Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики

Кафедра информатики и прикладной математики

Сети ЭВМ и Телекоммуникации

Лабораторная работа 2

Старался: Шкаруба Н.Е.
Группа: P3318

2017

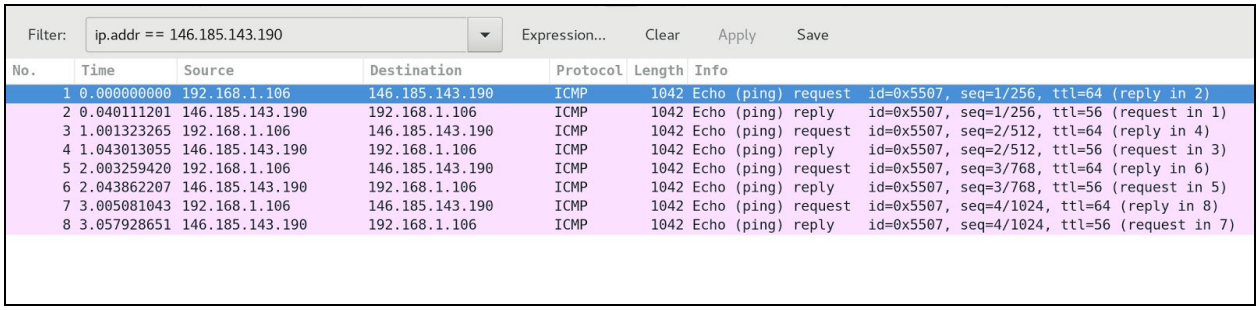
Цель работы:

Научиться работать с программами ping, wireshark

1. Ping:

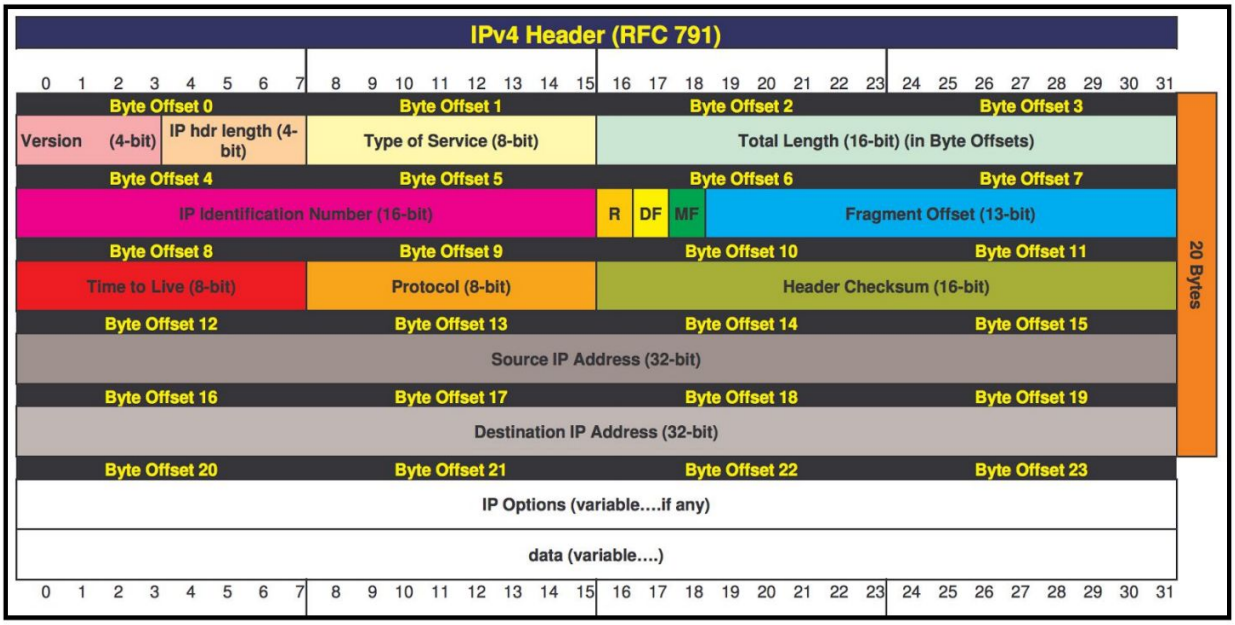
```
[sigma@magma main ]$ ping jovian -c 4 -s 1000
PING Jovian (146.185.143.190) 1000(1028) bytes of data.
1008 bytes from Jovian (146.185.143.190): icmp_seq=1 ttl=56 time=40.1 ms
1008 bytes from Jovian (146.185.143.190): icmp_seq=2 ttl=56 time=41.7 ms
1008 bytes from Jovian (146.185.143.190): icmp_seq=3 ttl=56 time=40.6 ms
1008 bytes from Jovian (146.185.143.190): icmp_seq=4 ttl=56 time=52.8 ms

--- Jovian ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 40.126/43.836/52.868/5.246 ms
```

Исполнение ping

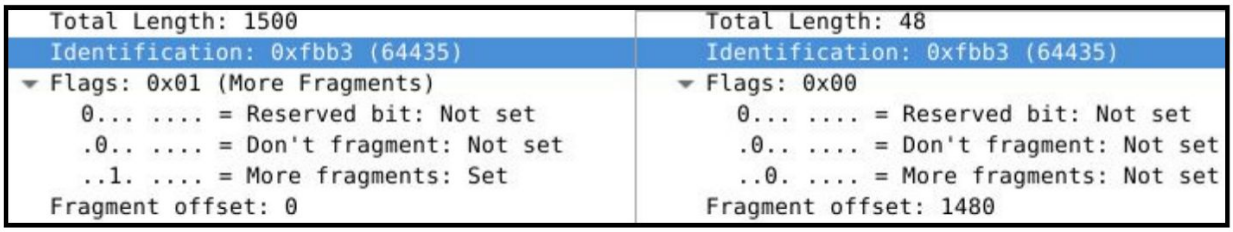


Окно wireshark после исполнения

1. **Имеет ли место фрагментация исходного пакета, какое поле на это указывает?**
   Фрагментация исходного пакета имеет место быть при превышении его размера MTU (обычно 1500 байт) без учёта Ethernet-фрейма. На фрагментацию указывает поле MF в заголовке IPv4, установленное у всех фрагментов пакета, кроме последнего.
   ask.wireshark.org/questions/41152/how-to-check-if-fragmentation-is-happening



Структура IPv4 заголовка



Снимок нескольких пакетов.

2. **Какая информация указывает, является ли фрагмент пакета последним или промежуточным?**
   На это указывает значение поля Fragment Offset в заголовке IPv4, показывающее смещение текущего фрагмента относительно всего пакета

3. **Чему равно количество фрагментов при передаче ping-пакетов?**
   Количество фрагментов вычисляется по формуле:
   (Packet Length / MTU) + (Packet Length % MTU ? 1 : 0)

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

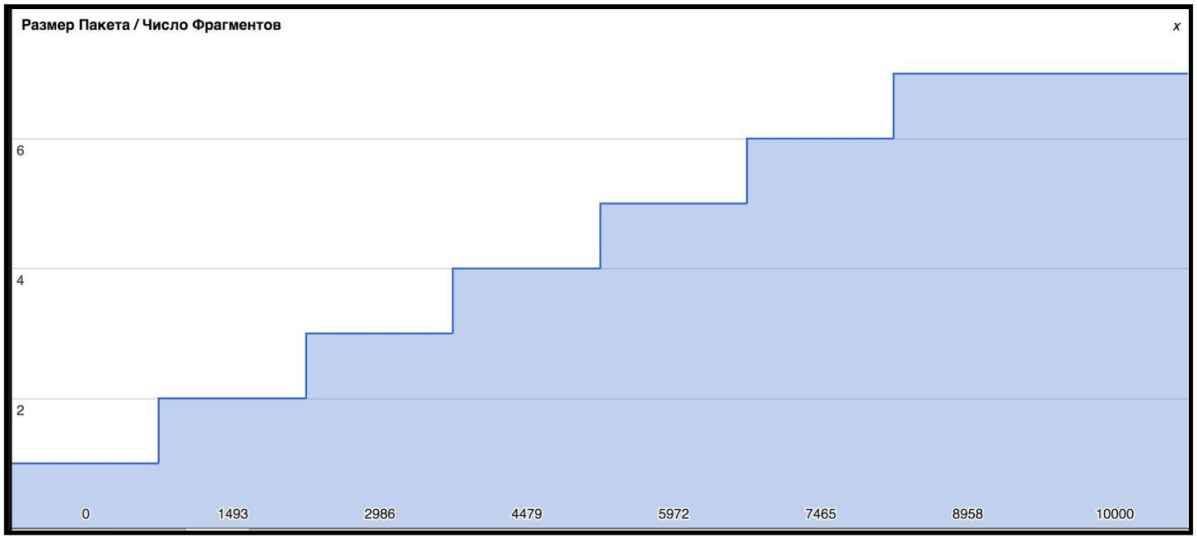


График зависимости размера пакетов от кол-ва его фрагментов.

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

```
$ ping -m [ttl] ...
```

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

Зависит от реализации программы. У меня хранится последовательность увеличивающаяся на 1 байт

```
0000  2c 56 dc 41 c1 cc 98 01  a7 9f b5 9f 08 00 45 00   ,V.A.... ......E.
0010  04 04 74 14 40 00 40 01  de 5a c0 a8 01 6a 92 b9   ..t.@.@. .Z...j..
0020  8f be 08 00 df be 55 07  00 04 3f a2 18 59 00 00   ......U. ..?..Y..
0030  00 00 11 f7 00 00 00 00  00 00 10 11 12 13 14 15   ........ ........
0040  16 17 18 19 1a 1b 1c 1d  1e 1f 20 21 22 23 24 25   ........ .. !"#$%
0050  26 27 28 29 2a 2b 2c 2d  2e 2f 30 31 32 33 34 35   &'()*+,- ./012345
0060  36 37 38 39 3a 3b 3c 3d  3e 3f 40 41 42 43 44 45   6789:;<= >?@ABCDE
0070  46 47 48 49 4a 4b 4c 4d  4e 4f 50 51 52 53 54 55   FGHIJKLM NOPQRSTU
0080  56 57 58 59 5a 5b 5c 5d  5e 5f 60 61 62 63 64 65   VWXYZ[\] ^_`abcde
0090  66 67 68 69 6a 6b 6c 6d  6e 6f 70 71 72 73 74 75   fghijklm nopqrstu
00a0  76 77 78 79 7a 7b 7c 7d  7e 7f 80 81 82 83 84 85   vwxyz{|} ~.......
00b0  86 87 88 89 8a 8b 8c 8d  8e 8f 90 91 92 93 94 95   ........ ........
00c0  96 97 98 99 9a 9b 9c 9d  9e 9f a0 a1 a2 a3 a4 a5   ........ ........
00d0  a6 a7 a8 a9 aa ab ac ad  ae af b0 b1 b2 b3 b4 b5   ........ ........
00e0  b6 b7 b8 b9 ba bb bc bd  be bf c0 c1 c2 c3 c4 c5   ........ ........
00f0  c6 c7 c8 c9 ca cb cc cd  ce cf d0 d1 d2 d3 d4 d5   ........ ........
0100  d6 d7 d8 d9 da db dc dd  de df e0 e1 e2 e3 e4 e5   ........ ........
0110  e6 e7 e8 e9 ea eb ec ed  ee ef f0 f1 f2 f3 f4 f5   ........ ........
```

Данные ping пакета

2. Traceroute:

```
[sigma@magma main ]$ sudo traceroute -I jovian
traceroute to jovian (146.185.143.190), 30 hops max, 60 byte packets
 1  router.asus.com (192.168.1.1)  6.391 ms  6.672 ms  7.607 ms
 2  188.243.32.1.pool.sknt.ru (188.243.32.1)  13.592 ms  13.653 ms  13.654 ms
 3  Router.sknt.ru (93.100.0.55)  14.254 ms  14.254 ms  14.253 ms
 4  185.37.128.22 (185.37.128.22)  13.304 ms  14.204 ms  14.232 ms
 5  ae8.RT.KM.SPB.RU.retn.net (87.245.252.157)  14.231 ms  14.230 ms  14.229 ms
 6  ae3-8.RT.TC2.AMS.NL.retn.net (87.245.233.17)  45.711 ms  35.284 ms  33.446 ms
 7  80.249.211.98 (80.249.211.98)  33.946 ms  44.218 ms  44.221 ms
 8  * * *
 9  Jovian (146.185.143.190)  45.163 ms  45.159 ms  45.148 ms
```

Выполнение traceroute

64	3.839587888	192.168.1.106	146.185.143.190	ICMP	74 Echo (ping) request id=0x5a19, seq=32/8192, ttl=11 (reply in 81)
65	3.839904603	87.245.233.17	192.168.1.106	ICMP	70 Time-to-live exceeded (Time to live exceeded in transit)
66	3.839948704	192.168.1.106	146.185.143.190	ICMP	74 Echo (ping) request id=0x5a19, seq=33/8448, ttl=11 (reply in 82)
67	3.865365293	87.245.233.17	192.168.1.106	ICMP	70 Time-to-live exceeded (Time to live exceeded in transit)
68	3.865450897	192.168.1.106	146.185.143.190	ICMP	74 Echo (ping) request id=0x5a19, seq=34/8704, ttl=12 (reply in 84)
69	3.865910416	80.249.211.98	192.168.1.106	ICMP	70 Time-to-live exceeded (Time to live exceeded in transit)
71	3.876187372	80.249.211.98	192.168.1.106	ICMP	70 Time-to-live exceeded (Time to live exceeded in transit)
72	3.876194043	80.249.211.98	192.168.1.106	ICMP	70 Time-to-live exceeded (Time to live exceeded in transit)
73	3.877144992	146.185.143.190	192.168.1.106	ICMP	74 Echo (ping) reply id=0x5a19, seq=30/7680, ttl=56 (request in 60)
74	3.877152077	146.185.143.190	192.168.1.106	ICMP	74 Echo (ping) reply id=0x5a19, seq=31/7936, ttl=56 (request in 61)
75	3.877155109	146.185.143.190	192.168.1.106	ICMP	74 Echo (ping) reply id=0x5a19, seq=27/6912, ttl=56 (request in 57)
76	3.877157174	146.185.143.190	192.168.1.106	ICMP	74 Echo (ping) reply id=0x5a19, seq=25/6400, ttl=56 (request in 55)
77	3.877159977	146.185.143.190	192.168.1.106	ICMP	74 Echo (ping) reply id=0x5a19, seq=26/6656, ttl=56 (request in 56)
78	3.877162113	146.185.143.190	192.168.1.106	ICMP	74 Echo (ping) reply id=0x5a19, seq=28/7168, ttl=56 (request in 58)
79	3.877164190	146.185.143.190	192.168.1.106	ICMP	74 Echo (ping) reply id=0x5a19, seq=29/7424, ttl=56 (request in 59)
81	3.877173525	146.185.143.190	192.168.1.106	ICMP	74 Echo (ping) reply id=0x5a19, seq=32/8192, ttl=56 (request in 64)
82	3.877176071	146.185.143.190	192.168.1.106	ICMP	74 Echo (ping) reply id=0x5a19, seq=33/8448, ttl=56 (request in 66)
83	3.877238805	192.168.1.106	146.185.143.190	ICMP	74 Echo (ping) request id=0x5a19, seq=35/8960, ttl=12 (reply in 85)
84	3.907561138	146.185.143.190	192.168.1.106	ICMP	74 Echo (ping) reply id=0x5a19, seq=34/8704, ttl=56 (request in 68)
85	3.912792207	146.185.143.190	192.168.1.106	ICMP	74 Echo (ping) reply id=0x5a19, seq=35/8960, ttl=56 (request in 83)

Wireshark's trace

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

   В заголовке содержится 20, а в поле данных - 32.

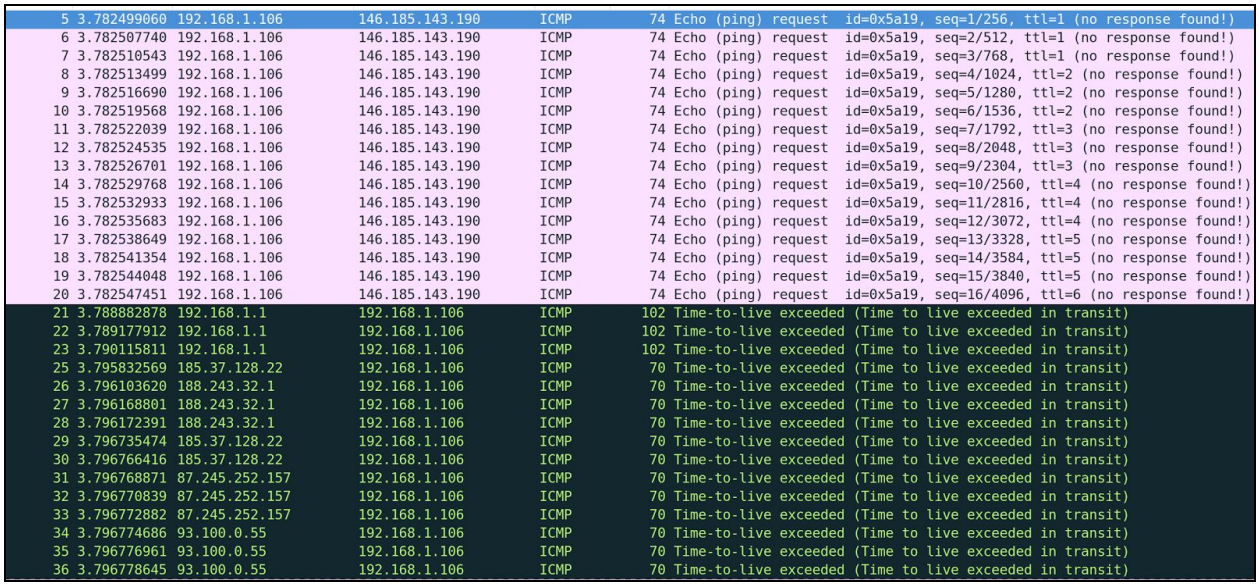
```
▾ Internet Protocol Version 4, Src: 192.168.1.106, Dst: 146.185.143.190
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
```

Описание заголовка

```
▾ Data (32 bytes)
    Data: 48494a4b4c4d4e4f505152535455565758595a5b5c5d5e5f...
```

Описание данных

2. **Как и почему именно так изменяется поле TTL в следующих друг за другом ICMP-пакетах traceroute?**
   Значение поля TTL увеличивается на каждые три запроса, это происходит так ввиду основ функционирования утилиты traceroute.



5	3.782499060	192.168.1.106	146.185.143.190	ICMP	74 Echo (ping) request id=0x5a19, seq=1/256, ttl=1 (no response found!)
6	3.782507740	192.168.1.106	146.185.143.190	ICMP	74 Echo (ping) request id=0x5a19, seq=2/512, ttl=1 (no response found!)
7	3.782510543	192.168.1.106	146.185.143.190	ICMP	74 Echo (ping) request id=0x5a19, seq=3/768, ttl=1 (no response found!)
8	3.782513499	192.168.1.106	146.185.143.190	ICMP	74 Echo (ping) request id=0x5a19, seq=4/1024, ttl=2 (no response found!)
9	3.782516690	192.168.1.106	146.185.143.190	ICMP	74 Echo (ping) request id=0x5a19, seq=5/1280, ttl=2 (no response found!)
10	3.782519568	192.168.1.106	146.185.143.190	ICMP	74 Echo (ping) request id=0x5a19, seq=6/1536, ttl=2 (no response found!)
11	3.782522039	192.168.1.106	146.185.143.190	ICMP	74 Echo (ping) request id=0x5a19, seq=7/1792, ttl=3 (no response found!)
12	3.782524535	192.168.1.106	146.185.143.190	ICMP	74 Echo (ping) request id=0x5a19, seq=8/2048, ttl=3 (no response found!)
13	3.782526701	192.168.1.106	146.185.143.190	ICMP	74 Echo (ping) request id=0x5a19, seq=9/2304, ttl=3 (no response found!)
14	3.782529768	192.168.1.106	146.185.143.190	ICMP	74 Echo (ping) request id=0x5a19, seq=10/2560, ttl=4 (no response found!)
15	3.782532933	192.168.1.106	146.185.143.190	ICMP	74 Echo (ping) request id=0x5a19, seq=11/2816, ttl=4 (no response found!)
16	3.782535683	192.168.1.106	146.185.143.190	ICMP	74 Echo (ping) request id=0x5a19, seq=12/3072, ttl=4 (no response found!)
17	3.782538649	192.168.1.106	146.185.143.190	ICMP	74 Echo (ping) request id=0x5a19, seq=13/3328, ttl=5 (no response found!)
18	3.782541354	192.168.1.106	146.185.143.190	ICMP	74 Echo (ping) request id=0x5a19, seq=14/3584, ttl=5 (no response found!)
19	3.782544048	192.168.1.106	146.185.143.190	ICMP	74 Echo (ping) request id=0x5a19, seq=15/3840, ttl=5 (no response found!)
20	3.782547451	192.168.1.106	146.185.143.190	ICMP	74 Echo (ping) request id=0x5a19, seq=16/4096, ttl=6 (no response found!)
21	3.788882878	192.168.1.1	192.168.1.106	ICMP	102 Time-to-live exceeded (Time to live exceeded in transit)
22	3.789177912	192.168.1.1	192.168.1.106	ICMP	102 Time-to-live exceeded (Time to live exceeded in transit)
23	3.790115811	192.168.1.1	192.168.1.106	ICMP	102 Time-to-live exceeded (Time to live exceeded in transit)
25	3.795832569	185.37.128.22	192.168.1.106	ICMP	70 Time-to-live exceeded (Time to live exceeded in transit)
26	3.796103620	188.243.32.1	192.168.1.106	ICMP	70 Time-to-live exceeded (Time to live exceeded in transit)
27	3.796168801	188.243.32.1	192.168.1.106	ICMP	70 Time-to-live exceeded (Time to live exceeded in transit)
28	3.796172391	188.243.32.1	192.168.1.106	ICMP	70 Time-to-live exceeded (Time to live exceeded in transit)
29	3.796735474	185.37.128.22	192.168.1.106	ICMP	70 Time-to-live exceeded (Time to live exceeded in transit)
30	3.796766416	185.37.128.22	192.168.1.106	ICMP	70 Time-to-live exceeded (Time to live exceeded in transit)
31	3.796768871	87.245.252.157	192.168.1.106	ICMP	70 Time-to-live exceeded (Time to live exceeded in transit)
32	3.796770839	87.245.252.157	192.168.1.106	ICMP	70 Time-to-live exceeded (Time to live exceeded in transit)
33	3.796772882	87.245.252.157	192.168.1.106	ICMP	70 Time-to-live exceeded (Time to live exceeded in transit)
34	3.796774686	93.100.0.55	192.168.1.106	ICMP	70 Time-to-live exceeded (Time to live exceeded in transit)
35	3.796776961	93.100.0.55	192.168.1.106	ICMP	70 Time-to-live exceeded (Time to live exceeded in transit)
36	3.796778645	93.100.0.55	192.168.1.106	ICMP	70 Time-to-live exceeded (Time to live exceeded in transit)

3. **Чем отличаются ICMP-пакеты, генерируемые утилитой traceroute, от ICMP-пакетов, генерируемых утилитой ping?**
   a. Значение поля TTL
   b. Размер данных
   c. Содержание данных

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

   ICMP reply сигнализирует об успешном получении запроса и является ответом на него. Посылается конечным узлом сети.
   ICMP error сигнализирует о том, что процесс передачи не будет завершен полностью (в данном случае из-за обнуления поля TTL пакета). Посылается промежуточными узлами сети.

   Этим пакеты необходимы для функционирования утилиты traceroute.

5. **Что изменится в работе traceroute, если убрать ключ “-d”? Какой дополнительный трафик при этом будет генерироваться?**
   Ключ -d для Windows-версии утилиты предотвращает попытки разрешить IP-адреса промежуточных узлов в символьные ссылки. При его отсутствии будет генерироваться дополнительный DNS-трафик. В linux аналогом этого ключа является ключ -n

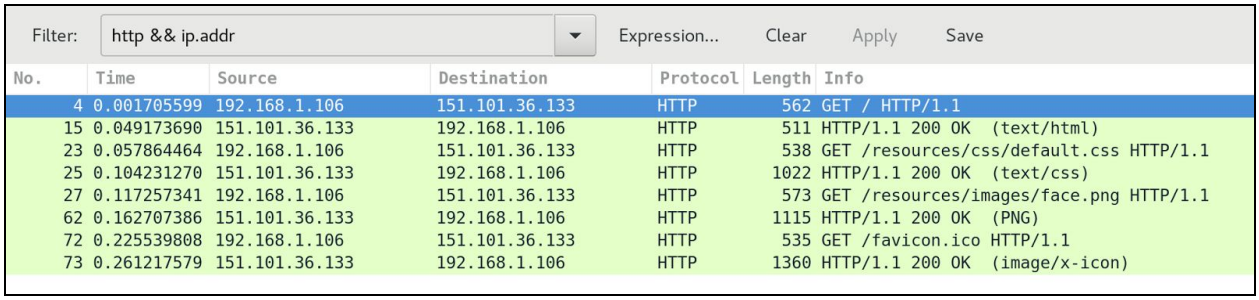
```
[sigma@magma main ]$ sudo traceroute -I -n jovian
traceroute to jovian (146.185.143.190), 30 hops max, 60 byte packets
 1  192.168.1.1  4.660 ms  4.919 ms  5.242 ms
 2  188.243.32.1  12.558 ms  12.614 ms  12.645 ms
 3  93.100.0.55  13.020 ms  13.357 ms  13.357 ms
 4  185.37.128.22  22.054 ms  23.898 ms  23.904 ms
 5  87.245.252.157  23.903 ms  23.903 ms  23.902 ms
 6  87.245.233.17  43.641 ms  39.463 ms  39.266 ms
 7  80.249.211.98  57.091 ms  51.049 ms  51.039 ms
 8  * * *
 9  146.185.143.190  58.482 ms  50.787 ms  48.927 ms
```

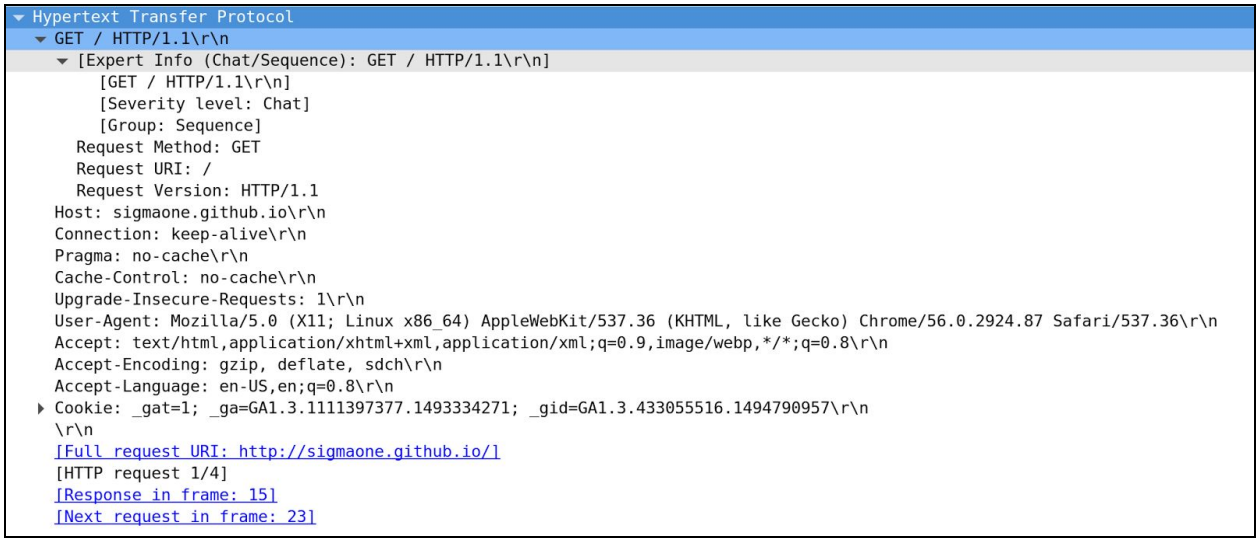
3. HTTP

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.

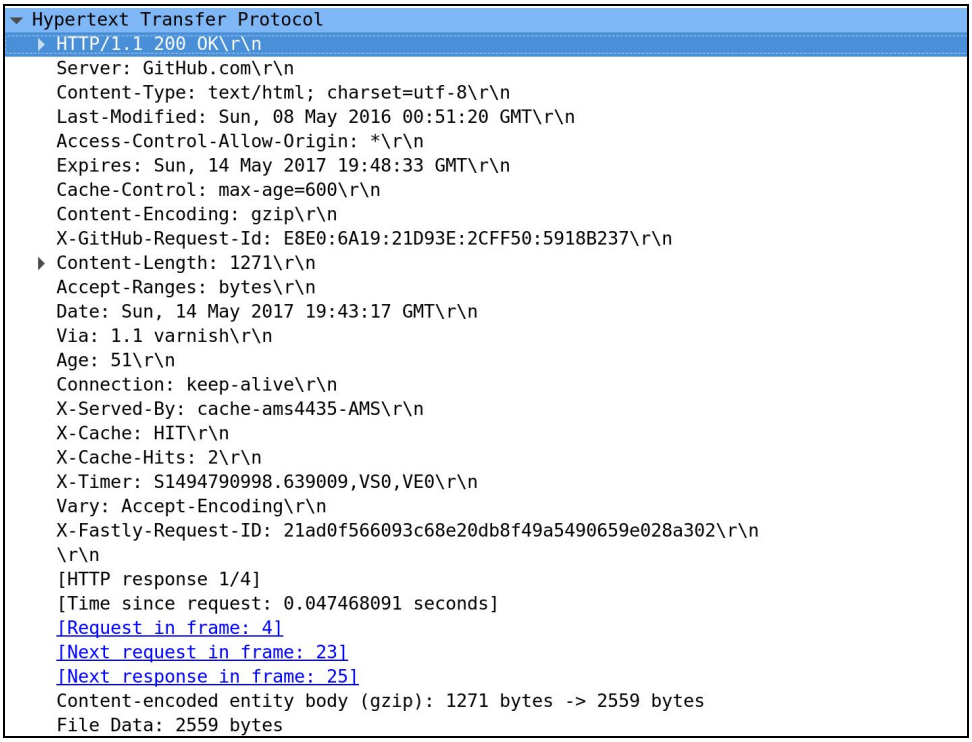
По результатам анализа собранной трассы покажите, каким образом протокол HTTP передавал содержимое страницы при первичном посещении страницы и при вторичном запросе-обновлении от браузера (т.е. при различных видах GET-запросов).



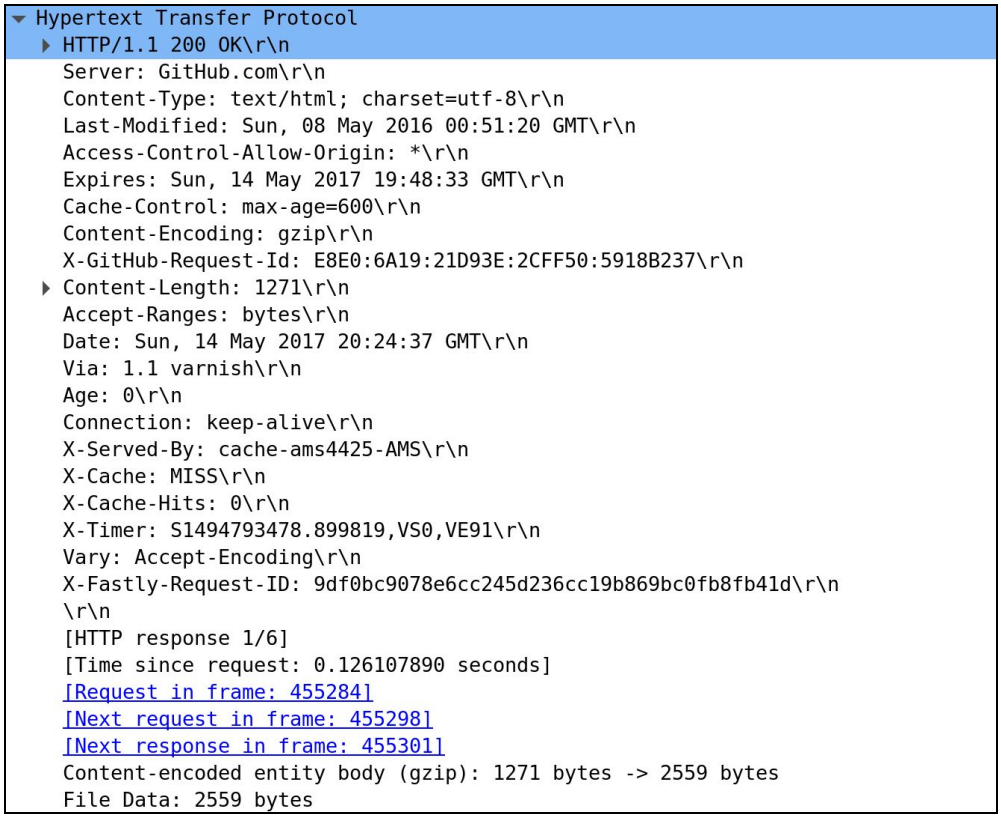


Первичный запрос к серверу

Вторичный запрос к серверу



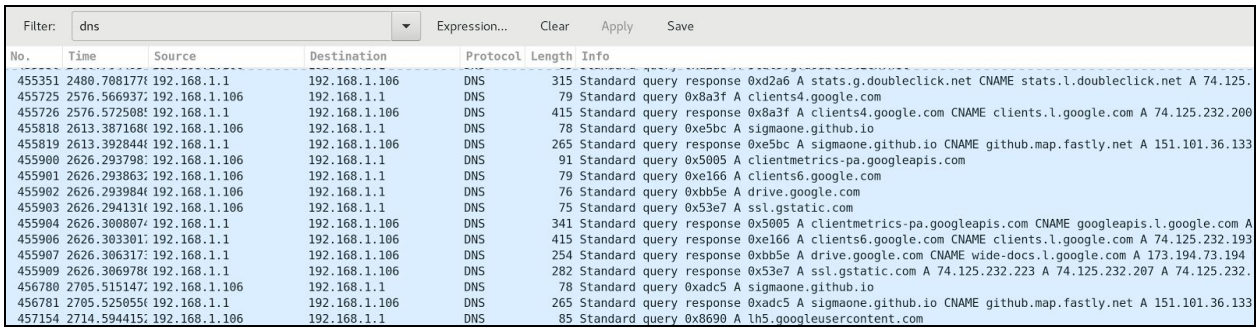
Первичный ответ сервера



Вторичный ответ сервера

4. DNS

Необходимо отследить и проанализировать трафик протокола DNS



Filter: dns — Expression... Clear Apply Save

No.	Time	Source	Destination	Protocol	Length	Info
455351	2480.708177	192.168.1.1	192.168.1.106	DNS	315	Standard query response 0xd2a6 A stats.g.doubleclick.net CNAME stats.l.doubleclick.net A 74.125.
455725	2576.566937	192.168.1.106	192.168.1.1	DNS	79	Standard query 0x8a3f A clients4.google.com
455726	2576.572508	192.168.1.1	192.168.1.106	DNS	415	Standard query response 0x8a3f A clients4.google.com CNAME clients.l.google.com A 74.125.232.200
455818	2613.387168	192.168.1.106	192.168.1.1	DNS	78	Standard query 0xe5bc A sigmaone.github.io
455819	2613.392844	192.168.1.1	192.168.1.106	DNS	265	Standard query response 0xe5bc A sigmaone.github.io CNAME github.map.fastly.net A 151.101.36.133
455900	2626.293798	192.168.1.106	192.168.1.1	DNS	91	Standard query 0x5005 A clientmetrics-pa.googleapis.com
455901	2626.293863	192.168.1.106	192.168.1.1	DNS	79	Standard query 0xe166 A clients6.google.com
455902	2626.293984	192.168.1.106	192.168.1.1	DNS	76	Standard query 0xbb5e A drive.google.com
455903	2626.294131	192.168.1.106	192.168.1.1	DNS	75	Standard query 0x53e7 A ssl.gstatic.com
455904	2626.300807	192.168.1.1	192.168.1.106	DNS	341	Standard query response 0x5005 A clientmetrics-pa.googleapis.com CNAME googleapis.l.google.com A
455906	2626.303301	192.168.1.1	192.168.1.106	DNS	415	Standard query response 0xe166 A clients6.google.com CNAME clients.l.google.com A 74.125.232.193
455907	2626.306317	192.168.1.1	192.168.1.106	DNS	254	Standard query response 0xbb5e A drive.google.com CNAME wide-docs.l.google.com A 173.194.73.194
455909	2626.306978	192.168.1.1	192.168.1.106	DNS	282	Standard query response 0x53e7 A ssl.gstatic.com A 74.125.232.223 A 74.125.232.207 A 74.125.232.
456780	2705.515147	192.168.1.106	192.168.1.1	DNS	78	Standard query 0xadc5 A sigmaone.github.io
456781	2705.525055	192.168.1.1	192.168.1.106	DNS	265	Standard query response 0xadc5 A sigmaone.github.io CNAME github.map.fastly.net A 151.101.36.133
457154	2714.594415	192.168.1.106	192.168.1.1	DNS	85	Standard query 0x8690 A lh5.googleusercontent.com

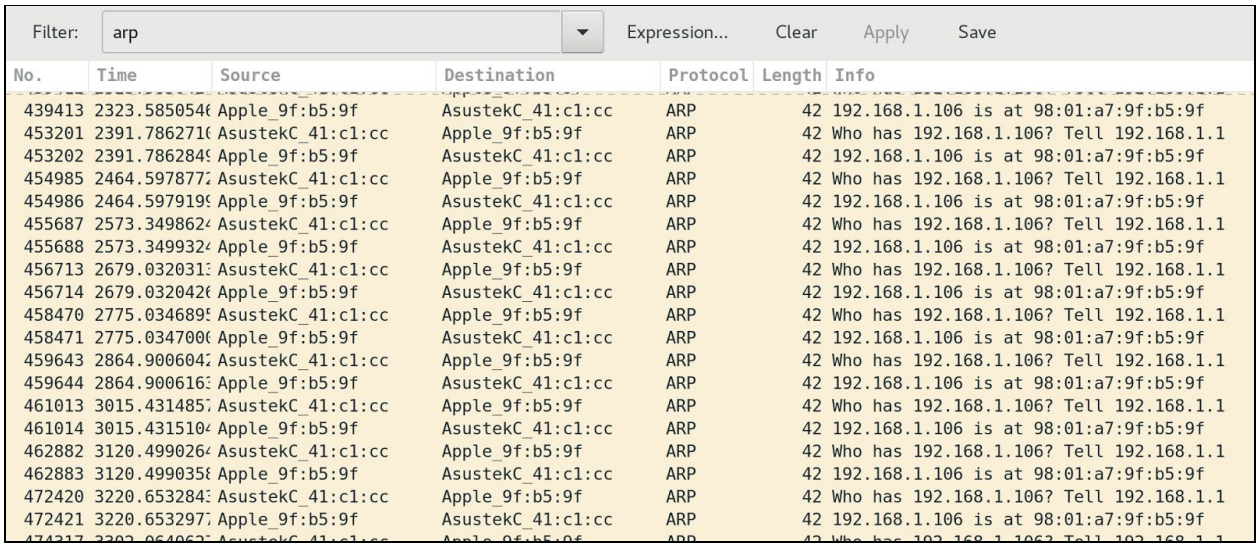
1. **Почему адрес, на который отправлен DNS-запрос, не совпадает с адресом посещаемого сайта?**
   Потому что запрос отправляется на специальный DNS-сервер, производящий разрешение символьных имён в IP-адреса

2. **Какие бывают типы DNS-запросов?**
   https://support.opendns.com/hc/en-us/articles/227986607-FAQ-what-are-the-DNS-Request-Types-

3. **В какой ситуации нужно выполнять независимые DNS-запросы для получения содержащихся на сайте изображений?**
   В этом есть необходимость тогда, когда изображения находятся на других доменах (например, CDN). В таком случае требуется дополнительное разрешение имён серверов с изображениями.

5. ARP

Необходимо отследить и проанализировать трафик протокола ARP



Filter: arp — Expression... Clear Apply Save

No.	Time	Source	Destination	Protocol	Length	Info
439413	2323.585054	Apple_9f:b5:9f	AsustekC_41:c1:cc	ARP	42	192.168.1.106 is at 98:01:a7:9f:b5:9f
453201	2391.786271	AsustekC_41:c1:cc	Apple_9f:b5:9f	ARP	42	Who has 192.168.1.106? Tell 192.168.1.1
453202	2391.786284	Apple_9f:b5:9f	AsustekC_41:c1:cc	ARP	42	192.168.1.106 is at 98:01:a7:9f:b5:9f
454985	2464.597877	AsustekC_41:c1:cc	Apple_9f:b5:9f	ARP	42	Who has 192.168.1.106? Tell 192.168.1.1
454986	2464.597919	Apple_9f:b5:9f	AsustekC_41:c1:cc	ARP	42	192.168.1.106 is at 98:01:a7:9f:b5:9f
455687	2573.349862	AsustekC_41:c1:cc	Apple_9f:b5:9f	ARP	42	Who has 192.168.1.106? Tell 192.168.1.1
455688	2573.349932	Apple_9f:b5:9f	AsustekC_41:c1:cc	ARP	42	192.168.1.106 is at 98:01:a7:9f:b5:9f
456713	2679.032031	AsustekC_41:c1:cc	Apple_9f:b5:9f	ARP	42	Who has 192.168.1.106? Tell 192.168.1.1
456714	2679.032042	Apple_9f:b5:9f	AsustekC_41:c1:cc	ARP	42	192.168.1.106 is at 98:01:a7:9f:b5:9f
458470	2775.034689	AsustekC_41:c1:cc	Apple_9f:b5:9f	ARP	42	Who has 192.168.1.106? Tell 192.168.1.1
458471	2775.034700	Apple_9f:b5:9f	AsustekC_41:c1:cc	ARP	42	192.168.1.106 is at 98:01:a7:9f:b5:9f
459643	2864.900604	AsustekC_41:c1:cc	Apple_9f:b5:9f	ARP	42	Who has 192.168.1.106? Tell 192.168.1.1
459644	2864.900616	Apple_9f:b5:9f	AsustekC_41:c1:cc	ARP	42	192.168.1.106 is at 98:01:a7:9f:b5:9f
461013	3015.431485	AsustekC_41:c1:cc	Apple_9f:b5:9f	ARP	42	Who has 192.168.1.106? Tell 192.168.1.1
461014	3015.431510	Apple_9f:b5:9f	AsustekC_41:c1:cc	ARP	42	192.168.1.106 is at 98:01:a7:9f:b5:9f
462882	3120.499026	AsustekC_41:c1:cc	Apple_9f:b5:9f	ARP	42	Who has 192.168.1.106? Tell 192.168.1.1
462883	3120.499035	Apple_9f:b5:9f	AsustekC_41:c1:cc	ARP	42	192.168.1.106 is at 98:01:a7:9f:b5:9f
472420	3220.653284	AsustekC_41:c1:cc	Apple_9f:b5:9f	ARP	42	Who has 192.168.1.106? Tell 192.168.1.1
472421	3220.653297	Apple_9f:b5:9f	AsustekC_41:c1:cc	ARP	42	192.168.1.106 is at 98:01:a7:9f:b5:9f

1. **Какие MAC-адреса присутствуют в захваченных пакетах ARP-протокола? Что означают эти адреса? Какие устройства они идентифицируют?**
   В захваченных пакетах присутствуют адреса устройств в сети, а также адрес широкого вещания ff:ff:ff:ff:ff:ff. Адреса устройств используются для идентификации в сети, адрес широкого вещания используется в поле Destination для того, чтобы пакет был доставлен всем устройствам в сети. Эти адреса идентифицируют все устройства в сети, будь то компьютеры, ноутбуки, роутеры, телефоны, или что-то иное.

2. **Для чего ARP-запрос содержит IP-адрес источника?**
   Это сделано для того, чтобы другие устройства в сети могли занести этот источник в свою ARP-таблицу

6. Nslookup

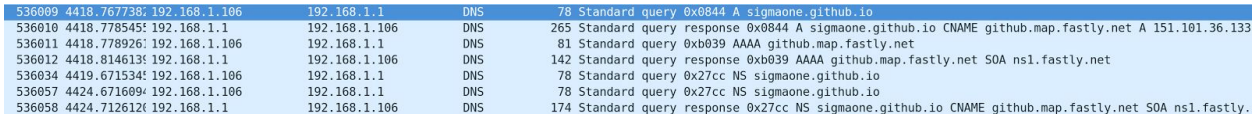
Необходимо отследить и проанализировать трафик протокола DNS.

```
[sigma@magma bin ]$ nslookup jovian
Server:         192.168.1.1
Address:        192.168.1.1#53

** server can't find jovian: NXDOMAIN
```

Эт потому что jovian прописан у меня в /etc/hosts

```
[sigma@magma bin ]$ nslookup sigmaone.github.io
Server:         192.168.1.1
Address:        192.168.1.1#53

Non-authoritative answer:
sigmaone.github.io      canonical name = github.map.fastly.net.
Name:   github.map.fastly.net
Address: 151.101.36.133

[sigma@magma bin ]$ nslookup -type=NS sigmaone.github.io
Server:         192.168.1.1
Address:        192.168.1.1#53

Non-authoritative answer:
sigmaone.github.io      canonical name = github.map.fastly.net.

Authoritative answers can be found from:
fastly.net
        origin = ns1.fastly.net
        mail addr = hostmaster.fastly.com
        serial = 2016110301
        refresh = 3600
        retry = 600
        expire = 604800
        minimum = 30
```

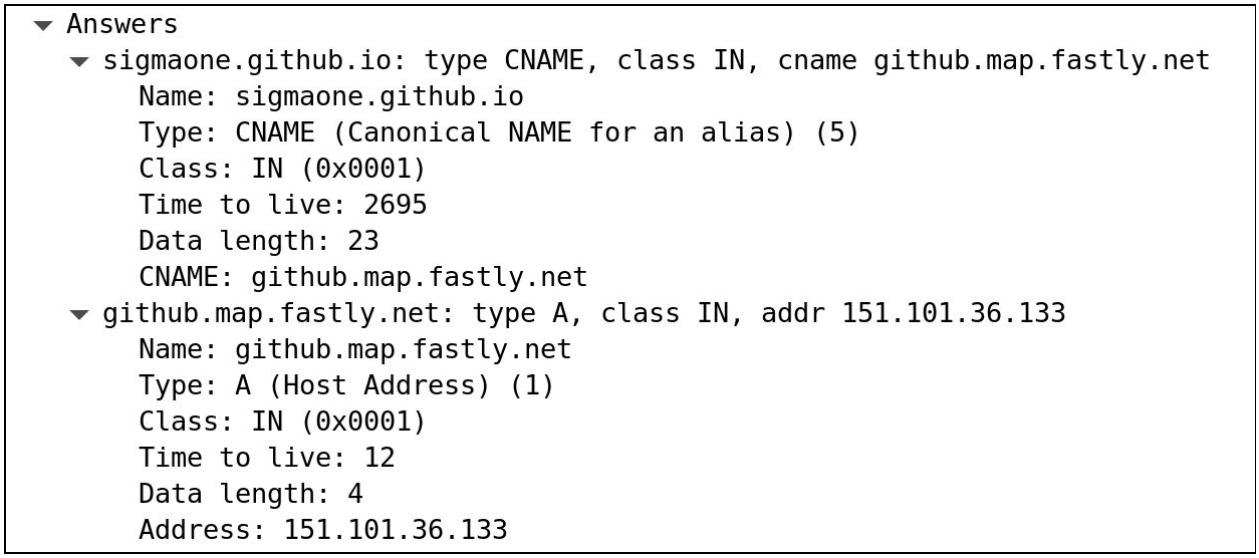
Пробуем sigmaone.github.io



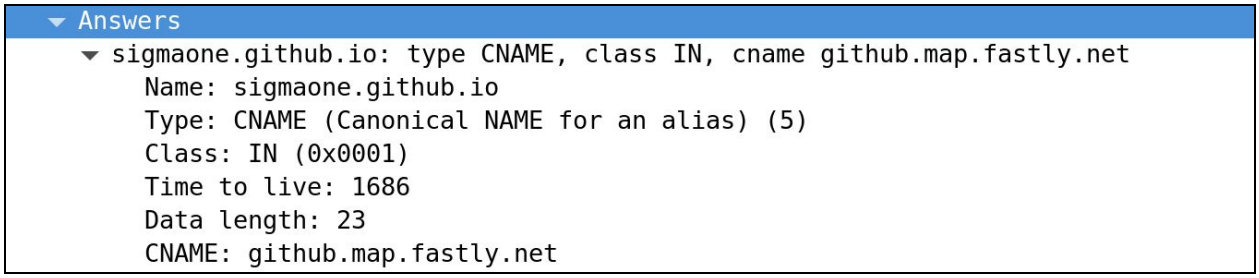
wireshark

1. **Чем различается трасса трафика при вызове nslookup с ключом -type=NS?**
   Просто происходят разные запросы.

2. **Что содержится в поле «Answers» DNS-ответа?**
   При запросе типа A в ответе содержится IPv4-адрес хоста. При запросе типа NS в SOA ответе содержится информация о домене, почтовом ящике, ответственном за доменную зону, а также временные интервалы, определяющие обновления DNS-записей



A request

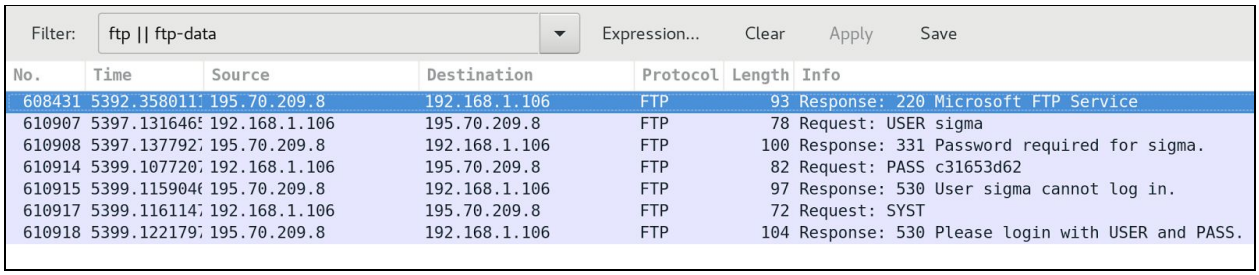


NS request

3. **Каковы имена серверов, возвращающих авторитативный (authoritative) отклик?**
   authoritative ответ мы получаем, если dns сервер имеет текущую запись. В моем случае этого нет, и dns спрашивает у вышестоящего dns.

10. FTP

Необходимо отследить и проанализировать трафик протокола FTP



No.	Time	Source	Destination	Protocol	Length	Info
608431	5392.358011	195.70.209.8	192.168.1.106	FTP	93	Response: 220 Microsoft FTP Service
610907	5397.131646	192.168.1.106	195.70.209.8	FTP	78	Request: USER sigma
610908	5397.137792	195.70.209.8	192.168.1.106	FTP	100	Response: 331 Password required for sigma.
610914	5399.107720	192.168.1.106	195.70.209.8	FTP	82	Request: PASS c31653d62
610915	5399.115904	195.70.209.8	192.168.1.106	FTP	97	Response: 530 User sigma cannot log in.
610917	5399.116114	192.168.1.106	195.70.209.8	FTP	72	Request: SYST
610918	5399.122179	195.70.209.8	192.168.1.106	FTP	104	Response: 530 Please login with USER and PASS.

1. **Сколько байт данных содержится в пакете FTP-DATA**
   Если оставшийся для загрузки размер файла > MTU, то MTU. Иначе - оставшийся для загрузки размер файла.

2. **Как выбирается порт транспортного уровня, который используется для передачи FTP-пакетов?**
   Канал для команд инстанциируется на порту 21. Канал для данных выбирается на случайном порте > 1023. Либо, если это немного другой вопрос, то в зависимости от режима FTP (активный / пассивный). В моём случае это пассивный

3. **Чем отличаются пакеты FTP от FTP-DATA?**
   FTP - являются служебными и используются для передачи команд.
   FTP-DATA применяется при загрузке файлов

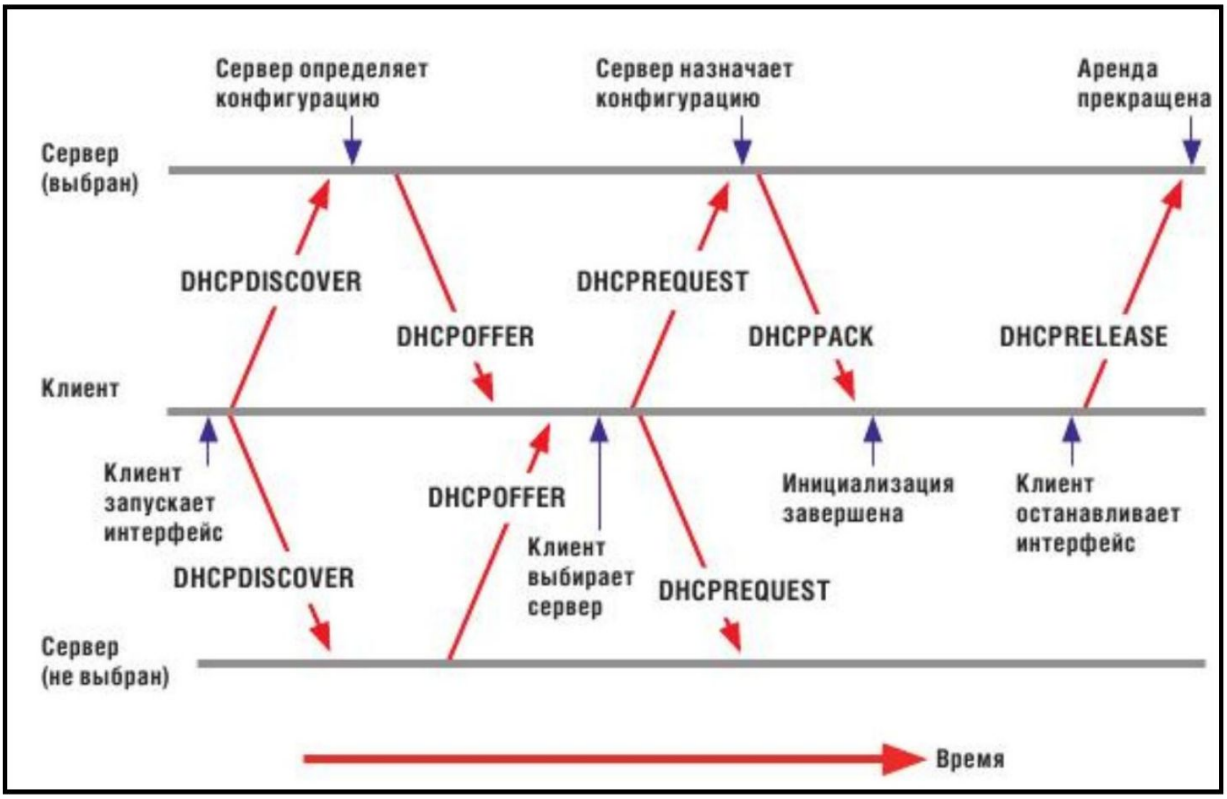
11. DHCP

Необходимо отследить и проанализировать трафик протокола DHCP.

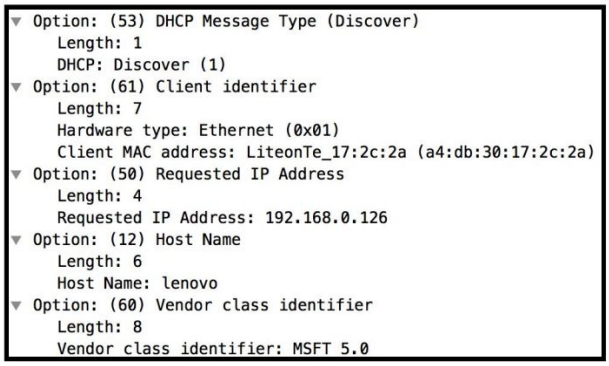
Filter: bootp Expression... Clear Apply Save

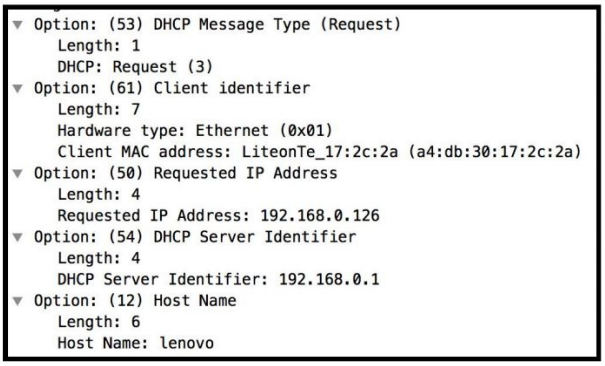
No.	Time	Source	Destination	Protocol	Length	Info
619717	5699.244809(	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0x9d9ca511
619718	5699.258286(	192.168.1.1	192.168.1.106	DHCP	342	DHCP Offer - Transaction ID 0x9d9ca511
619719	5699.258396(	0.0.0.0	255.255.255.255	DHCP	342	DHCP Request - Transaction ID 0x9d9ca511
619720	5699.263704(	192.168.1.1	192.168.1.106	DHCP	342	DHCP ACK - Transaction ID 0x9d9ca511

Нарисуйте временную диаграмму, иллюстрирующую последовательность обмена первыми четырьмя DHCP-пакетами Discover/Offer/Request/ACK



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.





2. **Как и почему менялись MAC- и IP-адреса источника и назначения в переданных DHCP-пакетах**
   При отравке Discover и Request пакетов IP-адрес источника равен 0.0.0.0, т.к. ему не присвоен IP, IP-адрес и MAC-адрес назначения соответствую широковещательным адресам, т.к. источнику неизвестно расположение DHCP-сервера.
   При отправке Offer и Ack пакетов MAC и IP адреса источника соответствуют адресам DHCP сервера, MAC адрес — адрес назначения, IP адрес назначения — адрес, предлагаемы/подтверженный IP адрес назначения.

3. **Каков IP-адрес DHCP-сервера?**
   192.168.1.1

4. **Что произойдёт, если очистить использованный фильтр “bootp”?**
   Будут отображены все пакеты, находившиеся в процессе передачи во время выполнения задания. Странный вопрос.

12. Заголовки

Level 1

Word Offset	Byte 0	Byte 1	Byte 2	Byte 3
0x0000	Destination MAC Address			
0x0010			Source MAC Address	
0x0020				
0x0030	Type (Level 2)			

Level 2-ARP

Word Offset	Byte 0	Byte 1	Byte 2	Byte 3
0x0000	Hardware Type (0x01)		Protocol Type (0x80)	
0x0010	HLEN (0x06)	PLEN (0x04)	Operation	
0x0020	Sender Hardware Address			
0x0030			Sender Protocol Address	
0x0040				
0x0050	Target Hardware Address			
0x0060			Target Protocol Address	
0x0070				

Level 2-IP

Word Offset	Byte 0	Byte 1	Byte 2	Byte 3
0x0000	Version (0x4)	Type (0x0)	Length	
0x0010	Identification		Flags	
0x0020	TTL	Protocol (0x1)	Checksum	
0x0030	Source IP			
0x0040	Destination IP			
0x0050	Data			
0x0060				
0x0070				

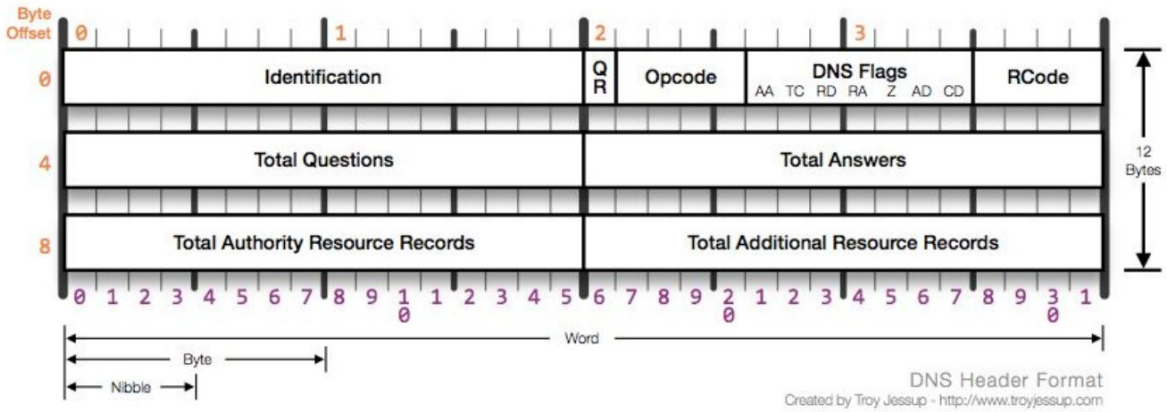
Level 3-ICMP

Word Offset	Byte 0	Byte 1	Byte 2	Byte 3
0x0000	Message Type	Code (0x0)	Checksum	
0x0010	Quench			
0x0050	Data			
0x0060				
0x0070				

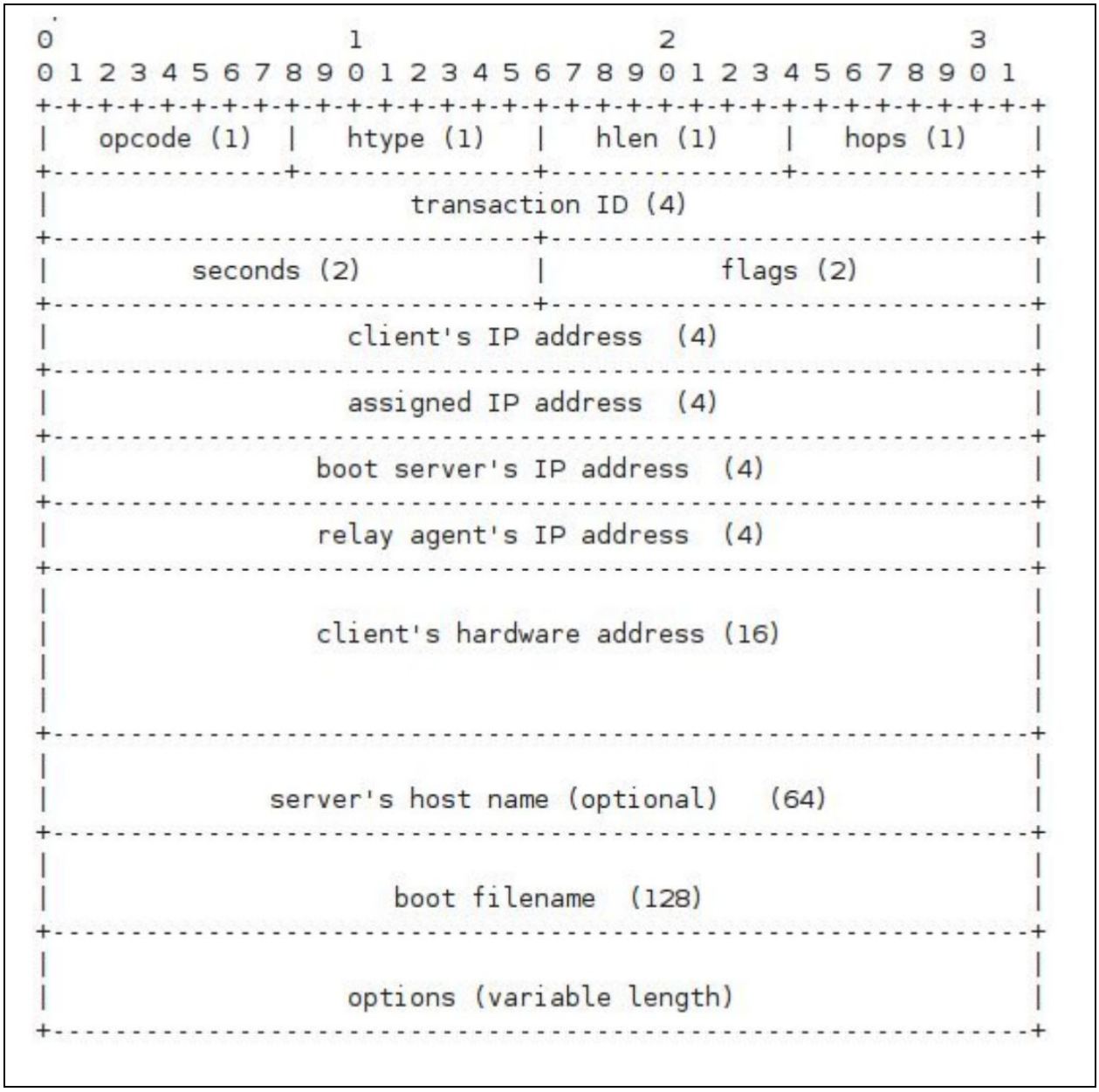
Level 3-UDP

Word Offset	Byte 0	Byte 1	Byte 2	Byte 3
0x0000	Source Port Number		Destination Port Number	
0x0010	Packet Length		Checksum	
0x0020	Data			
0x0030				
0x0040				

DNS Header



DHCP


0 1 2 3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
opcode (1)
htype (1)
hlen (1)
hops (1)
transaction ID (4)
seconds (2)
flags (2)
client's IP address (4)
assigned IP address (4)
boot server's IP address (4)
relay agent's IP address (4)
client's hardware address (16)
server's host name (optional) (64)
boot filename (128)
options (variable length)