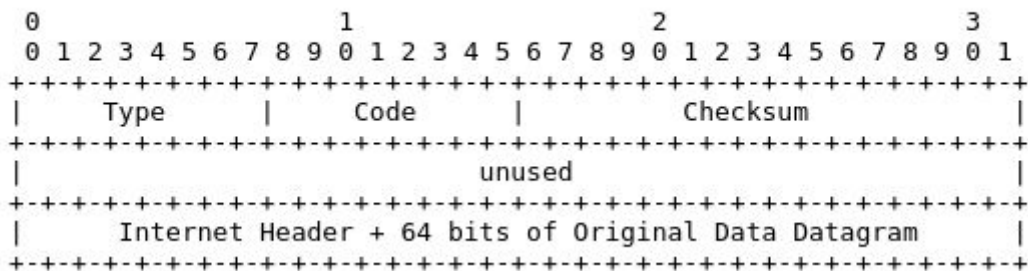


Рис. 26

```
[Checksum Status: Good]
▶ Internet Protocol Version 4, Src: 192.168.0.105, Dst: 184.175.101.76
▶ User Datagram Protocol, Src Port: 41547, Dst Port: 33437
```

Рис. 27

ICMP reply пакеты имеют формат такой же, как и ответ на ping запрос и сигнализируют о том, что пакет дошел до пункта назначения и построение маршрута можно завершить (в traceroute для этих целей используется ICMP Port unreachable). (см. рис 28). Пример протокола такого типа мы наблюдали при работе с утилитой ping (см. рис. 29).

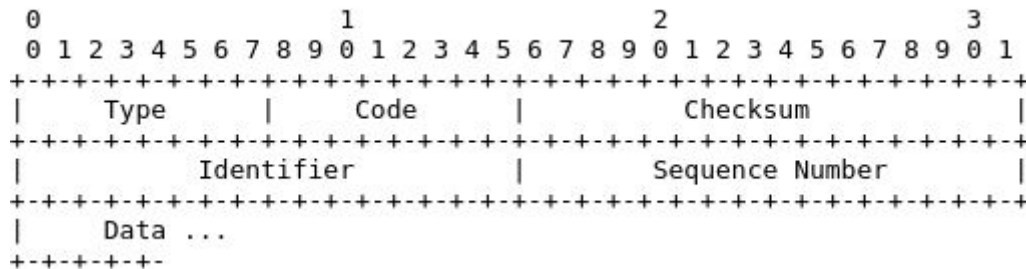


Рис. 28

```

Type: 8 (Echo (ping) request)
Code: 0
Checksum: 0x3c34 [correct]
[Checksum Status: Good]
Identifier (BE): 14527 (0x38bf)
Identifier (LE): 48952 (0xbf38)
Sequence number (BE): 11 (0x000b)
Sequence number (LE): 2816 (0x0b00)
[Response frame: 26]
Timestamp from icmp data: Apr  1, 2020 21:13:12.000000000 MSK
[Timestamp from icmp data (relative): 0.277522754 seconds]
▶ Data (92 bytes)
  
```

Рис. 29

5. Что изменится в работе tracert, если убрать ключ “-d”? Какой дополнительный трафик при этом будет генерироваться?

Утилита начнет преобразовывать IP адреса узлов сети в их строковые адреса, для этого потребуются дополнительные DNS запросы.

# Анализ HTTP-трафика

## Задание

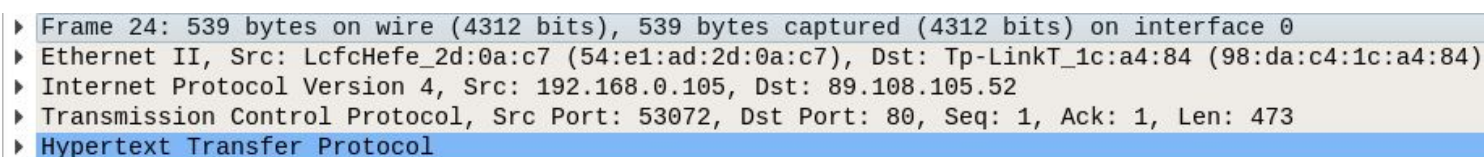
Необходимо отследить и проанализировать HTTP-трафик, создаваемый браузером при посещении Интернет-сайта, заданного по варианту. В списке захваченных пакетов необходимо проанализировать следующую пару HTTP-сообщений (запрос-ответ):

- GET-сообщение от клиента (браузера);
- ответ сервера

По результатам анализа собранной трассы покажите, каким образом протокол HTTP передавал содержимое страницы при первичном посещении страницы и при вторичном запросе-обновлении от браузера (т.е. при различных видах GET-запросов).

Сайт для анализа: <http://kalugina.ru/>

При первичном запросе страницы запрос имеет вид:



```
Frame 24: 539 bytes on wire (4312 bits), 539 bytes captured (4312 bits) on interface 0
Ethernet II, Src: LcfcHefe_2d:0a:c7 (54:e1:ad:2d:0a:c7), Dst: Tp-LinkT_1c:a4:84 (98:da:c4:1c:a4:84)
Internet Protocol Version 4, Src: 192.168.0.105, Dst: 89.108.105.52
Transmission Control Protocol, Src Port: 53072, Dst Port: 80, Seq: 1, Ack: 1, Len: 473
Hypertext Transfer Protocol
```

Рис 30

Первые 3 протокола уже были описаны при анализе утилиты ping и traceroute, поэтому сразу перейдем к описанию TCP и HTTP.

## Transmission Control Protocol (TCP)

Протокол транспортного уровня. Один из основных протоколов передачи данных интернета, предназначенный для управления передачей данных. TCP осуществляет надёжную передачу потока байтов от одного процесса к другому. TCP реализует управление потоком, управление перегрузкой, рукопожатие, надёжную передачу. На рисунке 31 показана структура TCP-протокола. Протокол хранит в себе порт источника, порт назначения, контрольную сумму, длину заголовка флаги и др. На рисунке 32 изображены данные, которые были получены при выполнении лабораторной работы.



Структура заголовка				
Бит	0 — 3	4 — 9	10 — 15	16 — 31
0	Порт источника, <b>Source Port</b>			Порт назначения, <b>Destination Port</b>
32	Порядковый номер, <b>Sequence Number (SN)</b>			
64	Номер подтверждения, <b>Acknowledgment Number (ACK SN)</b>			
96	Длина заголовка, <b>(Data offset)</b>	Зарезервировано	Флаги	Размер Окна, <b>Window size</b>
128	Контрольная сумма, <b>Checksum</b>			Указатель важности, <b>Urgent Point</b>
160	Опции (необязательное, но используется практически всегда)			
160/192+	Данные			

```

Source Port: 53072
Destination Port: 80
[Stream index: 10]
[TCP Segment Len: 473]
Sequence number: 1 (relative sequence number)
[Next sequence number: 474 (relative sequence number)]
Acknowledgment number: 1 (relative ack number)
1000 .... = Header Length: 32 bytes (8)
▼ Flags: 0x018 (PSH, 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: Set
  .... .... .0.. = Reset: Not set
  .... .... ..0. = Syn: Not set
  .... .... ...0 = Fin: Not set
  [TCP Flags: .....AP...]
Window size value: 502
[Calculated window size: 64256]
[Window size scaling factor: 128]
Checksum: 0x1440 [unverified]
[Checksum Status: Unverified]
Urgent pointer: 0
► Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
► [SEQ/ACK analysis]
► [Timestamps]
TCP payload (473 bytes)

```

Рис. 32

## Hypertext Transfer Protocol (HTTP)

Протокол прикладного уровня передачи данных. Структура http зависит от вида метода, но в общем случае выглядит так:

1. Стартовая строка -- определяет тип сообщения
2. Заголовки -- характеризуют тело сообщения, параметры передачи и прочие сведения

3. Тело сообщения — непосредственно данные сообщения. Обязательно должно отделяться от заголовков пустой строкой.

### Первичный запрос

Для первичного запроса http-протокол изображен на рисунке 33.



The image shows a network packet capture entry for an HTTP GET request. The packet is expanded to show the raw data, which is then decoded into a human-readable format. The request is a GET method to the root path (/) of the kalugina.ru domain, using HTTP/1.1. The user agent is identified as Mozilla/5.0 (X11; Linux x86\_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/80.0.3987.106 Safari/537.36. The request includes several headers: Host, Connection, Pragma, Cache-Control, Upgrade-Insecure-Requests, User-Agent, Accept, Accept-Encoding, and Accept-Language. The status bar at the bottom indicates the full request URI, the request number (1/2), and the frame numbers for the response (30) and the next request (44).

```
GET / HTTP/1.1\r\n
[Expert Info (Chat/Sequence): GET / HTTP/1.1\r\n]
Request Method: GET
Request URI: /
Request Version: HTTP/1.1
Host: kalugina.ru\r\n
Connection: keep-alive\r\n
Pragma: no-cache\r\n
Cache-Control: no-cache\r\n
Upgrade-Insecure-Requests: 1\r\n
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/80.0.3987.106 Safari/537.36\r\n
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9\r\n
Accept-Encoding: gzip, deflate\r\n
Accept-Language: ru-RU,ru;q=0.9,en-US;q=0.8,en;q=0.7\r\n
\r\n
[Full request URI: http://kalugina.ru/]
[HTTP request 1/2]
[Response in frame: 30]
[Next request in frame: 44]
```

Рис.33

### Первичный ответ

В ответ приходит HTTP response, содержащий текст страницы в html, данные о последнем изменении страницы и другие метаданные (см. рис 34)

```

HTTP/1.1 200 OK\r\n
  [Expert Info (Chat/Sequence): HTTP/1.1 200 OK\r\n]
    Response Version: HTTP/1.1
    Status Code: 200
    [Status Code Description: OK]
    Response Phrase: OK
    Date: Thu, 02 Apr 2020 16:36:11 GMT\r\n
    Server: Apache/2.4.6 (CentOS) OpenSSL/1.0.1e-fips PHP/5.4.16\r\n
    Last-Modified: Fri, 29 Apr 2016 19:26:14 GMT\r\n
    ETag: "954-531a49dc0e8ed"\r\n
    Accept-Ranges: bytes\r\n
  Content-Length: 2388\r\n
  Keep-Alive: timeout=5, max=100\r\n
  Connection: Keep-Alive\r\n
  Content-Type: text/html; charset=UTF-8\r\n
\r\n
[HTTP response 1/2]
[Time since request: 0.011541041 seconds]
[Request in frame: 24]
[Next request in frame: 44]
[Next response in frame: 45]
[Request URI: http://kalugina.ru/]
File Data: 2388 bytes
Line-based text data: text/html (46 lines)
<html>\n
\n
<table width="240" height="400" align="right" cellpadding="0" cellspacing="0">\n
  <tr>\n
    <td>\n
    </td>\n
  </tr>\n
</table>\n
\n
<p>\320\255\321\202\320\276\321\202 \321\201\320\260\320\271\321\202 \320\265\321\211\320\265
\n

```

Рис.34

### Повторный запрос

При повторном же запросе на сервер посылается информация о времени последней модификации запрашиваемой страницы и хэш страницы: поля If-modified-since и if-none-match. Т.е. для данной лабораторной работы сервер отправит обратно запрошенный ресурс с статусом 200, только если он будет изменен после указанной даты. (см. рис 35)



```

▼ GET / HTTP/1.1\r\n
  ▶ [Expert Info (Chat/Sequence): GET / HTTP/1.1\r\n]
    Request Method: GET
    Request URI: /
    Request Version: HTTP/1.1
    Host: kalugina.ru\r\n
    Connection: keep-alive\r\n
    Cache-Control: max-age=0\r\n
    Upgrade-Insecure-Requests: 1\r\n
    User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/80.0.3987.106 Safari/537.36\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9\r\n
    Accept-Encoding: gzip, deflate\r\n
    Accept-Language: ru-RU,ru;q=0.9,en-US;q=0.8,en;q=0.7\r\n
    If-None-Match: "954-531a49dc0e8ed"\r\n
    If-Modified-Since: Fri, 29 Apr 2016 19:26:14 GMT\r\n
    \r\n
    [Full request URI: http://kalugina.ru/]
    [HTTP request 1/1]
    [Response in frame: 222]

```

Рис 35

### Повторный ответ

В ответ сервер, если страница не была изменена, вернет код 304, либо новую версию страницы. И так как данные не были изменены, то в ответ на повторный запрос был возвращен код 304 (см.рис 36)

```

▼ HTTP/1.1 304 Not Modified\r\n
  ▶ [Expert Info (Chat/Sequence): HTTP/1.1 304 Not Modified\r\n]
    Response Version: HTTP/1.1
    Status Code: 304
    [Status Code Description: Not Modified]
    Response Phrase: Not Modified
    Date: Thu, 02 Apr 2020 16:36:36 GMT\r\n
    Server: Apache/2.4.6 (CentOS) OpenSSL/1.0.1e-fips PHP/5.4.16\r\n
    Connection: Keep-Alive\r\n
    Keep-Alive: timeout=5, max=100\r\n
    ETag: "954-531a49dc0e8ed"\r\n
    \r\n
    [HTTP response 1/1]
    [Time since request: 0.011150976 seconds]
    [Request in frame: 220]
    [Request URI: http://kalugina.ru/]

```

Рис. 36

# Анализ DNS-трафика

## Задание

Необходимо отследить и проанализировать трафик протокола DNS.

Сайт, использующийся для выполнения работы: kalugina.ru

## Структура DNS-пакета

Структура dns-пакета (см. рис. 37)

```
► Frame 4: 115 bytes on wire (920 bits), 115 bytes captured (920 bits) on interface 0
► Ethernet II, Src: Tp-LinkT_1c:a4:84 (98:da:c4:1c:a4:84), Dst: LcfcHefe_2d:0a:c7 (54:e1:ad:2d:0a:c7)
► Internet Protocol Version 4, Src: 192.168.0.1, Dst: 192.168.0.105
► User Datagram Protocol, Src Port: 53, Dst Port: 36884
► Domain Name System (response)
```

рис.37

DNS -- прикладной уровень. Чаще всего используется для получения IP-адреса по имени хоста (компьютера или устройства), получения информации о маршрутизации почты и/или обслуживающих узлах для протоколов в домене.

На рисунке 38 изображена структура DNS-пакета.



рис 38

На рисунке 39 и 40 изображены данные запроса и ответа, полученные в ходе выполнения работы, соответственно:

```
Transaction ID: 0x3fb5
▼ 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: 1
▼ Queries
  ▶ kalugina.ru: type A, class IN
▼ Additional records
  ▶ <Root>: type OPT
\[Response In: 17\]
```

рис 39

```
Transaction ID: 0x3fb5
▼ Flags: 0x8180 Standard query response, No error
  1... .. = Response: Message is a response
  .000 0... .. = Opcode: Standard query (0)
  .... ..0. .... = Authoritative: Server is not an authority for domain
  .... ..0. .... = Truncated: Message is not truncated
  .... ...1 .... = Recursion desired: Do query recursively
  .... ...1 .... = Recursion available: Server can do recursive queries
  .... .... .0.. .... = Z: reserved (0)
  .... .... ..0. .... = Answer authenticated: Answer/authority portion was not authenticated by the server
  .... .... ...0 .... = Non-authenticated data: Unacceptable
  .... .... ...0000 = Reply code: No error (0)
Questions: 1
Answer RRs: 1
Authority RRs: 2
Additional RRs: 1
▼ Queries
  ▶ kalugina.ru: type A, class IN
▼ Answers
  ▶ kalugina.ru: type A, class IN, addr 89.108.105.52
▼ Authoritative nameservers
  ▶ kalugina.ru: type NS, class IN, ns ns1.familydomain.ru
  ▶ kalugina.ru: type NS, class IN, ns ns2.familydomain.ru
▼ Additional records
  ▶ <Root>: type OPT
\[Request In: 16\]
[Time: 0.001572203 seconds]
```

рис 40

## Ответы на вопросы

1. Почему адрес, на который отправлен DNS-запрос, не совпадает с адресом посещаемого сайта?

Потому, что DNS запрос отправляется на адрес DNS сервера, чтобы по строковому адресу сайта узнать его IP.

## 2. Какие бывают типы DNS-запросов?

DNS запросы бывают прямыми, когда по имени хоста определяется его IP, и обратными, когда по IP определяется имя хоста. На Рис 39 и 40 представлен прямой DNS запрос.

## 3. В какой ситуации нужно выполнять независимые DNS-запросы для получения содержащихся на сайте изображений?

В случае, когда изображения представлены ссылкой на другое доменное имя. В этом случае необходимо делать DNS запросы для определения IP адресов этих доменов, чтобы получить изображения.

# Анализ ARP-трафика

## Задание

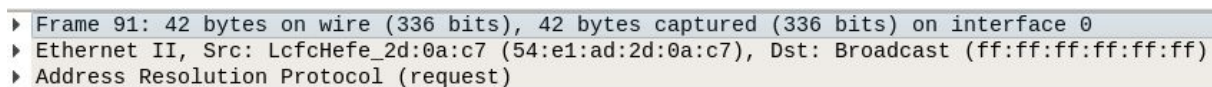
Необходимо отследить и проанализировать трафик протокола ARP.

Сайт, использующийся для выполнения работы: kalugina.ru

## Описание

Для отправки пакета компьютеру необходимо сначала получить MAC-адрес роутера. Для этого роутер отправляет широковещательное сообщение с ARP-запросом получателю с MAC-адресом FF:FF:FF:FF:FF:FF, которое принимается всеми компьютерами в сети, для получения MAC-адреса устройства с IP-адресом 192.168.0.1. В ответ роутер отправляет свой MAC-адрес, который записывается компьютером в кэшированную ARP-таблицу соответствий для дальнейшего использования.

На рис. 41 представлена структура запроса.



```
► Frame 91: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface 0
► Ethernet II, Src: LcfcHefe_2d:0a:c7 (54:e1:ad:2d:0a:c7), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
► Address Resolution Protocol (request)
```

Рис. 41

Frame 91 и Ethernet II были рассмотрены в пунктах выше. Нас интересует Address Resolution Protocol (ARP)

## Address Resolution Protocol (ARP)

### Структура запроса

Протокол сетевого уровня, предназначенный для определения MAC-адреса по IP-адресу другого компьютера. На рисунке 42 изображена структура ARP-пакетов, в котором: HTYPE -- номер протокола, который хранится в этом поле (например, из данных в лабораторной работе - Ethernet имеет номер 0x0001), PTYPE -- код сетевого протокола. (для IPv4 = 0x0800), HLEN длина физического адреса в байтах (адреса Ethernet имеют длину 6 байт), PLEN -- длина логического адреса в байтах (IPv4 адреса имеют длину 4 байта), operation -- код операции отправителя: 1 в случае запроса и 2 в случае ответа, SHA - физический адрес отправителя, SPA -- логический адрес отправителя, THA - физический адрес получателя (поле пусто при запросе), TPA -- логический адрес получателя.

На рисунке 43 представлены полученные данные в ходе выполнения лабораторной работы.

+	Bits 0 — 7	8 — 15	16 — 31
0	Hardware type (HTYPE)		Protocol type (PTYPE)
32	Hardware length (HLEN)	Protocol length (PLEN)	Operation (OPER)
64	Sender hardware address (SHA)		
?	Sender protocol address (SPA)		
?	Target hardware address (THA)		
?	Target protocol address (TPA)		

Рис.42

```
Hardware type: Ethernet (1)
Protocol type: IPv4 (0x0800)
Hardware size: 6
Protocol size: 4
Opcode: request (1)
Sender MAC address: LcfcHefe_2d:0a:c7 (54:e1:ad:2d:0a:c7)
Sender IP address: 192.168.0.105
Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
Target IP address: 192.168.0.1
```

Рис. 43

### Структура ответа

Структура ответа имеет аналогичный вид, описанный в структуре запроса. На рисунке 44 представлены данные, полученные в ходе выполнения лабораторной работы.

```
Hardware type: Ethernet (1)
Protocol type: IPv4 (0x0800)
Hardware size: 6
Protocol size: 4
Opcode: reply (2)
Sender MAC address: Tp-LinkT_1c:a4:84 (98:da:c4:1c:a4:84)
Sender IP address: 192.168.0.1
Target MAC address: LcfcHefe_2d:0a:c7 (54:e1:ad:2d:0a:c7)
Target IP address: 192.168.0.105
```

Рис 44

## Ответы на вопросы

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

В захваченных ARP пакетах присутствует 3 MAC адреса:

54:e1:ad:2d:0a:c7 -- идентифицирующий компьютер (см. рис. 43)

98:da:c4:1c:a4:84 -- идентифицирующий wifi роутер (см. рис. 43)

ff:ff:ff:ff:ff:ff -- широковещательный адрес. Используется для передачи пакетов всем устройствам локальной сети. (см. рис 40)

Эти адреса позволяют определить физический узел сети на канальном уровне.

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

В захваченных HTTP пакетах присутствует 2 MAC адреса:

54:e1:ad:2d:0a:c7 -- идентифицирующий компьютер

98:da:c4:1c:a4:84 -- идентифицирующий wifi роутер

В HTTP-пакетах также можно увидеть, что присутствующие MAC-адреса - адреса отправителя и получателя в сети Ethernet. Отправитель – устройство в сети; получатель – хост (роутер), использующийся для перенаправления фрейма на требуемый адрес в иной сети, а также для получения ответа и его пересылки изначальному отправителю. В отличие от ARP, у HTTP не наблюдается использования MAC-адреса FF:FF:FF:FF:FF:FF.

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

Т.к. запрос широковещательный, то другие устройства сети, получив этот запрос, могут добавить в ARP-таблицу информацию об отправителе.



# Анализ трафика утилиты nslookup

## Задание

Необходимо отследить и проанализировать трафик протокола DNS, сгенерированный в результате выполнения следующих действий

1. Настроить Wireshark-фильтр: "ip.addr == ваш\_IP\_адрес".
2. Запустить в командной строке команду "nslookup адрес\_сайта\_по\_варианту".
3. Дождаться отправки трёх DNS-запросов и трёх DNS-ответов (в работе нужно использовать только последние из них, т.к. первые два набора запросов/ответов специфичны для nslookup и не генерируются другими сетевыми приложениями).
4. Повторить предыдущие два шага, используя команду: "nslookup -type=NS имя\_сайта\_по\_варианту".

## Структура

### DNS-запрос

Запрос, генерируемый утилитой nslookup показан на рисунке 45. Здесь можно увидеть тип запроса kalugina.ru: type A. Это значит, что запрашиваемый запрос идет для IPv4. Для IPv6 тип запроса будет равен AAAA.

Кроме того, так как не был указан конкретный DNS-сервер, Ethernet-запрос отправлялся в wi-fi роутер, при указании dns-сервера - запрос будет отправляться напрямую в dns-сервер.

```
▶ Frame 219: 82 bytes on wire (656 bits), 82 bytes captured (656 bits) on interface 0
▶ Ethernet II, Src: LcfcHefe_2d:0a:c7 (54:e1:ad:2d:0a:c7), Dst: Tp-LinkT_1c:a4:84 (98:da:c4:1c:a4:84)
▶ Internet Protocol Version 4, Src: 192.168.0.105, Dst: 192.168.0.1
▶ User Datagram Protocol, Src Port: 37948, Dst Port: 53
▼ Domain Name System (query)
  Transaction ID: 0xa636
  ▼ 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: 1
  ▼ Queries
    ▼ kalugina.ru: type A, class IN
      Name: kalugina.ru
      [Name Length: 11]
      [Label Count: 2]
      Type: A (Host Address) (1)
      Class: IN (0x0001)
    ▼ Additional records
      ▶ <Root>: type OPT
      [Response In: 220]
```

Рис. 45



## DNS-ответ

На рисунке 46 изображен ответ на запрос. В поле answer можно увидеть addr - ip-адрес для kalugina.ru.

```
Transaction ID: 0xa636
▼ Flags: 0x8180 Standard query response, No error
  1... .. = Response: Message is a response
  .000 0... .. = Opcode: Standard query (0)
  .... 0... .. = Authoritative: Server is not an authority for domain
  .... 0... .. = Truncated: Message is not truncated
  .... 1... .. = Recursion desired: Do query recursively
  .... 1... .. = Recursion available: Server can do recursive queries
  .... 0... .. = Z: reserved (0)
  .... 0... .. = Answer authenticated: Answer/authority portion was not authenticated by the server
  .... 0... .. = Non-authenticated data: Unacceptable
  .... 0000 = Reply code: No error (0)
Questions: 1
Answer RRs: 1
Authority RRs: 2
Additional RRs: 1
▼ Queries
  ▼ kalugina.ru: type A, class IN
    Name: kalugina.ru
    [Name Length: 11]
    [Label Count: 2]
    Type: A (Host Address) (1)
    Class: IN (0x0001)
  ▼ Answers
    ▶ kalugina.ru: type A, class IN, addr 89.108.105.52
  ▼ Authoritative nameservers
    ▶ kalugina.ru: type NS, class IN, ns ns2.familydomain.ru
    ▶ kalugina.ru: type NS, class IN, ns ns1.familydomain.ru
  ▼ Additional records
    ▶ <Root>: type OPT
    [Request In: 219]
    [Time: 0.012679311 seconds]
```

Рис 46

## Запросы и ответы типа NS

Если обычный DNS-запрос напрямую ставит соответствие хоста и ip-адреса, то запросы типа NS. Ставит соответствие адрес узла, отвечающего за доменную зону (т.е. не напрямую, а ставит ip того, кто знает об искомом ip-адресе)

Пример DNS-ответа такого типа изображен на рисунке 47.

```

Transaction ID: 0x8faa
▼ Flags: 0x8500 Standard query response, No error
  1... .. = Response: Message is a response
  .000 0... .. = Opcode: Standard query (0)
  ....1... .. = Authoritative: Server is an authority for domain
  ....0... .. = Truncated: Message is not truncated
  ....1... .. = Recursion desired: Do query recursively
  ....0... .. = Recursion available: Server can't do recursive queries
  ....0... .. = Z: reserved (0)
  ....0... .. = Answer authenticated: Answer/authority portion was not authenticated by the server
  ....0... .. = Non-authenticated data: Unacceptable
  ....0000 = Reply code: No error (0)
Questions: 1
Answer RRs: 1
Authority RRs: 2
Additional RRs: 2
▼ Queries
  ▼ kalugina.ru: type A, class IN
    Name: kalugina.ru
    [Name Length: 11]
    [Label Count: 2]
    Type: A (Host Address) (1)
    Class: IN (0x0001)
▼ Answers
  ▶ kalugina.ru: type A, class IN, addr 89.108.105.52
▼ Authoritative nameservers
  ▶ kalugina.ru: type NS, class IN, ns ns1.familydomain.ru
  ▶ kalugina.ru: type NS, class IN, ns ns2.familydomain.ru
▼ Additional records
  ▶ ns1.familydomain.ru: type A, class IN, addr 89.108.105.52
  ▶ ns2.familydomain.ru: type A, class IN, addr 88.198.34.136
[Request In: 732]
[Time: 0.010469666 seconds]

```

Рис. 47

## Ответы на вопросы

### 1. Чем различается трасса трафика в п.2 и п.4, указанных выше?

В случае с запросом из п.2 в DNS-ответе содержался IP адрес указанного сайта, в DNS-ответе из п.4 содержались имена авторитативный DNS серверов, содержащие полную копию файла доменной зоны, в которой находится указанный сайт (см. рис. 47)

DNS	82 Standard query 0x18c9 NS kalugina.ru OPT
DNS	131 Standard query response 0x18c9 NS kalugina.ru NS ns1.familydomain.ru NS ns2.familydomain.ru OPT

Рис 47

### 2. Что содержится в поле «Answers» DNS-ответа?

В этом поле содержатся ответы на DNS запрос.

Для запроса из п.2 это имя хоста, класс и тип записи, время жизни записи, размер данных и запрашиваемый адрес хоста. (см. рис. 46)

Для запроса из п.4 это 2 ответа, содержащие имя хоста, класс и тип записи, время жизни записи, размер данных и имена авторитативный серверов. (см. рис. 47)

3. Каковы имена серверов, возвращающих авторитативный (authoritative) отклик?

На рисунке 47 - ответа типа NS, есть поле, в котором написаны адреса авторитативных серверов, а именно: ns1.familydomain.ru и ns2.familydomain.ru.

Авторитативные имена серверов содержат префикс вида ns1, ns2 и т. д.

# Анализ FTP-трафика

## Задание

Необходимо отследить и проанализировать трафик протокола FTP, сгенерированный в результате выполнения следующих действий:

- настроить Wireshark-фильтр «ftp || ftp-data»;
- скачать в браузере небольшой файл с соответствующего варианту FTP-сервера в Интернете.

Так как ftp-сайтов с инициалами kmm или mmk не существует, был выбран сайт, в котором встречается инициалы имени и фамилии (mk): <ftp://ftp.mk.bsdclub.org/>

## Структура

### FTP

На рисунке 49 представлена структура полученного пакета

```
▶ Frame 72: 72 bytes on wire (576 bits), 72 bytes captured (576 bits) on interface 0
▶ Ethernet II, Src: Tp-LinkT_1c:a4:84 (98:da:c4:1c:a4:84), Dst: LcfcHefe_2d:0a:c7 (54:e1:ad:2d:0a:c7)
▶ Internet Protocol Version 4, Src: 202.239.78.90, Dst: 192.168.0.105
▶ Transmission Control Protocol, Src Port: 21, Dst Port: 51180, Seq: 489, Ack: 186, Len: 6
▶ File Transfer Protocol (FTP)
```

Рис 49

FTP -- протокол прикладного уровня для передачи файлов по сети.

FTP-request содержит команду для выполнения. Для скачивания файла с сайта ftp послал 10 команд-запросов и получил 10 ftp-request'ов. В таблице 2 находится последовательное описание выполнения скачивания файла

Таблица 2

No	Request	Описание	Response	Описание
1	USER	Имя пользователя для входа на сервер	Guest login ok, type your name as password	Гостевой логин в порядке, введите ваше имя как пароль
2	PASS	Пароль	Guest login ok	Гостевой логин в порядке
3	SYST	Возвращает тип системы	UNIX	UNIX
4	PWD	Возвращает текущий каталог	"/" is current directory	"/" - текущий каталог
5	TYPE	Установить тип	Type set to I	Бинарные данные

		передачи файла		
6	SIZE	Возвращает размер файла	1581	Размер в битах
7	CWD	Сменить каталог.	pub/FreeBSD/TP530Cs/ Not a directory	Переданный аргумент - не каталог
8	PASW	Сервер возвращает на адрес и порт, к которому нужно подключиться, чтобы забрать данные	Entering Passing Mode	Вход в режим прохождения (необходимо выполнить определенную последовательность для скачивания)
9	RETR	Скачать файл.	Opening BINARY mode data connection for 'pub/FreeBSD/TP530Cs/XF86Config'	Открытие соединения для передачи данных в бинарном режиме
10	QUIT	Выход	Service closing control connection	Закрытие соединения

Пример запроса и ответа на примере команды RETR изображен на рисунках 50 и 51 соответственно.

```

▼ File Transfer Protocol (FTP)
  ▼ RETR /pub/FreeBSD/TP530Cs/XF86Config\r\n
    Request command: RETR
    Request arg: /pub/FreeBSD/TP530Cs/XF86Config
    [Current working directory: /]

```

Рис. 50

```

▼ 150 Opening BINARY mode data connection for '/pub/FreeBSD/TP530Cs/XF86Config' (1581 bytes).\r\n
  Response code: File status okay; about to open data connection (150)
  Response arg: Opening BINARY mode data connection for '/pub/FreeBSD/TP530Cs/XF86Config' (1581 bytes).
  [Current working directory: /]

```

Рис. 51

## FTP-DATA

FTP-data используется для передачи данных.

Пример FTP-DATA изображен на рисунке 52

```
FTP Data (1581 bytes data)
[Setup frame: 52]
[Setup method: PASV]
[Command: PASV]
Command frame: 51
[Current working directory: /]
▼ Line-based text data (71 lines)
  # XF86Config for IBM ThinkPad 530CS\n
  \n
  Section "Files"\n
    RgbPath\t"/usr/X11R6/lib/X11/rgb"\n
  \n
    FontPath\t"/usr/X11R6/lib/X11/fonts/misc/"\n
    FontPath\t"/usr/X11R6/lib/X11/fonts/Type1/"\n
    FontPath\t"/usr/X11R6/lib/X11/fonts/Speedo/"\n
    FontPath\t"/usr/X11R6/lib/X11/fonts/75dpi/"\n
    FontPath\t"/usr/X11R6/lib/X11/fonts/100dpi/"\n
  EndSection\n
  \n
  Section "ServerFlags"\n
  EndSection\n
  \n
  Section "Keyboard"\n
    Protocol\t"Standard"\n
    AutoRepeat\t500 5\n
```

Рис. 52

## Ответы на вопросы

1. Сколько байт данных содержится в пакете FTP-DATA?

В пакете FTP-DATA максимум может содержаться 1448 байт данных

Protocol	Length	Info
FTP-DATA		1514 FTP Data: 1448 bytes

Рис 53

Это связано с тем, что MTU=1500, куда входит заголовок IP -- 20 байт и заголовок TCP -- 32 байта (см. рис 53)

2. Как выбирается порт транспортного уровня, который используется для передачи FTP-пакетов?

Для передачи ftp пакетов, клиент открывает случайный порт в диапазоне от 1025. На сервере используется порт 21. (см. рис. 54)

Transmission Control Protocol, Src Port: 37780 (37780), Dst Port: 21 (21), Seq: 48, Ack: 98, Len: 5

Рис 54

### 3. Чем отличаются пакеты FTP от FTP-DATA?

Пакеты FTP используются для передачи команд. FTP-DATA используется для передачи данных. (см. рис. 50-52)

## Анализ DHCP-трафика

### Задание

Необходимо отследить и проанализировать трафик протокола DHCP, сгенерированный в результате выполнения действий, описанных в методических материалах.

### Структура

На рисунке 55 показан один цикл сброса-запроса ip-адреса

Source	Destination	Protocol	Length	Info
192.168.0.106	192.168.0.1	DHCP	342	DHCP Release
0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover
192.168.0.1	192.168.0.106	DHCP	590	DHCP Offer
0.0.0.0	255.255.255.255	DHCP	342	DHCP Request
192.168.0.1	192.168.0.106	DHCP	590	DHCP ACK

Рис. 55

BOOTP -- сетевой протокол, используемый для автоматического получения клиентом IP-адреса.

DHCP -- сетевой протокол, позволяющий сетевым устройствам автоматически получать IP-адрес и другие параметры, необходимые для работы в сети TCP/IP. На рисунке 56 изображена структура dhcp-пакета.

Протокол DHCP является надстройкой над BOOTP и позволяет серверу выделять IP-адреса клиентам динамически на ограниченный срок.



Dynamic Host Configuration Protocol				
Bit Offset	0–15		16–31	
0	OpCode	Hardware Type	Hardware Length	Hops
32	Transaction ID			
64	Seconds Elapsed		Flags	
96	Client IP Address			
128	Your IP Address			
160	Server IP Address			
196	Gateway IP Address			
228+	Client Hardware Address (16 bytes)			
	Server Host Name (64 bytes)			
	Boot File (128 bytes)			
	Options			

Рис. 56

Realice. Сброс текущего ip

На рисунке 57 показан запрос для сброса ip адреса. Тип этого запроса -- Boot Request

Рис. 57

Discover. Обнаружение DHCP

В начале выполняется широковещательный запрос по всей физической сети с целью обнаружить доступные DHCP-серверы. Отправляется сообщение типа DHCPDISCOVER, при этом в качестве IP-адреса источника указывается 0.0.0.0 (т.к. компьютер ещё не имеет собственного IP-адреса после сброса) (см. рис. 55), а в качестве адреса назначения — широковещательный адрес 255.255.255.255. (см. рис 55 и 58)



Данные представлены на рисунке 59.

[illegible]

Рис 59.

## Request. Запрос DHCP

Выбрав одну из конфигураций, предложенных DHCP-серверами, клиент отправляет запрос DHCP (**DHCPREQUEST**). Он рассылается широковещательно; при этом к опциям, указанным клиентом в сообщении DHCPDISCOVER, добавляется специальная опция — идентификатор сервера — указывающая адрес DHCP-сервера, выбранного клиентом (в данном случае — 192.168.0.1). (См. рис. 60)

### ▼ Bootstrap Protocol (Request)

```
Message type: Boot Request (1)
Hardware type: Ethernet (0x01)
Hardware address length: 6
Hops: 0
Transaction ID: 0x5f66cf60
Seconds elapsed: 0
▼ Bootp flags: 0x0000 (Unicast)
    0... .... = Broadcast flag: Unicast
    .000 0000 0000 0000 = Reserved flags: 0x0000
Client IP address: 0.0.0.0
Your (client) IP address: 0.0.0.0
Next server IP address: 0.0.0.0
Relay agent IP address: 0.0.0.0
Client MAC address: LcfcHefe_2d:0a:c7 (54:e1:ad:2d:0a:c7)
Client hardware address padding: 0000000000000000000000
Server host name not given
Boot file name not given
Magic cookie: DHCP
▼ Option: (53) DHCP Message Type (Request)
    Length: 1
    DHCP: Request (3)
▼ Option: (54) DHCP Server Identifier
    Length: 4
    DHCP Server Identifier: 192.168.0.1
▼ Option: (50) Requested IP Address
    Length: 4
    Requested IP Address: 192.168.0.106
▼ Option: (12) Host Name
    Length: 6
    Host Name: marina
▼ Option: (55) Parameter Request List
    Length: 13
    Parameter Request List Item: (1) Subnet Mask
    Parameter Request List Item: (28) Broadcast Address
    Parameter Request List Item: (2) Time Offset
    Parameter Request List Item: (3) Router
    Parameter Request List Item: (15) Domain Name
    Parameter Request List Item: (6) Domain Name Server
    Parameter Request List Item: (119) Domain Search
    Parameter Request List Item: (12) Host Name
    Parameter Request List Item: (44) NetBIOS over TCP/IP Name Server
    Parameter Request List Item: (47) NetBIOS over TCP/IP Scope
    Parameter Request List Item: (26) Interface MTU
    Parameter Request List Item: (121) Classless Static Route
    Parameter Request List Item: (42) Network Time Protocol Servers
▼ Option: (255) End
    Option End: 255
Padding: 0000000000000000000000000000000000000000
```

Рис 60

## АСК. Подтверждение ДНСР

Наконец, сервер подтверждает запрос и направляет это подтверждение (DNCPACK) клиенту. После этого клиент должен настроить свой сетевой интерфейс, используя предоставленные опции. (см. рис 61)

- ▼ Bootstrap Protocol (ACK)

Message type: Boot Reply (2)

Hardware type: Ethernet (0x01)

Hardware address length: 6

Hops: 0

Transaction ID: 0x5f66cf60

Seconds elapsed: 0

▼ Bootp flags: 0x0000 (Unicast)

0... .. = Broadcast flag: Unicast

```
.000 0000 0000 0000 = Reserved flags: 0x0000
```

Client IP address: 0.0.0.0

Your (client) IP address: 192.168.0.106

```
Next server IP address: 0.0.0.0
```

Relay agent IP address: 0.0.0.0

Client MAC address: LcfcHefe\_2d:0a:c7 (54:e1:ad:2d:0a:c7)

```
Client hardware address padding: 000000000000000000000000
```

Server host name not given

Boot file name not given

Magic cookie: DHCP

▼ Option: (53) DHCP Message Type (ACK)

Length: 1

DHCP: ACK (5)

▼ Option: (54) DHCP Server Identifier

Length: 4

DHCP Server Identifier: 192.168.0.1

▼ Option: (51) IP Address Lease Time

Length: 4

IP Address Lease Time: (7200s) 2 hours

▼ Option: (1) Subnet Mask

Length: 4

Subnet Mask: 255.255.255.0

▼ Option: (3) Router

Length: 4

Router: 192.168.0.1

▼ Option: (6) Domain Name Server

Length: 4

Domain Name Server: 192.168.0.1

▼ Option: (255) End

Option End: 255

[illegible]

Рис 61

## Временная диаграмма

На рисунке 63 представлена временная диаграмма, иллюстрирующая последовательность обмена DHCP-пакетами. Подробное описание последовательности можно найти выше, в пункте со структурой пакета



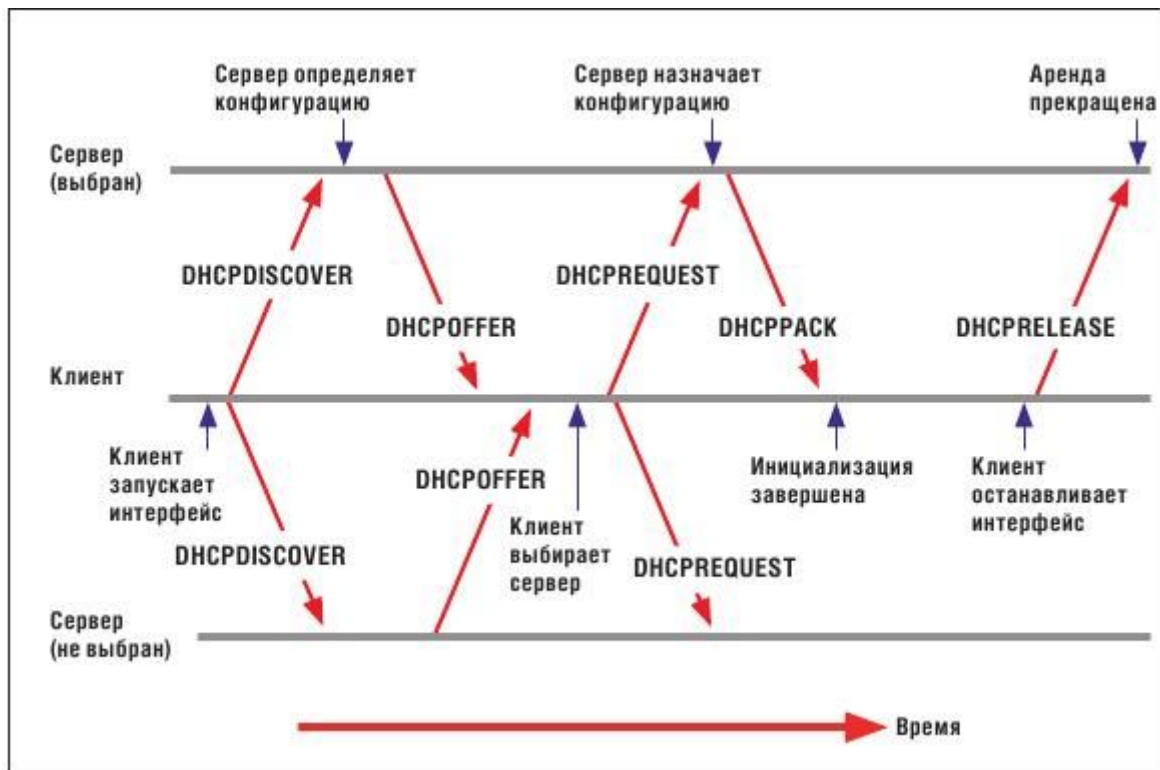


Рис. 63.

## Используемые порты

Для выполнения данных запросов-ответов используются порты 67 и 68.  
 На рисунке 64-67 изображены порты для Discover, Offer, Request, ASK соответственно.

```
User Datagram Protocol, Src Port: 68, Dst Port: 67
Bootstrap Protocol (Discover)
```

Рис 64

```
User Datagram Protocol, Src Port: 67, Dst Port: 68
Bootstrap Protocol (Offer)
```

Рис 65

```
User Datagram Protocol, Src Port: 68, Dst Port: 67
Bootstrap Protocol (Request)
```

Рис 66

```
User Datagram Protocol, Src Port: 67, Dst Port: 68
Bootstrap Protocol (ACK)
```

Рис. 67

## Ответы на вопросы

### 1. Чем различаются пакеты «DHCP Discover» и «DHCP Request»?

DHCP Discover посылается в качестве запроса на получение конфигураций от одного или более DHCP серверов, после их ответа выбирается одна из них и посылается DHCP Request, в котором указывается запрашиваемый IP адрес и идентификатор DHCP сервера. (см. рис 58, 60)

### 2. Как и почему менялись MAC- и IP-адреса источника и назначения в переданных DHCP-пакетах.

При отправке Discover и Request пакетов IP-адрес источника равен 0.0.0.0, т. к. ему не присвоен IP. IP-адрес и MAC-адрес назначения соответствуют широковещательным адресом, т. к. источнику неизвестно расположение DHCP-сервера. На рисунке 68 проиллюстрировано содержание Discover.

```
Ethernet II, Src: LcfcHefe_2d:0a:c7 (54:e1:ad:2d:0a:c7), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
Internet Protocol Version 4, Src: 0.0.0.0, Dst: 255.255.255.255
```

Рис 68.

При отправке Offer и Ack пакетов MAC и IP адреса источника соответствуют адресам DHCP сервера, MAC адрес — адрес назначения, IP адрес назначения — адрес, предлагаемый/подтвержденный IP адрес назначения. (см рис 69)

```
Ethernet II, Src: Tp-LinkT_1c:a4:84 (98:da:c4:1c:a4:84), Dst: LcfcHefe_2d:0a:c7 (54:e1:ad:2d:0a:c7)
Internet Protocol Version 4, Src: 192.168.0.1, Dst: 192.168.0.106
```

Рис 69

### 3. Каков IP-адрес DHCP-сервера?

IP-адрес DHCP-сервера - 192.168.0.1

```
▼ Option: (54) DHCP Server Identifier
  Length: 4
  DHCP Server Identifier: 192.168.0.1
```

Рис 70

### 4. Что произойдет, если очистить использованный фильтр “bootp”?

Отобразятся все пакеты, захваченные за время выполнения задания

# Анализ Skype-трафика

## Задание

Проанализировать трафик генерируемый программой skype при передаче текста, аудио, видео

## Структура документа

### Текст

Текстовые сообщения в скайпе передаются посредством TCP протокола, описанного выше, но зашифрованного при помощи SSL (Secure Sockets Layer). На рисунке 71 представлены данные, полученные при отправке сообщения.

### Secure Sockets Layer (SSL)

SSL (англ. — уровень защищённых сокетов) — криптографический протокол. Он хранит в себе тип контента, версию - в нашем случае TLS 1.2, размер и само сообщение.



```

▶ Frame 106004: 1270 bytes on wire (10160 bits), 1270 bytes captured (10160 bits) on interface 0
▶ Ethernet II, Src: LcfcHefe_2d:0a:c7 (54:e1:ad:2d:0a:c7), Dst: Tp-LinkT_1c:a4:84 (98:da:c4:1c:a4:84)
▶ Internet Protocol Version 4, Src: 192.168.0.106, Dst: 52.149.21.60
▼ Transmission Control Protocol, Src Port: 58720, Dst Port: 443, Seq: 26020, Ack: 11598, Len: 1216
    Source Port: 58720
    Destination Port: 443
    [Stream index: 455]
    [TCP Segment Len: 1216]
    Sequence number: 26020 (relative sequence number)
    [Next sequence number: 27236 (relative sequence number)]
    Acknowledgment number: 11598 (relative ack number)
    0101 .... = Header Length: 20 bytes (5)
    ▼ Flags: 0x018 (PSH, 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: Set
        .... .... .0.. = Reset: Not set
        .... .... ..0. = Syn: Not set
        .... .... ...0 = Fin: Not set
        [TCP Flags: .....AP...]
    Window size value: 501
    [Calculated window size: 64128]
    [Window size scaling factor: 128]
    Checksum: 0x3ed6 [unverified]
    [Checksum Status: Unverified]
    Urgent pointer: 0
    ▶ [SEQ/ACK analysis]
    ▶ [Timestamps]
    TCP payload (1216 bytes)
    TCP segment data (1216 bytes)
    ▶ [2 Reassembled TCP Segments (2656 bytes): #106003(1440), #106004(1216)]
    ▼ Secure Sockets Layer
        ▼ TLSv1.2 Record Layer: Application Data Protocol: http-over-tls
            Content Type: Application Data (23)
            Version: TLS 1.2 (0x0303)
            Length: 2651
            Encrypted Application Data: 00000000000000ad0d94d73224f9a90a66a2bcbdf6d8f8b...

```

Рис 71

## Аудио

Передается UDP-протоколом. На рисунке 72 показаны аудио-данные переданные в скайпе.

```

▶ Frame 322: 109 bytes on wire (872 bits), 109 bytes captured (872 bits) on interface 0
▶ Ethernet II, Src: Tp-LinkT_1c:a4:84 (98:da:c4:1c:a4:84), Dst: LcfcHefe_2d:0a:c7 (54:e1:ad:2d:0a:c7)
▶ Internet Protocol Version 4, Src: 176.53.227.14, Dst: 192.168.0.106
▼ User Datagram Protocol, Src Port: 63002, Dst Port: 55638
    Source Port: 63002
    Destination Port: 55638
    Length: 75
    Checksum: 0x9672 [unverified]
    [Checksum Status: Unverified]
    [Stream index: 0]
    ▼ Data (67 bytes)
        Data: 90683b58130f97640000e001bede00011247973954728c5b...
        [Length: 67]

```

Рис. 72

## Видео

Видео передается скайпом тоже UDP-протоколом. Отличить UDP-протокол с видео и UDP протокол с аудио можно, если посмотреть на размер пакетов. Пакеты с аудио-данными отличаются более чем в 10 раз. На рисунке 73 представлены и аудио и видео-данные, причем сверху аудио, снизу - видео.

192.168.0.106	176.53.227.14	UDP	125 55638 → 63002	Len=83
176.53.227.14	192.168.0.106	UDP	109 63002 → 55638	Len=67
192.168.0.106	176.53.227.14	UDP	1132 55638 → 63002	Len=1090
192.168.0.106	176.53.227.14	UDP	1132 55638 → 63002	Len=1090

Рис. 73

На рисунке 74 представлены данные, полученные при передаче видео.

```
► Frame 330: 1181 bytes on wire (9448 bits), 1181 bytes captured (9448 bits) on interface 0
► Ethernet II, Src: LcfcHefe_2d:0a:c7 (54:e1:ad:2d:0a:c7), Dst: Tp-LinkT_1c:a4:84 (98:da:c4:1c:a4:84)
► Internet Protocol Version 4, Src: 192.168.0.106, Dst: 176.53.227.14
▼ User Datagram Protocol, Src Port: 55638, Dst Port: 63002
  Source Port: 55638
  Destination Port: 63002
  Length: 1147
  Checksum: 0x0681 [unverified]
  [Checksum Status: Unverified]
  [Stream index: 0]
▼ Data (1139 bytes)
  Data: 90fa530d0d2a47920000fecdbede000212479ecd22098096...
  [Length: 1139]
```

Рис. 74

## Ответы на вопросы

1. Чем различаются пакета разных видов Skype-трафика (текст, аудио, видео)?

Подводя итоге написанного выше: Для передачи видео и аудио используется протокол UDP. Пакеты с аудио и видео различаются только размерами.

Для своей работы скайп может использовать следующие порты:

- 443/TCP
- 3478-3481/UDP
- 50000-60000/UDP

Таким образом, все текстовые сообщения были отправлены, используя порт 433, все аудиоматериалы были отправлены с порта 63002, а все видеоматериалы -- с 55638.

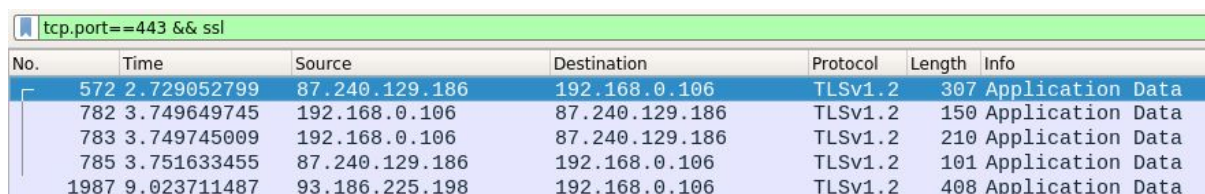
## 2. Какой Wireshark-фильтр следует использовать для независимой идентификации Skype-трафика разных видов (текст, аудио, видео)?

Так как скайп может выбрать единственный tcp порт -- то существует команда, которая будет справедлива для любого устройства и любой сессии: для идентификации текста необходимо установить фильтр: **tcp.port==443 && ssl** (см. рис.75)

Для работы с аудио и видеоматериалами скайп выбирает рандомный порт. Поэтому, написанные команды будут отличаться для каждой сессии:

Для идентификации аудиоданных: **udp.srcport==63002** (см. рис 76)

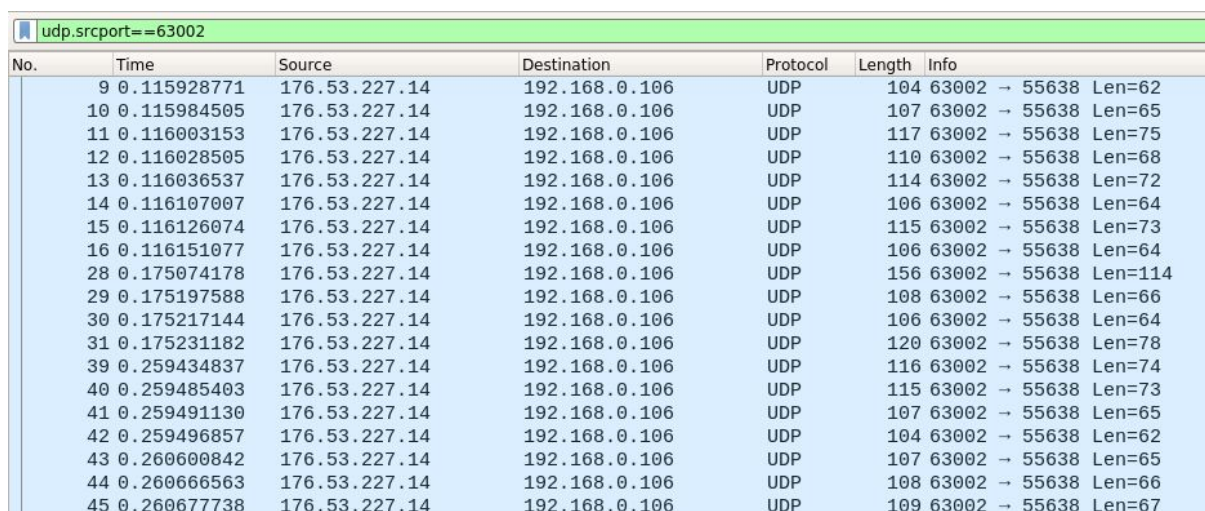
Для идентификации видеоданных: **udp.srcport==55638** (см. рис. 77)



Wireshark packet capture screenshot showing a filter **tcp.port==443 && ssl** applied. The packet list shows several TLSv1.2 packets (Application Data) between source IP 87.240.129.186 and destination IP 192.168.0.106.

No.	Time	Source	Destination	Protocol	Length	Info
572	2.729052799	87.240.129.186	192.168.0.106	TLSv1.2	307	Application Data
782	3.749649745	192.168.0.106	87.240.129.186	TLSv1.2	150	Application Data
783	3.749745009	192.168.0.106	87.240.129.186	TLSv1.2	210	Application Data
785	3.751633455	87.240.129.186	192.168.0.106	TLSv1.2	101	Application Data
1987	9.023711487	93.186.225.198	192.168.0.106	TLSv1.2	408	Application Data

Рис 75



Wireshark packet capture screenshot showing a filter **udp.srcport==63002** applied. The packet list shows numerous UDP packets (63002 -> 55638) between source IP 176.53.227.14 and destination IP 192.168.0.106.

No.	Time	Source	Destination	Protocol	Length	Info
9	0.115928771	176.53.227.14	192.168.0.106	UDP	104	63002 -> 55638 Len=62
10	0.115984505	176.53.227.14	192.168.0.106	UDP	107	63002 -> 55638 Len=65
11	0.116003153	176.53.227.14	192.168.0.106	UDP	117	63002 -> 55638 Len=75
12	0.116028505	176.53.227.14	192.168.0.106	UDP	110	63002 -> 55638 Len=68
13	0.116036537	176.53.227.14	192.168.0.106	UDP	114	63002 -> 55638 Len=72
14	0.116107007	176.53.227.14	192.168.0.106	UDP	106	63002 -> 55638 Len=64
15	0.116126074	176.53.227.14	192.168.0.106	UDP	115	63002 -> 55638 Len=73
16	0.116151077	176.53.227.14	192.168.0.106	UDP	106	63002 -> 55638 Len=64
28	0.175074178	176.53.227.14	192.168.0.106	UDP	156	63002 -> 55638 Len=114
29	0.175197588	176.53.227.14	192.168.0.106	UDP	108	63002 -> 55638 Len=66
30	0.175217144	176.53.227.14	192.168.0.106	UDP	106	63002 -> 55638 Len=64
31	0.175231182	176.53.227.14	192.168.0.106	UDP	120	63002 -> 55638 Len=78
39	0.259434837	176.53.227.14	192.168.0.106	UDP	116	63002 -> 55638 Len=74
40	0.259485403	176.53.227.14	192.168.0.106	UDP	115	63002 -> 55638 Len=73
41	0.259491130	176.53.227.14	192.168.0.106	UDP	107	63002 -> 55638 Len=65
42	0.259496857	176.53.227.14	192.168.0.106	UDP	104	63002 -> 55638 Len=62
43	0.260600842	176.53.227.14	192.168.0.106	UDP	107	63002 -> 55638 Len=65
44	0.260666563	176.53.227.14	192.168.0.106	UDP	108	63002 -> 55638 Len=66
45	0.260677738	176.53.227.14	192.168.0.106	UDP	109	63002 -> 55638 Len=67

Рис 76

udp.srcport==55638						
No.	Time	Source	Destination	Protocol	Length	Info
76	0.413845688	192.168.0.106	176.53.227.14	UDP	1066	55638 → 63002 Len=1024
77	0.413882354	192.168.0.106	176.53.227.14	UDP	1066	55638 → 63002 Len=1024
78	0.413897580	192.168.0.106	176.53.227.14	UDP	1064	55638 → 63002 Len=1022
79	0.413916647	192.168.0.106	176.53.227.14	UDP	797	55638 → 63002 Len=755
80	0.413929986	192.168.0.106	176.53.227.14	UDP	797	55638 → 63002 Len=755
81	0.413942977	192.168.0.106	176.53.227.14	UDP	797	55638 → 63002 Len=755
82	0.413957085	192.168.0.106	176.53.227.14	UDP	797	55638 → 63002 Len=755
83	0.413969447	192.168.0.106	176.53.227.14	UDP	797	55638 → 63002 Len=755
84	0.413983625	192.168.0.106	176.53.227.14	UDP	797	55638 → 63002 Len=755
85	0.414066108	192.168.0.106	176.53.227.14	UDP	797	55638 → 63002 Len=755
86	0.414152292	192.168.0.106	176.53.227.14	UDP	797	55638 → 63002 Len=755
87	0.414217035	192.168.0.106	176.53.227.14	UDP	1082	55638 → 63002 Len=1040
88	0.424633557	192.168.0.106	176.53.227.14	UDP	1082	55638 → 63002 Len=1040

Рис. 77