

федеральное государственное автономное образовательное учреждение
высшего образования
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

ОТЧЕТ

по лабораторной работе №3

«Анализ трафика компьютерных сетей с помощью утилиты
Wireshark»

по дисциплине «**Компьютерные сети**»

Вариант ЛР4

Автор: Кулаков Н. В.

Факультет: ПИиКТ

Группа: Р33312

Преподаватель: Алиев Т. И.



УНИВЕРСИТЕТ ИТМО

Санкт-Петербург 2023

1. Постановка задачи и исходные данные

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

В процессе произвести наблюдения за передаваемым трафиком с компьютера пользователя в Интернет и в обратном направлении с использованием специализированной утилиты Wireshark, которая позволяет наблюдать структуру передаваемых кадров, пакетов и сегментов данных различных сетевых протоколов. При выполнении УИР требуется проанализировать последовательности команд и назначение служебных данных, используемых для организации обмена данными в различных протоколах.

2. Выполнение

2.1. Анализ трафика утилиты ping

```
nikit@host lab-4 % ping -s 200 kulakov-pro.ru
PING kulakov-pro.ru (37.140.192.82) 200(228) bytes of data.
208 bytes from server51.hosting.reg.ru (37.140.192.82): icmp_seq=1 ttl=45 time=62.8 ms
208 bytes from server51.hosting.reg.ru (37.140.192.82): icmp_seq=2 ttl=45 time=53.9 ms
208 bytes from server51.hosting.reg.ru (37.140.192.82): icmp_seq=3 ttl=45 time=53.6 ms
208 bytes from server51.hosting.reg.ru (37.140.192.82): icmp_seq=4 ttl=45 time=50.2 ms
208 bytes from server51.hosting.reg.ru (37.140.192.82): icmp_seq=5 ttl=45 time=48.0 ms
```

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

ICMP пакет передается внутри пакета IP, который передается по Ethernet, максимальный размер кадра которого равен 1518 байт. Соответственно, при передаче пакет ICMP фрагментируется. Ниже представлен результат при передаче 6000 байт.

No	Time	Source	Destination	Protocol	Length	Info
3107	514.391415538	192.168.1.197	192.168.1.72	DNS	90	Standard query response 0x5f77 AAAA kulakov-pro.ru AAAA 2a09:f940:2:2:1:1:0:51
3108	514.391815733	192.168.1.72	37.140.192.82	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=0, ID=3113) [Reassembled in #3112]
3109	514.391829632	192.168.1.72	37.140.192.82	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=3113) [Reassembled in #3112]
3110	514.391834940	192.168.1.72	37.140.192.82	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=2960, ID=3113) [Reassembled in #3112]
3111	514.391856731	192.168.1.72	37.140.192.82	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=4440, ID=3113) [Reassembled in #3112]
3112	514.391861620	192.168.1.72	37.140.192.82	ICMP	122	Echo (ping) request id=0x1c72, seq=1/256, ttl=64 (no response found!)
3113	515.415513899	192.168.1.72	37.140.192.82	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=0, ID=334c) [Reassembled in #3117]
3114	515.415539223	192.168.1.72	37.140.192.82	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=334c) [Reassembled in #3117]
3115	515.415545369	192.168.1.72	37.140.192.82	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=2960, ID=334c) [Reassembled in #3117]
3116	515.415552563	192.168.1.72	37.140.192.82	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=4440, ID=334c) [Reassembled in #3117]
3117	515.415552563	192.168.1.72	37.140.192.82	ICMP	122	Echo (ping) request id=0x1c72, seq=2/512, ttl=64 (no response found!)

Ниже представлено содержимое кадра 3110. Мы видим, что пакет IPv4, который в нем лежит, имеет размер 1500 байт (с учетом заголовка), а размер кадра Ethernet равен 1514 байтам.

```
Frame 3110: 1514 bytes on wire (12112 bits), 1514 bytes captured (12112 bits) on interface wlp2s0, id 0
Ethernet II, Src: Chongqin_b1:1f:9d (5c:3a:45:b1:1f:9d), Dst: 2e:9d:3b:01:ff:7f (2e:9d:3b:01:ff:7f)
  Destination: 2e:9d:3b:01:ff:7f (2e:9d:3b:01:ff:7f)
  Source: Chongqin_b1:1f:9d (5c:3a:45:b1:1f:9d)
  Type: IPv4 (0x0800)
Internet Protocol Version 4, Src: 192.168.1.72, Dst: 37.140.192.82
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
  Total Length: 1500
  Identification: 0x3113 (12563)
  001. .... = Flags: 0x1, More fragments
    0... .... = Reserved bit: Not set
    .0.. .... = Don't fragment: Not set
    ..1. .... = More fragments: Set
    ...0 0001 0111 0010 = Fragment Offset: 2960
  Time to Live: 64
  Protocol: ICMP (1)
  Header Checksum: 0x7acd [validation disabled]
  [Header checksum status: Unverified]
  Source Address: 192.168.1.72
  Destination Address: 37.140.192.82
  [Reassembled IPv4 in frame: 3112]
```

Сам заголовок Ethernet II содержит MAC адреса — 12 байт и 2 байта, отвечающие за тип протокола выше (IPv4).

0000	2e 9d 3b 01 ff 7f 5c 3a 45 b1 1f 9d 08 00 45 00	..;... \: E.....E
0010	00 6c 31 13 02 e4 40 01 9e cb c0 a8 01 48 25 8c	l1...@... H%
0020	c0 52 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25	R..... !"#\$\$%
0030	26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35	&'()*+,-./012345
0040	36 37 38 39 3a 3b 3c 3d 3e 3f 40 41 42 43 44 45	6789:;<=>?@ABCDE
0050	46 47 48 49 4a 4b 4c 4d 4e 4f 50 51 52 53 54 55	FGHIJKLM NOPQRSTU
0060	56 57 58 59 5a 5b 5c 5d 5e 5f 60 61 62 63 64 65	VWXYZ[\] ^_`abcde
0070	66 67 68 69 6a 6b 6c 6d 6e 6f	fghijklm no

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

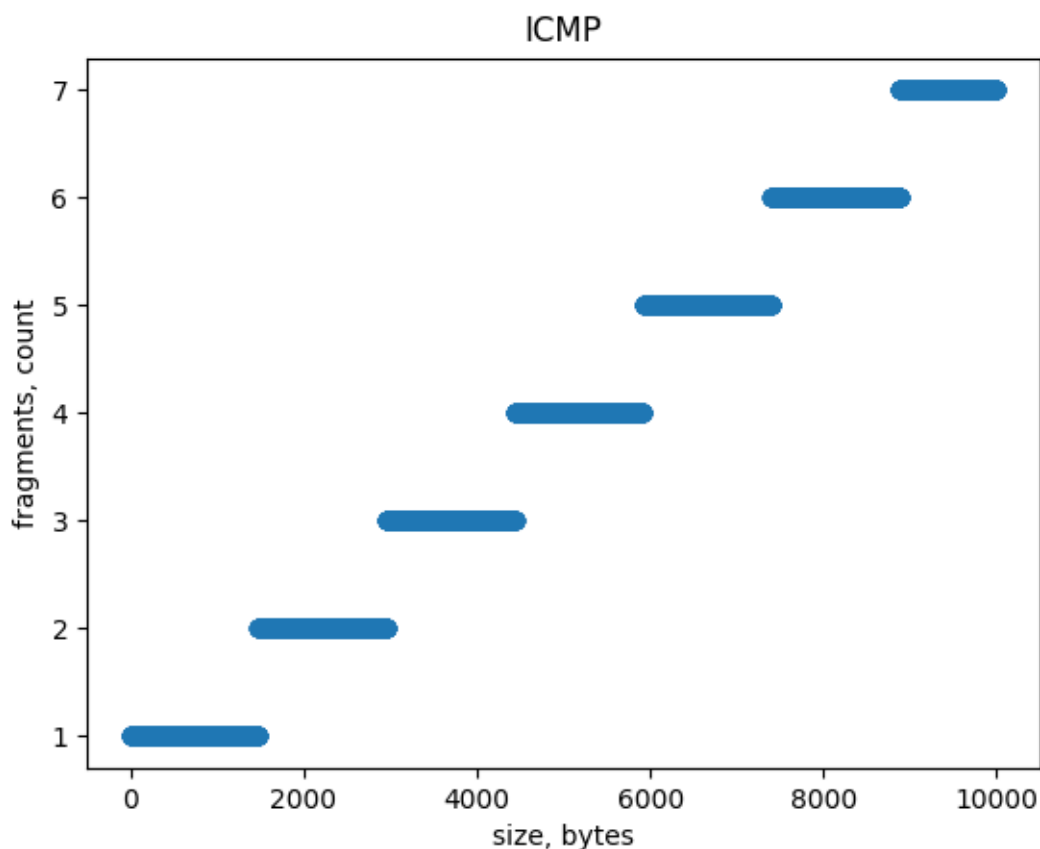
Флаг «More fragments» в пакете IPv4 отвечает за то, является ли фрагмент не последним.

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

Количество фрагментов прямопропорционально размеру передаваемого пакета. Так если передается пакет размера 6000 (6028) байт, то для его полной передачи с учетом потребуется 5 фрагментов, в связи с ограничением размера кадра

Ethernet (описано выше). Так в каждый пакет будет уместиться IPv4 максимум 1480 байт данных, потому что пакет IPv4 вкладывается в кадр Ethernet.

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



5. Как изменить поле TTL с помощью утилиты ping?

```
-t ttl  
ping only. Set the IP Time to Live.
```

6. Что содержится в поле данных ping-пакета?

Мусор в виде ASCII символов, если это так можно трактовать. Другими словами, однобайтовые слова, последовательно идущие друг за другом и увеличивающиеся на единицу.

0190	f6 f7 f8 f9 fa fb fc fd fe ff 00 01 02 03 04 05
01a0	06 07 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15
01b0	16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25 !"#%&
01c0	26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35	&'()*+,-./012345
01d0	36 37 38 39 3a 3b 3c 3d 3e 3f 40 41 42 43 44 45	6789:;<=>?@ABCDE
01e0	46 47 48 49 4a 4b 4c 4d 4e 4f 50 51 52 53 54 55	FGHIJKLM NOPQRSTU
01f0	56 57 58 59 5a 5b 5c 5d 5e 5f 60 61 62 63 64 65	VWXYZ[\]^_`abcde
0200	66 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75	fghijklm nopqrstu
0210	76 77 78 79 7a 7b 7c 7d 7e 7f 80 81 82 83 84 85	vwxyz{ } ~.....

2.2. Анализ трафика утилиты tracert (traceroute)

```
(machine-learning) nikit@host lab-4 % traceroute kulakov-pro.ru
traceroute to kulakov-pro.ru (37.140.192.82), 30 hops max, 60 byte packets
 1  192.168.1.197 (192.168.1.197)  6.090 ms  6.041 ms  6.014 ms
 2  * * *
 3  * * *
 4  * * *
 5  * * *
 6  * * *
 7  * * *
 8  188.170.161.57 (188.170.161.57)  56.102 ms  52.769 ms  55.991 ms
 9  188.170.161.56 (188.170.161.56)  52.693 ms  67.227 ms  63.935 ms
10  * * *
11  37.29.3.22 (37.29.3.22)  70.367 ms  70.339 ms  66.187 ms
12  185.140.148.19 (185.140.148.19)  66.153 ms  69.381 ms  185.140.148.31 (185.140.148.31)  44.77
13  ms
14  * * *
15  * * *
16  * * *
17  * * *
18  * * *
19  server51.hosting.reg.ru (37.140.192.82)  53.716 ms  58.197 ms  58.239 ms
```

В соответствии с документацией, звездочки означают, что либо маршрутизатор в течение какого-то времени не отвечает, либо временно перегружен, либо некоторые маршрутизаторы специально не реагируют на данные сообщения, так как им запретили. Таким образом, в некоторых случаях даже когда выставим огромное значение TTL, то все равно не достигнем конечного узла.

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

Ниже представлен ответ от маршрутизатора с типом time-limit-exceeds. В заголовке IP содержится 20 байт. В поле данных 32 байта

```
Internet Protocol Version 4, Src: 188.170.161.56, Dst: 192.168.1.72
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  Differentiated Services Field: 0x30 (DSCP: AF12, ECN: Not-ECT)
  Total Length: 96
  Identification: 0x5164 (20836)
  000. .... = Flags: 0x0
    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: 247
  Protocol: ICMP (1)
  Header Checksum: 0x5235 [validation disabled]
  [Header checksum status: Unverified]
  Source Address: 188.170.161.56
  Destination Address: 192.168.1.72
  Internet Control Message Protocol
  Data (32 bytes)
```

```
Internet Control Message Protocol
  Type: 11 (Time-to-live exceeded)
  Code: 0 (Time to live exceeded in transit)
  Checksum: 0x9cf7 [correct]
  [Checksum Status: Good]
  Unused: 00
  Length: 17
  [Length of original datagram: 68]
  Unused: 0000
  Internet Protocol Version 4, Src: 192.168.1.72, Dst: 37.140.192.82
  User Datagram Protocol, Src Port: 53009, Dst Port: 33459
    Source Port: 53009
    Destination Port: 33459
      [Expert Info (Chat/Sequence): Possible traceroute: hop #9, attempt #1]
      Length: 40
      Checksum: 0x1105 [unverified]
      [Checksum Status: Unverified]
      [Stream index: 29]
      UDP payload (32 bytes)
```

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.

Для определения пути tracer отправляет пакеты с TTL, который увеличивается. Таким образом, когда TTL сравнится в маршрутизаторе равным 0, то тот посылает ICMP сообщение с информацией о том, что TTL exceeds. Таким образом, по умолчанию сначала посылается пакет с TTL равным 1 (n-ое кол-во штук), затем с TTL равным 2 и т. д. до тех пор пока не достигнем конечного узла.

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

```

0040  82 9a 00 28 41 8e 40 41 42 43 44 45 46 47 48 49  ... (A @A BCDEFGHI
0050  4a 4b 4c 4d 4e 4f 50 51 52 53 54 55 56 57 58 59  JKLMNOPQ RSTUVWXY
0060  5a 5b 5c 5d 5e 5f                                Z[\]^_

```

В данном случае тоже возвращаются инкрементирующиеся слова, однако начиная с другого числа.

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

При отправке получал только 2 варианта ответов от ICMP: TTL exceeded — пакет достиг маршрутизатора, второй — порт недоступен. Как правило второе указывает на состояние перегрузки или проблему с конфигурацией процесса на хосте, создающем отчеты, значит, вероятно, маршрутизатор не ожидает таких запросов.

Internet Control Message Protocol	Internet Control Message Protocol
Type: 11 (Time-to-live exceeded)	Type: 3 (Destination unreachable)
Code: 0 (Time to live exceeded in transit)	Code: 3 (Port unreachable)
Checksum: 0x624a [correct]	Checksum: 0xa505 [correct]
[Checksum Status: Good]	[Checksum Status: Good]
Unused: 00000000	Unused: 00000000

Второй ответ получаем уже только на конечном узле:

```

19 server51.hosting.reg.ru (37.140.192.82) 56.858 ms 56.853 ms 66.507 ms
(machine-learning) nikitahost lab-4 % 

```

70 9.595787900	37.140.192.82	192.168.1.72	ICMP	70 Time-to-live exceeded (Time to live exceeded in transit)
74 9.616551074	185.140.148.31	192.168.1.72	ICMP	70 Time-to-live exceeded (Time to live exceeded in transit)
126 14.640943425	37.140.192.82	192.168.1.72	ICMP	102 Destination unreachable (Port unreachable)
127 14.640986728	37.140.192.82	192.168.1.72	ICMP	102 Destination unreachable (Port unreachable)
128 14.652822880	37.140.192.82	192.168.1.72	ICMP	102 Destination unreachable (Port unreachable)

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

Из документации WINDOWS:

Параметр -d с помощью команды tracert программа TRACERT не требуется выполнять поиск в DNS для каждого IP-адреса, так, что команда TRACERT отображает IP-адрес ближних интерфейсов маршрутизаторов.

В Linux этот флаг связан с дополнительной отладочной информацией от сокетов.

2.3. Анализ HTTP-трафика

В данном пункте описано только то, что удалось получить. Запросы от браузера видны только TLS и TCP.

106	10.3995/4484	64.233.162.105	192.168.1.72	TLSv1.3	293 Application Data
107	10.402781146	64.233.162.105	192.168.1.72	TLSv1.3	129 Application Data
108	10.402885699	192.168.1.72	64.233.162.105	TCP	66 42314 → 443 [ACK] Seq=1675 Ack=2667 Win=64128 Len=0 TSval=297259124 TSecr=3197852920
109	10.410391773	64.233.162.105	192.168.1.72	TLSv1.3	98 Application Data
110	10.413053820	64.233.162.105	192.168.1.72	TLSv1.3	97 Application Data
111	10.413654658	64.233.162.105	192.168.1.72	TLSv1.3	105 Application Data
112	10.413793085	192.168.1.72	64.233.162.105	TCP	66 42314 → 443 [ACK] Seq=1675 Ack=2769 Win=64128 Len=0 TSval=297259135 TSecr=3197852923
113	10.413842743	192.168.1.72	64.233.162.105	TLSv1.3	105 Application Data
114	10.463715752	64.233.162.105	192.168.1.72	TCP	66 443 → 42314 [ACK] Seq=2769 Ack=1714 Win=68352 Len=0 TSval=3197852984 TSecr=297259135
115	10.870089477	192.168.1.72	64.233.162.105	TLSv1.3	177 Application Data
116	10.902043231	64.233.162.105	192.168.1.72	TCP	66 443 → 42314 [ACK] Seq=2769 Ack=1825 Win=68352 Len=0 TSval=3197853423 TSecr=297259591
117	10.924676761	64.233.162.105	192.168.1.72	TLSv1.3	293 Application Data
118	10.927902979	64.233.162.105	192.168.1.72	TLSv1.3	127 Application Data
119	10.927903887	64.233.162.105	192.168.1.72	TLSv1.3	98 Application Data
120	10.927904865	64.233.162.105	192.168.1.72	TLSv1.3	97 Application Data
121	10.927906261	64.233.162.105	192.168.1.72	TLSv1.3	105 Application Data
122	10.928061940	192.168.1.72	64.233.162.105	TCP	66 42314 → 443 [ACK] Seq=1825 Ack=3159 Win=64128 Len=0 TSval=297259649 TSecr=3197853437
123	10.928113064	192.168.1.72	64.233.162.105	TLSv1.3	105 Application Data
124	10.967647622	64.233.162.105	192.168.1.72	TCP	66 443 → 42314 [ACK] Seq=3159 Ack=1864 Win=68352 Len=0 TSval=3197853488 TSecr=297259649
125	14.003264960	149.154.167.41	192.168.1.72	SSL	171 Continuation Data
126	14.043486625	192.168.1.72	149.154.167.41	TCP	66 34696 → 443 [ACK] Seq=1 Ack=106 Win=3982 Len=0 TSval=3551603239 TSecr=354598648

Рассмотрим какой-то из пакетов HTTP. Как можем увидеть, протокол HTTP является прикладным протоколом, и поэтому работает поверх протокола TCP.

735	94.672455563	192.168.1.72	149.154.167.41	HTTP	415 POST /api HTTP/1.1 (application/x-www-form-urlencoded)
736	94.672474490	192.168.1.72	149.154.167.50	HTTP	567 POST /api HTTP/1.1 (application/x-www-form-urlencoded)
744	94.760951189	149.154.167.41	192.168.1.72	HTTP	393 HTTP/1.1 200 OK
748	94.764160854	149.154.167.50	192.168.1.72	HTTP	300 HTTP/1.1 200 OK
811	95.140348682	192.168.1.72	149.154.167.41	HTTP	451 POST /api HTTP/1.1 (application/x-www-form-urlencoded)
817	95.151932906	192.168.1.72	149.154.167.50	HTTP	471 POST /api HTTP/1.1 (application/x-www-form-urlencoded)
821	95.398195600	149.154.167.41	192.168.1.72	HTTP	296 HTTP/1.1 200 OK
822	95.398196997	149.154.167.50	192.168.1.72	HTTP	373 HTTP/1.1 200 OK
1301	133.750665091	192.168.1.72	149.154.167.41	HTTP	407 GET /api HTTP/1.1
1336	134.098447923	192.168.1.72	2.23.167.179	OCSP	481 Request
1338	134.149943987	2.23.167.179	192.168.1.72	OCSP	955 Response

```
▶ Frame 1301: 407 bytes on wire (3256 bits), 407 bytes captured (3256 bits) on interface wlp2s0, id 0
▶ Ethernet II, Src: Chongqin_b1:1f:9d (5c:3a:45:b1:1f:9d), Dst: 2e:9d:3b:01:ff:7f (2e:9d:3b:01:ff:7f)
▶ Internet Protocol Version 4, Src: 192.168.1.72, Dst: 149.154.167.41
▶ Transmission Control Protocol, Src Port: 54054, Dst Port: 80, Seq: 1, Ack: 1, Len: 341
▼ Hypertext Transfer Protocol
  ▼ GET /api HTTP/1.1\r\n
    ▶ [Expert Info (Chat/Sequence): GET /api HTTP/1.1\r\n]
      Request Method: GET
      Request URI: /api
      Request Version: HTTP/1.1
      Host: 149.154.167.41\r\n
      User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/112.0\r\n
      Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8\r\n
      Accept-Language: en-US,en;q=0.5\r\n
      Accept-Encoding: gzip, deflate\r\n
      Connection: keep-alive\r\n
      Upgrade-Insecure-Requests: 1\r\n
      \r\n
      [Full request URI: http://149.154.167.41/api]
      [HTTP request 1/1]
```

Сообщение прокола представляется в виде:

- <Тип запроса (GET/POST)> <путь относительно корня>
<Протокол/версия>
- Затем с новой строки (переноса) идут значения вида ключ:значение
- После идет тело запроса, может отсутствовать


```

0040 d9 0a 47 45 54 20 2f 61 70 69 20 48 54 54 50 2f GET /api HTTP/
0050 31 2e 31 0d 0a 48 6f 73 74 3a 20 31 34 39 2e 31 1.1.0 Host: 149.1
0060 35 34 2e 31 36 37 2e 34 31 0d 0a 55 73 65 72 2d 54.167.4 1.0 User-
0070 41 67 65 6e 74 3a 20 4d 6f 7a 69 6c 6c 61 2f 35 Agent: Mozilla/5
0080 2e 30 20 28 58 31 31 3b 20 4c 69 6e 75 78 20 78 .0 (X11; Linux x
0090 38 36 5f 36 34 3b 20 72 76 3a 31 30 39 2e 30 29 86_64; rv:109.0)
00a0 20 47 65 63 6b 6f 2f 32 30 31 30 30 31 30 31 20 Gecko/2.0.0.101
00b0 46 69 72 65 66 6f 78 2f 31 31 32 2e 30 0d 0a 41 Firefox/112.0.0 A
00c0 63 63 65 70 74 3a 20 74 65 78 74 2f 68 74 6d 6c ccept: text/html
00d0 2c 61 70 70 6c 69 63 61 74 69 6f 6e 2f 78 68 74 ,application/xht
00e0 6d 6c 2b 78 6d 6c 2c 61 70 70 6c 69 63 61 74 69 ml+xml,application
00f0 6f 6e 2f 78 6d 6c 3b 71 3d 30 2e 39 2c 69 6d 61 on/xml;q=0.9,ima
0100 67 65 2f 61 76 69 66 2c 69 6d 61 67 65 2f 77 65 ge/avif,image/we
0110 62 70 2c 2a 2f 2a 3b 71 3d 30 2e 38 0d 0a 41 63 bp,*/*;q=0.8 Ac
0120 63 65 70 74 2d 4c 61 6e 67 75 61 67 65 3a 20 65 ccept-Language: e
0130 6e 2d 55 53 2c 65 6e 3b 71 3d 30 2e 35 0d 0a 41 n-US,en;q=0.5 A
0140 63 63 65 70 74 2d 45 6e 63 6f 64 69 6e 67 3a 20 ccept-Encoding:
0150 67 7a 69 70 2c 20 64 65 66 6c 61 74 65 0d 0a 43 gzip, deflate C
0160 6f 6e 6e 65 62 74 69 6f 6e 2a 20 6b 65 65 70 2d onnection: keep

```

Ответ выглядит в виде:

- <Протокол/версия> <Статус> <перенос>
- Заголовок вида ключ-значение
- Тело запроса

```

Hypertext Transfer Protocol
  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
    Connection: keep-alive\r\n
    Content-type: application/octet-stream\r\n
    Pragma: no-cache\r\n
    Cache-control: no-store\r\n
    Content-length: 160\r\n
    \r\n
    [HTTP response 1/1]
    [Time since request: 0.246264091 seconds]
    [Request in frame: 817]
    [Request URI: http://149.154.167.50:80/api]
    File Data: 160 bytes
  Data (160 bytes)
    Data: 00000000000000000110397a25385a648c000000632416055939d67561f335cdf095dbc6
    [Length: 160]

```

С помощью утилиты wget удалось послать запрос на <http://kulakov-pro.ru> и получить следующий ответ:

1298	168.472349064	192.168.1.72	37.140.192.82	HTTP	195 GET / HTTP/1.1
1300	168.555047194	37.140.192.82	192.168.1.72	HTTP	526 HTTP/1.1 301 Moved Permanently (text/html)
1385	179.790134072	192.168.1.72	37.140.192.82	HTTP	195 GET / HTTP/1.1
1387	179.851023014	37.140.192.82	192.168.1.72	HTTP	526 HTTP/1.1 301 Moved Permanently (text/html)

Нас перенаправляет на https версию страницы.

```
Hypertext Transfer Protocol
  HTTP/1.1 301 Moved Permanently\r\n
    [Expert Info (Chat/Sequence): HTTP/1.1 301 Moved Permanently\r\n]
    Response Version: HTTP/1.1
    Status Code: 301
    [Status Code Description: Moved Permanently]
    Response Phrase: Moved Permanently
  Server: nginx\r\n
  Date: Tue, 09 May 2023 12:33:23 GMT\r\n
  Content-Type: text/html; charset=iso-8859-1\r\n
  Transfer-Encoding: chunked\r\n
  Connection: keep-alive\r\n
  Location: https://kulakov-pro.ru/\r\n
  \r\n
  [HTTP response 1/1]
  [Time since request: 0.060888942 seconds]
  [Request in frame: 1385]
  [Request URI: http://kulakov-pro.ru/]
  HTTP chunked response
```

Тело запроса:

```
File Data: 231 bytes
  Line-based text data: text/html (7 lines)
  <!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">\n
  <html><head>\n
  <title>301 Moved Permanently</title>\n
  </head><body>\n
  <h1>Moved Permanently</h1>\n
  <p>The document has moved <a href="https://kulakov-pro.ru/">here</a>.</p>\n
  </body></html>\n
```

Подытожив, понимаем, что wireshark не отображает https запросы и ответы.

2.4. Анализ DNS-трафика

Отчищаем dns, перезапускаем сервис:

```
sudo /etc/init.d/dnsmasq restart
```

57	3.386735928	64.233.161.99	192.168.1.72	TCP	66	443 → 58372 [ACK] Seq=1603 Ack=392 Win=267 Len=0 TSval=909785213 TSecr=1484511040
58	3.596063554	44.240.24.56	192.168.1.72	TCP	66	[TCP ACKed unseen segment] 443 → 52368 [ACK] Seq=1 Ack=2 Win=187 Len=0 TSval=2929296495
59	3.693973158	192.168.1.72	192.168.1.197	DNS	74	Standard query 0x41d2 A kulakov-pro.ru
60	3.697342406	192.168.1.197	192.168.1.72	DNS	90	Standard query response 0x41d2 A kulakov-pro.ru A 37.140.192.82
61	3.697431175	192.168.1.72	192.168.1.197	DNS	74	Standard query 0xcddc AAAA kulakov-pro.ru
62	3.706612408	192.168.1.197	192.168.1.72	DNS	102	Standard query response 0xcddc AAAA kulakov-pro.ru AAAA 2a00:f940:2:2:1:1:0:51
63	3.702444401	192.168.1.72	192.168.1.197	DNS	74	Standard query 0x1d9d A kulakov-pro.ru
64	3.707127129	192.168.1.197	192.168.1.72	DNS	90	Standard query response 0x1d9d A kulakov-pro.ru A 37.140.192.82
65	3.707733218	192.168.1.72	37.140.192.82	TCP	74	53932 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=4079998375 TSecr=0 WS=
66	3.757221578	37.140.192.82	192.168.1.72	TCP	74	443 → 53932 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1360 SACK_PERM TSval=128117171 T
67	3.757265997	192.168.1.72	37.140.192.82	TCP	66	53932 → 443 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=4079998425 TSecr=128117171
68	3.760875711	192.168.1.72	37.140.192.82	TLSv1.3	583	Client Hello
69	3.814637283	37.140.192.82	192.168.1.72	TCP	66	443 → 53932 [ACK] Seq=1 Ack=518 Win=30080 Len=0 TSval=128117229 TSecr=4079998428
70	3.827483670	37.140.192.82	192.168.1.72	TLSv1.3	1414	Server Hello, Change Cipher Spec, Application Data
71	3.827516565	192.168.1.72	37.140.192.82	TCP	66	53932 → 443 [ACK] Seq=518 Ack=1349 Win=64128 Len=0 TSval=4079998495 TSecr=128117229
72	3.830740472	37.140.192.82	192.168.1.72	TCP	2814	443 → 53932 [PSH, ACK] Seq=1349 Ack=518 Win=30080 Len=2748 TSval=128117229 TSecr=407999
73	3.830741938	37.140.192.82	192.168.1.72	TLSv1.3	572	Application Data, Application Data, Application Data

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

Для того, чтобы получить IP адрес сайта отправляется запрос на DNS сервер, который возвращает IP адрес уже сайта. На рисунке ниже мы можем видеть, что получается как IPv4, так и IPv6 адреса необходимой страницы. Адресом назначения указан адрес маршрутизатора (модема телефона).

2. Какие бывают типы DNS-запросов?

- Прямой — запрос на преобразование имени (символьного адреса) хоста в его IP-адрес.
- Обратный — запрос на преобразование адреса хоста в его имя.
- Рекурсивный - DNS-сервер опрашивает серверы (в порядке убывания уровня зон в имени), пока не найдёт ответ или не обнаружит, что домен не существует.
- Нерекурсивный (итеративный) - DNS-сервер либо возвращает данные о зоне, за которую он ответственен, либо возвращает ошибку.

Также выделяют запросы типа:

- A — получение IPv4
- AAAA — получение IPv6
- CNAME — получение канонического имени
- MX — получение информации о почтовых серверах, ответственных за обработку почту для данного домена
- NS (Name Server) — вернуть список DNS серверов, ответственных за данный домен
- PTR - обратная DNS-запись или запись указателя связывает IP-адрес хоста с его каноническим именем.

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

- Если изображения хранятся на отдельном сервере или поддомене
- Используется CDN (Content Delivery Network), где ресурсы могут храниться на разных серверах

2.5. Анализ ARP-трафика

Очищаем таблицу MAC адресов:

```
nikit@host lab-4 % arp -a
? (192.168.1.197) at 2e:9d:3b:01:ff:7f [ether] on wlp2s0
nikit@host lab-4 % arp --delete 192.168.1.197
```

Как можно заметить ниже ARP запросы посылаются как от маршрутизатора (192.168.1.197), так и от компьютера (192.168.1.72). Broadcast означает, что данный запрос предназначен всем узлам в данной локальной сети.

3491	396.604475809	2e:9d:3b:01:ff:7f	Chongqin_b1:1f:9d	ARP	42	Who has 192.168.1.72? Tell 192.168.1.197
3492	396.604503397	Chongqin_b1:1f:9d	2e:9d:3b:01:ff:7f	ARP	42	192.168.1.72 is at 5c:3a:45:b1:1f:9d
3506	407.484092450	Chongqin_b1:1f:9d	Broadcast	ARP	42	Who has 192.168.1.197? Tell 192.168.1.72
3507	407.488023304	2e:9d:3b:01:ff:7f	Chongqin_b1:1f:9d	ARP	42	192.168.1.197 is at 2e:9d:3b:01:ff:7f

```
▶ Frame 3926: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface wlp2s0, id 0
▶ Ethernet II, Src: 2e:9d:3b:01:ff:7f (2e:9d:3b:01:ff:7f), Dst: Chongqin_b1:1f:9d (5c:3a:45:b1:1f:9d)
▼ Address Resolution Protocol (request)
  Hardware type: Ethernet (1)
  Protocol type: IPv4 (0x0800)
  Hardware size: 6
  Protocol size: 4
  Opcode: request (1)
  Sender MAC address: 2e:9d:3b:01:ff:7f (2e:9d:3b:01:ff:7f)
  Sender IP address: 192.168.1.197
  Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
  Target IP address: 192.168.1.72
```

```
▶ Frame 3927: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface wlp2s0, id 0
▶ Ethernet II, Src: Chongqin_b1:1f:9d (5c:3a:45:b1:1f:9d), Dst: 2e:9d:3b:01:ff:7f (2e:9d:3b:01:ff:7f)
▼ Address Resolution Protocol (reply)
  Hardware type: Ethernet (1)
  Protocol type: IPv4 (0x0800)
  Hardware size: 6
  Protocol size: 4
  Opcode: reply (2)
  Sender MAC address: Chongqin_b1:1f:9d (5c:3a:45:b1:1f:9d)
  Sender IP address: 192.168.1.72
  Target MAC address: 2e:9d:3b:01:ff:7f (2e:9d:3b:01:ff:7f)
  Target IP address: 192.168.1.197
```

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

При запросе присутствуют MAC адрес отправителя, при ответе MAC адрес отправителя и получателя. MAC адрес — физический адрес устройства. Ниже показано, что он является глобальным индивидуальным для отправителя — компьютера и локальным индивидуальным для модема (телефона, раздающего интернет) (заданы соответствующие биты MAC адреса):

```
▼ Ethernet II, Src: Chongqin_b1:1f:9d (5c:3a:45:b1:1f:9d), Dst: 2e:9d:3b:01:ff:7f (2e:9d:3b:01:ff:7f)
  ▼ Destination: 2e:9d:3b:01:ff:7f (2e:9d:3b:01:ff:7f)
    Address: 2e:9d:3b:01:ff:7f (2e:9d:3b:01:ff:7f)
    ....1. .... = LG bit: Locally administered address (this is NOT the factory default)
    ....0. .... = IG bit: Individual address (unicast)
  ▼ Source: Chongqin_b1:1f:9d (5c:3a:45:b1:1f:9d)
    Address: Chongqin_b1:1f:9d (5c:3a:45:b1:1f:9d)
    ....0. .... = LG bit: Globally unique address (factory default)
    ....0. .... = IG bit: Individual address (unicast)
  Type: ARP (0x0806)
  Address Resolution Protocol (reply)
```

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

Те же самые, что и в ARP запросе. Теперь по ним определяется с какого компьютера на какой маршрутизатор (модем) отправить/получить сообщение.

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

Скорее всего для определения коллизий со стороны получателя запроса, который по своей таблице ARP сможет это определить, однако конкретной информации по этому найти не удалось. ARP же протокол канального уровня, поэтому для понимания куда послать ответ ему IP адреса не нужны, так как с запросом был послан MAC адрес источника.

2.6. Анализ трафика утилиты nslookup

Утилита nslookup предназначена для получения различной информации с DNS серверов.

Первая команда, которая была послана для получения dns имени по ip адресу. Тип запроса — PTR, то есть получаем по известному ip адресу dns имя. Ниже представлено содержимое dns сообщения отправителя и получателя.

```
nikit@host lab-4 % nslookup 37.140.192.82
82.192.140.37.in-addr.arpa      name = server51.hosting.reg.ru.
Authoritative answers can be found from:
```

```

Domain Name System (query)
Transaction ID: 0x3947
Flags: 0x0100 Standard query
Questions: 1
Answer RRs: 0
Authority RRs: 0
Additional RRs: 0
Queries
  82.192.140.37.in-addr.arpa: type PTR, class IN
    Name: 82.192.140.37.in-addr.arpa
    [Name Length: 26]
    [Label Count: 6]
    Type: PTR (domain name PoinTeR) (12)
    Class: IN (0x0001)
    [Response: Truncated]

```

```

Queries
  82.192.140.37.in-addr.arpa: type PTR, class IN
    Name: 82.192.140.37.in-addr.arpa
    [Name Length: 26]
    [Label Count: 6]
    Type: PTR (domain name PoinTeR) (12)
    Class: IN (0x0001)
Answers
  82.192.140.37.in-addr.arpa: type PTR, class IN, server51.hosting.reg.ru
    Name: 82.192.140.37.in-addr.arpa
    Type: PTR (domain name PoinTeR) (12)
    Class: IN (0x0001)
    Time to live: 19916 (5 hours, 31 minutes, 56 seconds)
    Data length: 25
    Domain Name: server51.hosting.reg.ru

```

Данный запрос — запрос типа NS, то есть для получения DNS-серверов, которые отвечают за разрешение ip адреса и dns имени для данного доменного имени. Таким образом, определили, что авторитативный ответ может быть получен от DNS-серверов ns1.hosting.reg.ru и ns2.hosting.reg.ru.

```

nikit@host lab-4 % nslookup -type=ns kulakov-pro.ru
Server:          192.168.1.197
Address:         192.168.1.197#53

Non-authoritative answer:
KULAKOV-PRO.ru  nameserver = ns1.hosting.reg.ru.
KULAKOV-PRO.ru  nameserver = ns2.hosting.reg.ru.

Authoritative answers can be found from:
ns1.hosting.reg.ru    internet address = 31.31.194.245
ns1.hosting.reg.ru    internet address = 31.31.194.251
ns1.hosting.reg.ru    internet address = 31.31.196.37
ns1.hosting.reg.ru    internet address = 31.31.196.52
ns1.hosting.reg.ru    internet address = 31.31.196.61
ns1.hosting.reg.ru    internet address = 31.31.196.180
ns1.hosting.reg.ru    internet address = 31.31.198.177
ns1.hosting.reg.ru    internet address = 37.140.192.20
ns1.hosting.reg.ru    internet address = 37.140.192.93
ns1.hosting.reg.ru    internet address = 37.140.193.121
ns1.hosting.reg.ru    internet address = 37.140.196.144
ns1.hosting.reg.ru    internet address = 194.58.91.38
ns1.hosting.reg.ru    internet address = 194.67.73.6
ns1.hosting.reg.ru    internet address = 194.67.73.9

```

```

Domain Name System (query)
  Transaction ID: 0x6c40
  Flags: 0x0100 Standard query
  Questions: 1
  Answer RRs: 0
  Authority RRs: 0
  Additional RRs: 0
  Queries
    kulakov-pro.ru: type NS, class IN
      Name: kulakov-pro.ru
      [Name Length: 14]
      [Label Count: 2]
      Type: NS (authoritative Name Server) (2)
      Class: IN (0x0001)
[Response In: 61]

```

```

Domain Name System (response)
  Transaction ID: 0x6c40
  Flags: 0x8180 Standard query response, No error
  Questions: 1
  Answer RRs: 2
  Authority RRs: 0
  Additional RRs: 14
  Queries
    kulakov-pro.ru: type NS, class IN
      Name: kulakov-pro.ru
      [Name Length: 14]
      [Label Count: 2]
      Type: NS (authoritative Name Server) (2)
      Class: IN (0x0001)
  Answers
    KULAKOV-PRO.ru: type NS, class IN, ns ns1.hosting.reg.ru
      Name: KULAKOV-PRO.ru
      Type: NS (authoritative Name Server) (2)
      Class: IN (0x0001)
      Time to live: 1591 (26 minutes, 31 seconds)
      Data length: 20
      Name Server: ns1.hosting.reg.ru
    KULAKOV-PRO.ru: type NS, class IN, ns ns2.hosting.reg.ru
      Name: KULAKOV-PRO.ru
      Type: NS (authoritative Name Server) (2)
      Class: IN (0x0001)
      Time to live: 1591 (26 minutes, 31 seconds)
      Data length: 6
      Name Server: ns2.hosting.reg.ru
  Additional records
    ns1.hosting.reg.ru: type A, class IN, addr 31.31.194.245
      Name: ns1.hosting.reg.ru
      Type: A (Host Address) (1)
      Class: IN (0x0001)

```

36	3.383502...	192.168.1.72	192.168.1.197	DNS	86 Standard query 0x3947 PTR 82.192.140.37.in-addr.arpa
37	3.456506...	192.168.1.197	192.168.1.72	DNS	123 Standard query response 0x3947 PTR 82.192.140.37.in-addr.arpa PTR server51.hosting.reg.ru
60	11.33486...	192.168.1.72	192.168.1.197	DNS	74 Standard query 0x6c40 NS kulakov-pro.ru
61	11.37140...	192.168.1.197	192.168.1.72	DNS	360 Standard query response 0x6c40 NS kulakov-pro.ru NS ns1.hosting.reg.ru NS ns2.hosting.reg.ru A 31.31.194.245

Неавторитетная запись означает, что соотношение было получено нет от dns-сервера, который отвечает за данный сектор доменных имен. Так, например, если мы попросим авторитетный dns-сервер предоставить эту информацию, то этого предупреждения не будет.

```

nikit@host lab-4 % nslookup kulakov-pro.ru ns1.hosting.reg.ru
Server:      ns1.hosting.reg.ru
Address:     31.31.194.245#53

Name:   kulakov-pro.ru
Address: 37.140.192.82
Name:   kulakov-pro.ru
Address: 2a00:f940:2:2:1:1:0:51

```


В поле ANSWERS содержатся соответствующие записи на запрос. Одному запросу может соответствовать несколько записей ответа, как во втором примере. Поля соответствуют типу запроса.

2.7. Анализ FTP-трафика

Для получения чего-либо по протоколу ftp был осуществлен запрос на gitlab.se.ifmo.ru.

```
nikit@host lab-4 % wget ftp://gitlab.se.ifmo.ru
--2023-05-09 18:50:28--  ftp://gitlab.se.ifmo.ru/
=> '.listing'
Resolving gitlab.se.ifmo.ru... 77.234.214.82
Connecting to gitlab.se.ifmo.ru|77.234.214.82|:21... connected.
Logging in as anonymous ... Logged in!
==> SYST ... done.      ==> PWD ... done.
==> TYPE I ... done.    ==> CWD not needed.
==> PASV ... done.      ==> LIST ... done.

.listing                  [ <=>                  ]      242  --.-KB/s    in 0.01s

2023-05-09 18:50:29 (18.8 KB/s) - '.listing' saved [242]

Removed '.listing'.
Wrote HTML-ized index to 'index.html' [388].
```

Соответствующие ему сообщения в wireshark:

31...	69.68436...	77.234.214.82	192.168.1.72	FTP	110 Response: 220 ProFTPD 1.3.5a Server (FTP Server) [77.234.214.82]
31...	69.68453...	192.168.1.72	77.234.214.82	FTP	70 Request: USER anonymous
31...	69.72298...	77.234.214.82	192.168.1.72	FTP	129 Response: 331 Anonymous login ok, send your complete email address as your password
31...	69.72331...	192.168.1.72	77.234.214.82	FTP	67 Request: PASS -wget@
31...	69.76466...	77.234.214.82	192.168.1.72	FTP	104 Response: 230 Anonymous access granted, restrictions apply
31...	69.76487...	192.168.1.72	77.234.214.82	FTP	60 Request: SYST
31...	69.80437...	77.234.214.82	192.168.1.72	FTP	73 Response: 215 UNIX Type: L8
31...	69.80462...	192.168.1.72	77.234.214.82	FTP	59 Request: PWD
31...	69.85152...	77.234.214.82	192.168.1.72	FTP	88 Response: 257 "/" is the current directory
31...	69.85174...	192.168.1.72	77.234.214.82	FTP	62 Request: TYPE I
31...	69.88320...	77.234.214.82	192.168.1.72	FTP	73 Response: 200 Type set to I
31...	69.88339...	192.168.1.72	77.234.214.82	FTP	60 Request: PASV
31...	69.92342...	77.234.214.82	192.168.1.72	FTP	105 Response: 227 Entering Passive Mode (77,234,214,82,202,21).
31...	69.96749...	192.168.1.72	77.234.214.82	FTP	63 Request: LIST -a
31...	70.01302...	77.234.214.82	192.168.1.72	FTP	109 Response: 150 Opening BINARY mode data connection for file list
31...	70.02573...	77.234.214.82	192.168.1.72	FTP	308 FTP Data: 242 bytes (PASV) (LIST -a)
31...	70.07554...	77.234.214.82	192.168.1.72	FTP	77 Response: 226 Transfer complete

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

Для данного запроса количество байт данных соответствует 242. Ниже представлено подтверждение:

```

> Frame 3177: 308 bytes on wire (2464 bits), 308 bytes captured (2464 bits) on interface wlp2s0, i
> Ethernet II, Src: 2e:9d:3b:01:ff:7f (2e:9d:3b:01:ff:7f), Dst: Chongqin_b1:1f:9d (5c:3a:45:b1:1f:
> Internet Protocol Version 4, Src: 77.234.214.82, Dst: 192.168.1.72
> Transmission Control Protocol, Src Port: 51733, Dst Port: 46330, Seq: 1, Ack: 1, Len: 242
FTP Data (242 bytes data)
[Setup frame: 3168]
[Setup method: PASV]
[Command: LIST -a]
[Current working directory: /]
- Line-based text data (4 lines)
  drwxr-xr-x  4 root    wheel      4 Dec  5  2016  .\r\n
  drwxr-xr-x  4 root    wheel      4 Dec  5  2016  ..\r\n
  d-wrxw---  2 uploader ftp       2 Mar 23  2020 incoming\r\n
  drwxr-xr-x  4 root    wheel      4 May 16  2018 pub\r\n

```

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

Для протокола FTP стандартный порт для управления — 21. На моем компьютере он был выставлен в 49300.

```

- Transmission Control Protocol, Src Port: 49300, Dst Port: 21, Seq: 49, Ack: 254, Len: 6
  Source Port: 49300
  Destination Port: 21
  [Stream index: 52]
  [Conversation completeness: Complete, WITH_DATA (31)]
  [TCP Segment Len: 6]
  Sequence Number: 49      (relative sequence number)
  Sequence Number (raw): 2225053986
  [Next Sequence Number: 55      (relative sequence number)]
  Acknowledgment Number: 254      (relative ack number)
  Acknowledgment number (raw): 2213448416
  0101 .... = Header Length: 20 bytes (5)
  Flags: 0x018 (PSH, ACK)

```

FTP-data использует как правило случайный порт. В сообщении N31 пассивным портом был выставлен 51733, этот порт затем использовался сервером для отправки данных компьютеру. На стороне компьютера был выставлен порт 46330.

```

TCP payload (31 bytes)
- File Transfer Protocol (FTP)
  - 227 Entering Passive Mode (77,234,214,82,202,21).\r\n
    Response code: Entering Passive Mode (227)
    Response arg: Entering Passive Mode (77,234,214,82,202,21).
    Passive IP address: 77.234.214.82
    Passive port: 51733
  [Current working directory: /]

```

3. Чем отличаются пакеты FTP от FTP-DATA?

FTP используется для управления каналом: аутентификации, навигации и общего управления передачей файлов.

FTP-DATA - отвечает за передачу фактических данных файла между клиентом и сервером.

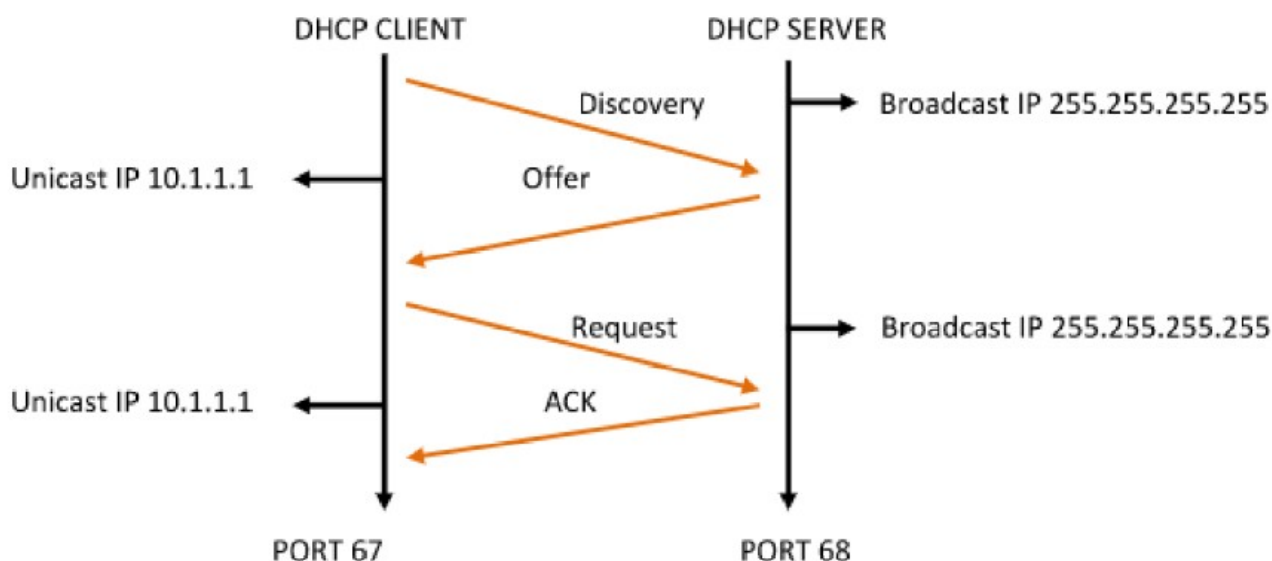
Это можно увидеть по описанию сообщений в Wireshark.

2.8. Анализ DHCP-трафика

Для того, чтобы стриггерить посылку DHCP запросов воспользуемся командами ниже:

```
nikit@host lab-4 % sudo nmap --script broadcast-dhcp-discover
Starting Nmap 7.93 ( https://nmap.org ) at 2023-05-09 19:11 MSK
Pre-scan script results:
  broadcast-dhcp-discover:
    Response 1 of 1:
      Interface: wlp2s0
      IP Offered: 192.168.1.241
      DHCP Message Type: DHCPOFFER
      Server Identifier: 192.168.1.197
      IP Address Lease Time: 59m59s
      Renewal Time Value: 29m59s
      Rebinding Time Value: 52m29s
      Subnet Mask: 255.255.255.0
      Broadcast Address: 192.168.1.255
      Router: 192.168.1.197
      Domain Name Server: 192.168.1.197
      Vendor Specific Information: ANDROID_METERED
WARNING: No targets were specified, so 0 hosts scanned.
Nmap done: 0 IP addresses (0 hosts up) scanned in 10.23 seconds
nikit@host lab-4 % sudo dhcpcd -T wlp2s0
dhcpcd=9.4.1 starting
```

Благодаря первой команде инициируется discover + offer, благодаря второй — request + ack. То есть компьютер посылает первый запрос, ему от модема (телефона) посылается ответ.



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

DHCP-discover — запрос поиска DHCP-сервера. DHCP-request — запрос на присвоение запрошенного IP адреса (в поле данных указывается).

```
magic cookie: DHCP
  Option: (50) Requested IP Address (192.168.1.72)
    Length: 4
    Requested IP Address: 192.168.1.72
  Option: (53) DHCP Message Type (Request)
    Length: 1
    DHCP: Request (3)
```

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

1) DHCP-discover — инициируется клиентом. MAC адрес отправителя — компьютер, IP адрес — 0.0.0.0 (не задан); MAC адрес получателя — broadcast, IP адрес — 255.255.255.255 (limited broadcast — все устройства в локальной сети).

2) DHCP-offer — инициируется сервером. MAC адрес и IP адрес отправителя — сервера; MAC адрес получателя — broadcast, IP адрес получателя — 255.255.255.255.

3) DHCP-request — инициируется клиентом. MAC адрес отправителя — компьютер, IP адрес — 0.0.0.0 (не задан); MAC адрес получателя — broadcast, IP адрес — 255.255.255.255 (limited broadcast)

4) DHCP-ack — инициируется сервером. MAC адрес и IP адрес отправителя — сервера; MAC адрес получателя — компьютер, IP адрес — запрошенный в DHCP-request.

3. Каков IP-адрес DHCP-сервера?

192.168.1.197

```

  ▸ User Datagram Protocol, Src Port: 67, Dst Port: 68
  ▸ Dynamic Host Configuration Protocol (Offer)
    Message type: Boot Reply (2)
    Hardware type: Ethernet (0x01)
    Hardware address length: 6
    Hops: 0
    Transaction ID: 0x5cfcdd90
    Seconds elapsed: 0
  ▸ Bootp flags: 0x8000, Broadcast flag (Broadcast)
    Client IP address: 0.0.0.0
    Your (client) IP address: 192.168.1.241
    Next server IP address: 192.168.1.197

```

4. Что произойдёт, если очистить использованный фильтр «bootp» (dhcp)?

Поскольку это был единственный фильтр, то будут просто отображаться все запросы-ответы.

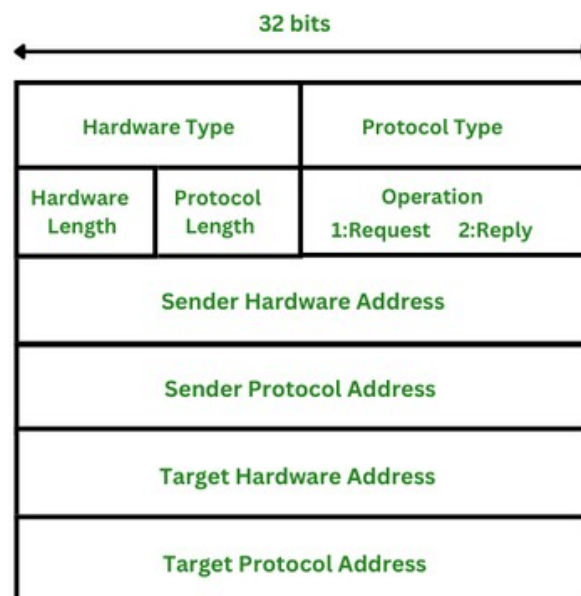
2.9. Структуры наблюдаемых пакетов заголовков

2.9.1. Ethernet II

Адрес получателя	Адрес отправителя	Тип	Данные	CRC
6	6	2	46 ... 1500	4

Источник : [intuit](http://intuit.ru)

2.9.2. ARP



Источник: [geeksforgeeks](https://www.geeksforgeeks.com/)

2.9.3. IPv4

Биты	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
	Номер версии	Длина заголовка	Тип сервиса (DS-байт)												Общая длина (байт)																	
			PR	D	T	R	ECN																									
Идентификатор пакета															Флаги			Смещение фрагмента (кратно 8 байтам)														
															-	DF	MF															
Время жизни (TTL – Time To Live)									Протокол (6 - TCP, 17 - UDP, 1 - ICMP)							Контрольная сумма заголовка (в дополнительном коде)																
IP-адрес источника																																
IP-адрес назначения																																
Параметры																								Наполнение								

Источник: Презентация №2 лекции по Компьютерным сетям, Алиев Тауфик Измайлович, 2023.

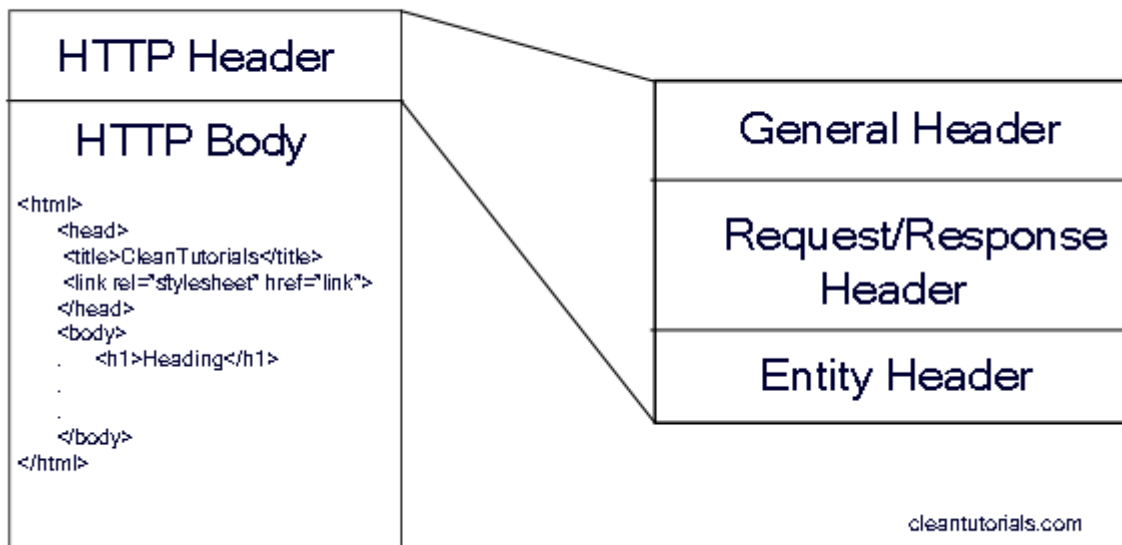
2.9.4. ICMP

Биты	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
Тип (Type)									Код (Kod)								Контрольная сумма (Check Sum)															
Служебная информация (зависит от типа и кода)																																

Источник: Презентация №2 лекции по Компьютерным сетям, Алиев Тауфик Измайлович, 2023.

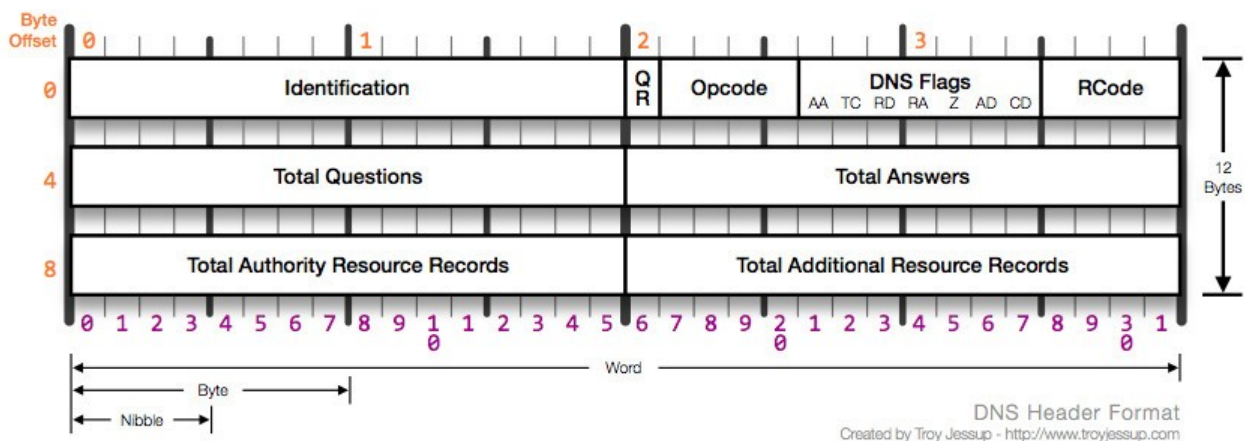
2.9.5. HTTP

HTTP Request/response



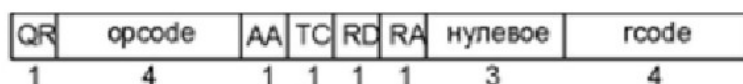
2.9.6. DNS

DNS Header



Identification — позволяет клиенту определить, на какой запрос пришел отклик.

Флаги:



где,

QR - тип сообщения

opcode - код операции

AA - авторитетный ответ

TC - "обрезано"

RD - "требуется рекурсия"

RA - "рекурсия возможна"

rcode - код возврата

Следующие четыре 16-битных поля указывают на количество пунктов в четырех полях переменной длины, которые завершают запись. В запросе number of questions обычно равно 1, а остальные три счетчика равны 0. В отклике number of answers по меньшей мере равно 1, а оставшиеся два счетчика могут быть как нулевыми, так и ненулевыми.

2.9.7. FTP

Зависит от текущего режима работы (активный, пассивный). Подробнее здесь:

<https://www.rhyshaden.com/ftp.htm>

2.9.8. DHCP

0	7	8	15	16	23	24	31
Operation code		Hardware address type		Hardware address length		Hops	
Client ID							
Start time				Flags			
Client address							
Offered address							
Server address							
Relay agent address							
Client hardware address							
Server name							
File name							
Options							

Источник: [wikimedia commons](https://commons.wikimedia.org/wiki/File:DHCP_packet_structure.png)

- Operation code - тип DHCP-сообщения. Если значение 0×01 – запрос к серверу, иначе — оно является ответом DHCP-сервера.
- Hardware Type - тип адреса на канальном уровне. DHCP может работать поверх различных протоколов на канальном уровне, поэтому нужно указывать на каком именно.
- Hardware Length - длина аппаратного адреса в байтах.
- Hops - количество промежуточных маршрутизаторов, которые находятся на пути между клиентом и сервером.
- Transaction ID – идентификатор процесса получения IP-адреса.

- Seconds Elapsed - время в секундах с момента начала процесса получения IP.
- Flags - поле для флагов и параметров протокола.
- Client IP Address - IP-адрес клиента. Не пусто, если у клиента уже есть IP и он хочет продлить время аренды IP-адреса.
- Your ID Address – предложенный DHCP-сервером IP клиенту.
- Server IP Address - IP-адрес сервера.
- Client Hardware Address – MAC клиента.
- Server Hostname – доменное имя сервера (если присутствует).
- Boot File - служит указателем для бездисковых рабочих станций о имени файла инициализации на сервере.
- Options – информация для динамической конфигурации хоста.

3. Выводы

В ходе выполнения лабораторной работы с помощью программы Wireshark убедился в том, каким образом представляются пакеты, как они обрабатываются при прохождении от соседних уровней или одного и того же уровня, как в пакете IPv4 лежит ICMP пакет, который принадлежит как и IP сетевому уровню. В целом, подтвердились те знания, которые были получены на лекции.