Университет ИТМО

Факультет ПИиКТ

Сети ЭВМ и телекоммуникации

Лабораторная работа 3

Выполнил:

Студент группы P3310

Глушков Д. С.

Санкт-Петербург

2020 год

Цель работы:

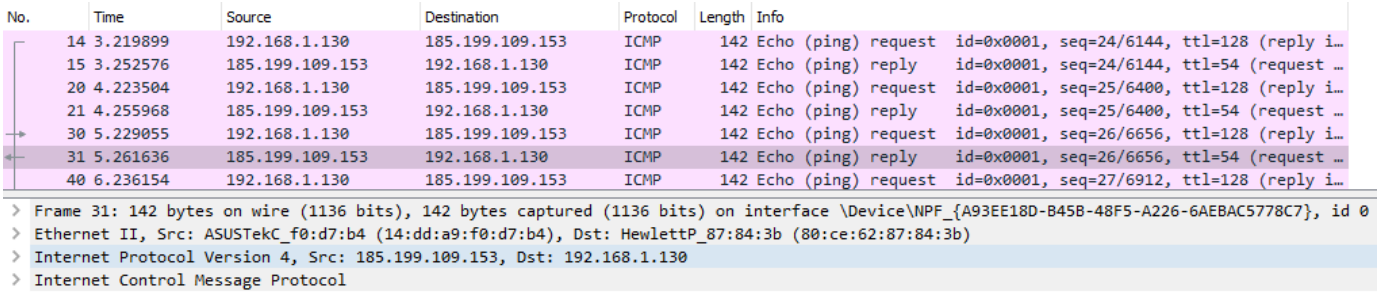
Изучить структуру протокольных блоков данных, анализируя реальный трафик на компьютере студента с помощью бесплатно распространяемой утилиты Wireshark.

Анализ трафика утилиты ping

В качестве исследуемого ресурса выбран allacee.github.io.

На рисунках представлен результат работы программы

```
ping -l 200 allacee.github.io
```



На рисунке видно, что данный пакет содержит заголовки, соответствующие 3 уровням:

1) Ethernet – канальный уровень
2) Internet Protocol – сетевой уровень
3) ICMP – прикладной уровень

1. **Имеет ли место фрагментация исходного пакета, какое поле на это указывает?**
   Фрагментация возникает только в том случае, когда объем передаваемых данных превышает максимальную вместимость пакета. Как видно на рисунке ниже, максимальное число байт, передаваемых в одном пакете равно 1480.

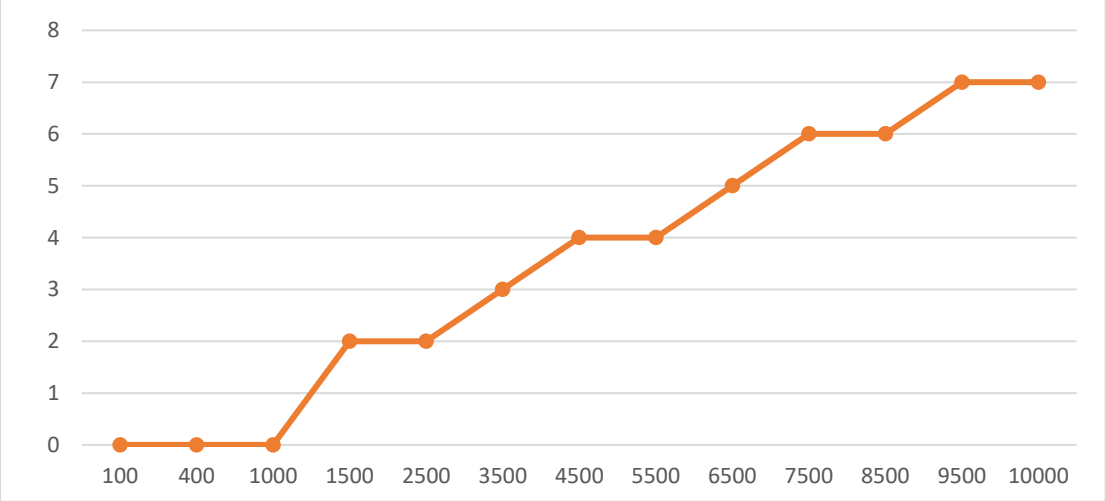
```
[4 IPv4 Fragments (5008 bytes): #12333(1480), #12334(1480), #12335(1480), #12336(568)]
    [Frame: 12333, payload: 0-1479 (1480 bytes)]
    [Frame: 12334, payload: 1480-2959 (1480 bytes)]
    [Frame: 12335, payload: 2960-4439 (1480 bytes)]
    [Frame: 12336, payload: 4440-5007 (568 bytes)]
    [Fragment count: 4]
```

2. **Какая информация указывает, является ли фрагмент пакета последним или промежуточным?**
   Раздел Flags фрагмента содержит информацию о том, есть ли фрагменты после него.

```
▼ Flags: 0x00b9
    0... .... .... .... = Reserved bit: 0
    .0.. .... .... .... = Don't fragment: 0
    ..0. .... .... .... = More fragments: 1
...0 0101 1100 1000 = Fragment offset: 1480
Time to live: 128
```

3. **Чему равно количество фрагментов при передаче ping-пакетов?**
   Как следует из пункта 1, максимальная длина пакета – 1480, следовательно, количество фрагментов = длина передаваемого сообщения / 1480 с округлением в меньшую сторону.

4. **Построить график, в котором на оси абсцисс находится размер_пакета, а по оси ординат –количество фрагментов, на которое был разделён каждый ping-пакет.**
   На рисунке ниже представлена зависимость количества фрагментов от размера пакета.



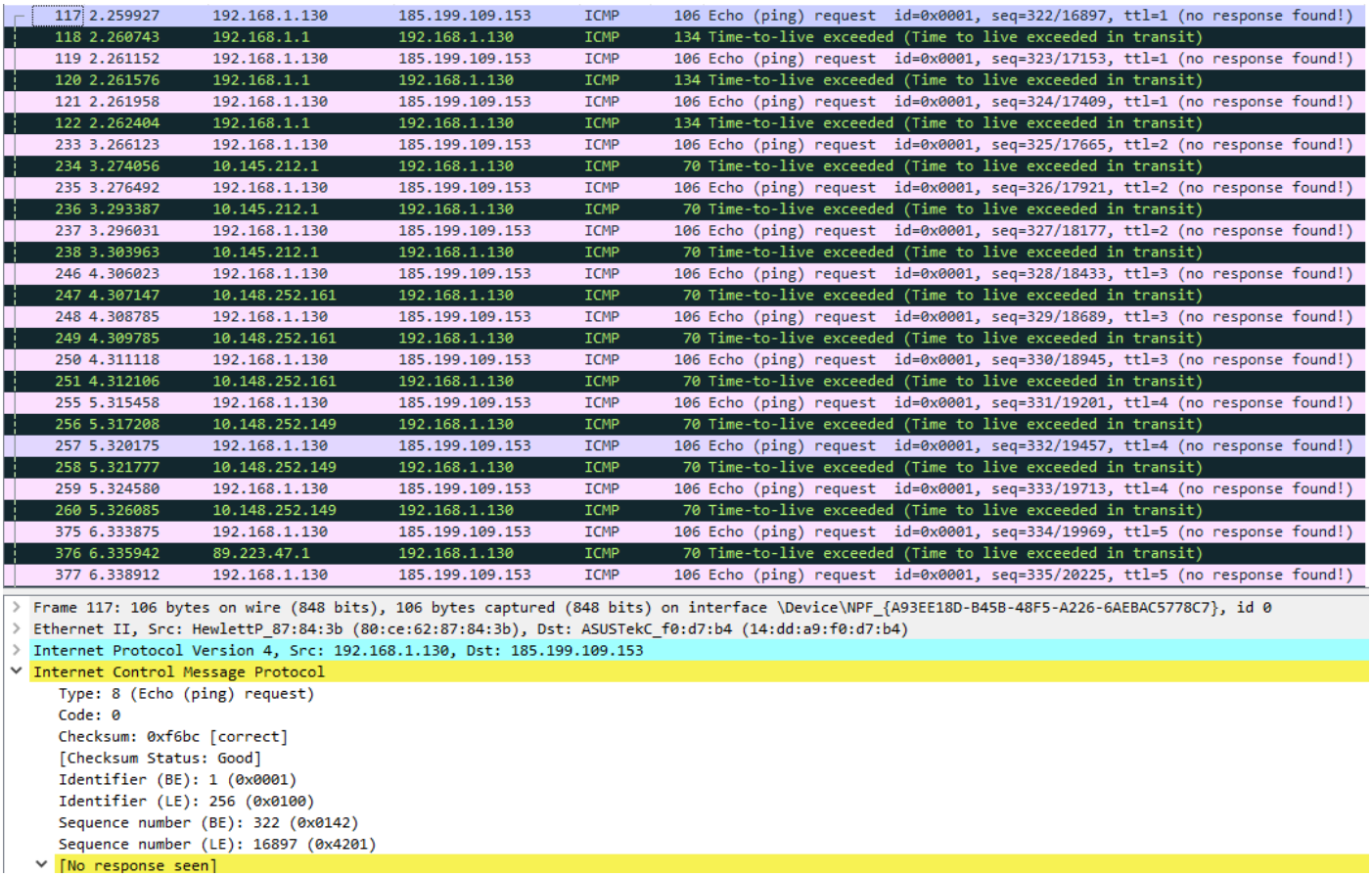
5. **Как изменить поле TTL с помощью утилиты ping?**
   `ping -i TTL` - Задание срока жизни пакета (поле "Time To Live").

6. **Что содержится в поле данных ping-пакета?**
   В поле данных содержатся случайные символы

```
80 ce 62 87 84 3b 14 dd  a9 f0 d7 b4 08 00 45 00   ··b··;·· ·····E·
00 80 ec 0a 00 00 36 01  ae e7 b9 c7 6d 99 c0 a8   ·····6· ····m···
01 82 00 00 f2 d3 00 01  00 1a 61 62 63 64 65 66   ······ ··abcdef
67 68 69 6a 6b 6c 6d 6e  6f 70 71 72 73 74 75 76   ghijklmn opqrstuv
77 61 62 63 64 65 66 67  68 69 6a 6b 6c 6d 6e 6f   wabcdefg hijklmno
70 71 72 73 74 75 76 77  61 62 63 64 65 66 67 68   pqrstuvw abcdefgh
69 6a 6b 6c 6d 6e 6f 70  71 72 73 74 75 76 77 61   ijklmnop qrstuvwa
62 63 64 65 66 67 68 69  6a 6b 6c 6d 6e 6f 70 71   bcdefghi jklmnopq
72 73 74 75 76 77 61 62  63 64 65 66 67 68         rstuvwab cdefgh
```

Анализ трафика утилиты tracert (traceroute)

```
tracert -d allacee.github.com
```



117	2.259927	192.168.1.130	185.199.109.153	ICMP	106 Echo (ping) request id=0x0001, seq=322/16897, ttl=1 (no response found!)
118	2.260743	192.168.1.1	192.168.1.130	ICMP	134 Time-to-live exceeded (Time to live exceeded in transit)
119	2.261152	192.168.1.130	185.199.109.153	ICMP	106 Echo (ping) request id=0x0001, seq=323/17153, ttl=1 (no response found!)
120	2.261576	192.168.1.1	192.168.1.130	ICMP	134 Time-to-live exceeded (Time to live exceeded in transit)
121	2.261958	192.168.1.130	185.199.109.153	ICMP	106 Echo (ping) request id=0x0001, seq=324/17409, ttl=1 (no response found!)
122	2.262404	192.168.1.1	192.168.1.130	ICMP	134 Time-to-live exceeded (Time to live exceeded in transit)
233	3.266123	192.168.1.130	185.199.109.153	ICMP	106 Echo (ping) request id=0x0001, seq=325/17665, ttl=2 (no response found!)
234	3.274056	10.145.212.1	192.168.1.130	ICMP	70 Time-to-live exceeded (Time to live exceeded in transit)
235	3.276492	192.168.1.130	185.199.109.153	ICMP	106 Echo (ping) request id=0x0001, seq=326/17921, ttl=2 (no response found!)
236	3.293387	10.145.212.1	192.168.1.130	ICMP	70 Time-to-live exceeded (Time to live exceeded in transit)
237	3.296031	192.168.1.130	185.199.109.153	ICMP	106 Echo (ping) request id=0x0001, seq=327/18177, ttl=2 (no response found!)
238	3.303963	10.145.212.1	192.168.1.130	ICMP	70 Time-to-live exceeded (Time to live exceeded in transit)
246	4.306023	192.168.1.130	185.199.109.153	ICMP	106 Echo (ping) request id=0x0001, seq=328/18433, ttl=3 (no response found!)
247	4.307147	10.148.252.161	192.168.1.130	ICMP	70 Time-to-live exceeded (Time to live exceeded in transit)
248	4.308785	192.168.1.130	185.199.109.153	ICMP	106 Echo (ping) request id=0x0001, seq=329/18689, ttl=3 (no response found!)
249	4.309785	10.148.252.161	192.168.1.130	ICMP	70 Time-to-live exceeded (Time to live exceeded in transit)
250	4.311118	192.168.1.130	185.199.109.153	ICMP	106 Echo (ping) request id=0x0001, seq=330/18945, ttl=3 (no response found!)
251	4.312106	10.148.252.161	192.168.1.130	ICMP	70 Time-to-live exceeded (Time to live exceeded in transit)
255	5.315458	192.168.1.130	185.199.109.153	ICMP	106 Echo (ping) request id=0x0001, seq=331/19201, ttl=4 (no response found!)
256	5.317208	10.148.252.149	192.168.1.130	ICMP	70 Time-to-live exceeded (Time to live exceeded in transit)
257	5.320175	192.168.1.130	185.199.109.153	ICMP	106 Echo (ping) request id=0x0001, seq=332/19457, ttl=4 (no response found!)
258	5.321777	10.148.252.149	192.168.1.130	ICMP	70 Time-to-live exceeded (Time to live exceeded in transit)
259	5.324580	192.168.1.130	185.199.109.153	ICMP	106 Echo (ping) request id=0x0001, seq=333/19713, ttl=4 (no response found!)
260	5.326085	10.148.252.149	192.168.1.130	ICMP	70 Time-to-live exceeded (Time to live exceeded in transit)
375	6.333875	192.168.1.130	185.199.109.153	ICMP	106 Echo (ping) request id=0x0001, seq=334/19969, ttl=5 (no response found!)
376	6.335942	89.223.47.1	192.168.1.130	ICMP	70 Time-to-live exceeded (Time to live exceeded in transit)
377	6.338912	192.168.1.130	185.199.109.153	ICMP	106 Echo (ping) request id=0x0001, seq=335/20225, ttl=5 (no response found!)

```
> Frame 117: 106 bytes on wire (848 bits), 106 bytes captured (848 bits) on interface \Device\NPF_{A93EE18D-B45B-48F5-A226-6AEBAC5778C7}, id 0
> Ethernet II, Src: HewlettP_87:84:3b (80:ce:62:87:84:3b), Dst: ASUSTekC_f0:d7:b4 (14:dd:a9:f0:d7:b4)
> Internet Protocol Version 4, Src: 192.168.1.130, Dst: 185.199.109.153
v Internet Control Message Protocol
    Type: 8 (Echo (ping) request)
    Code: 0
    Checksum: 0xf6bc [correct]
    [Checksum Status: Good]
    Identifier (BE): 1 (0x0001)
    Identifier (LE): 256 (0x0100)
    Sequence number (BE): 322 (0x0142)
    Sequence number (LE): 16897 (0x4201)
  v [No response seen]
```

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

   Длина IP заголовка 20 байт.

   Длина поля данных = общая длина – длина заголовка = 92 – 20 = 72

```
v Internet Protocol Version 4, Src: 192.168.1.130, Dst: 185.199.109.153
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 92
    Identification: 0x5718 (22296)
```

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

   После отправки трёх пакетов TTL увеличивается на единицу. При попадании в каждый узел значение TTL уменьшается на единицу. TRACERT отправляет первого эхо-пакета с TTL равным 1 и увеличивает значение TTL на 1 для каждого последующего, отправляемого пока назначение не ответит или пока не будет достигнуто максимальное значение поля TTL. Сообщений ICMP «Time

Exceeded», который промежуточные маршрутизаторы отправить назад отображается маршрут.

529	10.400258	192.168.1.130	185.199.109.153	ICMP	106 Echo (ping) request id=0x0001, seq=346/23041, ttl=9 (no response found!)
530	10.432127	87.245.233.17	192.168.1.130	ICMP	70 Time-to-live exceeded (Time to live exceeded in transit)
531	10.432816	192.168.1.130	185.199.109.153	ICMP	106 Echo (ping) request id=0x0001, seq=347/23297, ttl=9 (no response found!)
532	10.464547	87.245.233.17	192.168.1.130	ICMP	70 Time-to-live exceeded (Time to live exceeded in transit)
533	10.465259	192.168.1.130	185.199.109.153	ICMP	106 Echo (ping) request id=0x0001, seq=348/23553, ttl=9 (no response found!)
534	10.497256	87.245.233.17	192.168.1.130	ICMP	70 Time-to-live exceeded (Time to live exceeded in transit)
537	11.468385	192.168.1.130	185.199.109.153	ICMP	106 Echo (ping) request id=0x0001, seq=349/23809, ttl=10 (no response found…
538	11.501470	80.249.212.183	192.168.1.130	ICMP	134 Time-to-live exceeded (Time to live exceeded in transit)
539	11.502698	192.168.1.130	185.199.109.153	ICMP	106 Echo (ping) request id=0x0001, seq=350/24065, ttl=10 (no response found…
541	11.535785	80.249.212.183	192.168.1.130	ICMP	134 Time-to-live exceeded (Time to live exceeded in transit)
542	11.537359	192.168.1.130	185.199.109.153	ICMP	106 Echo (ping) request id=0x0001, seq=351/24321, ttl=10 (no response found…
543	11.570578	80.249.212.183	192.168.1.130	ICMP	134 Time-to-live exceeded (Time to live exceeded in transit)
653	12.543079	192.168.1.130	185.199.109.153	ICMP	106 Echo (ping) request id=0x0001, seq=352/24577, ttl=11 (reply in 654)
654	12.575455	185.199.109.153	192.168.1.130	ICMP	106 Echo (ping) reply id=0x0001, seq=352/24577, ttl=54 (request in 653)
655	12.577808	192.168.1.130	185.199.109.153	ICMP	106 Echo (ping) request id=0x0001, seq=353/24833, ttl=11 (reply in 656)
656	12.610434	185.199.109.153	192.168.1.130	ICMP	106 Echo (ping) reply id=0x0001, seq=353/24833, ttl=54 (request in 655)
657	12.612511	192.168.1.130	185.199.109.153	ICMP	106 Echo (ping) request id=0x0001, seq=354/25089, ttl=11 (reply in 658)
658	12.644810	185.199.109.153	192.168.1.130	ICMP	106 Echo (ping) reply id=0x0001, seq=354/25089, ttl=54 (request in 657)

3. **Чем отличаются ICMP-пакеты, генерируемые утилитой tracert, от ICMP-пакетов, генерируемых утилитой ping (см. предыдущее задание).**
   Размером блока данных, типами запросов.

4. **Чем отличаются полученные пакеты «ICMP reply» от «ICMP error» и зачем нужны оба этих типа ответов?**
   Отличаются кодом типом type, причиной их генерации, а также в ICMP reply есть поле data, совпадающие с request, а в ICMP error отсутствует, но может добавляться дополнительная информация об ошибке.

```
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: 92
Identification: 0x7a10 (31248)
```

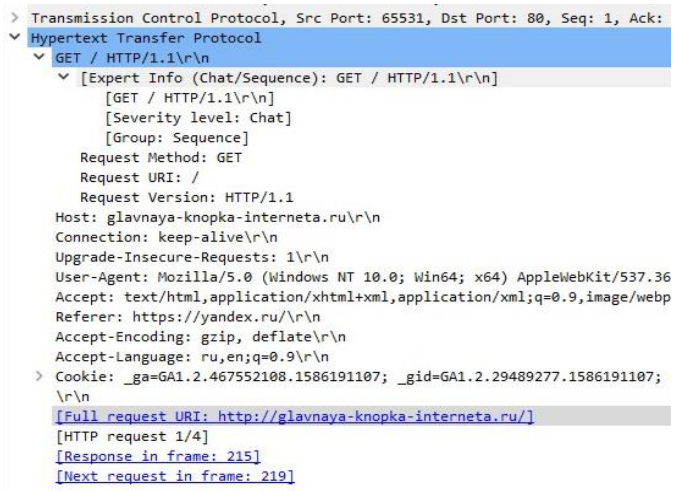
5. **Что изменится в работе tracert, если убрать ключ “-d”? Какой дополнительный трафик при этом будет генерироваться**
   Ключ -d предотвращает попытки команды tracert разрешения IP-адресов промежуточных маршрутизаторов в имена. Без этого ключа добавятся запросы к DNS-серверу.

59665	578.844197	89.223.47.1	192.168.1.130	ICMP	70 Time-to-live exceeded (Time to live exceeded in transit)
59666	578.845254	192.168.1.130	185.199.109.153	ICMP	106 Echo (ping) request id=0x0001, seq=210/53760, ttl=5 (no response found!)
59667	578.847242	89.223.47.1	192.168.1.130	ICMP	70 Time-to-live exceeded (Time to live exceeded in transit)
59668	578.848564	192.168.1.130	185.199.109.153	ICMP	106 Echo (ping) request id=0x0001, seq=211/54016, ttl=5 (no response found!)
59670	578.850339	89.223.47.1	192.168.1.130	ICMP	70 Time-to-live exceeded (Time to live exceeded in transit)
59674	578.851402	192.168.1.130	192.168.1.1	DNS	84 Standard query 0xc204 PTR 1.47.223.89.in-addr.arpa
59676	578.864566	192.168.1.1	192.168.1.130	DNS	183 Standard query response 0xc204 PTR 1.47.223.89.in-addr.arpa PTR lgw1.nev…

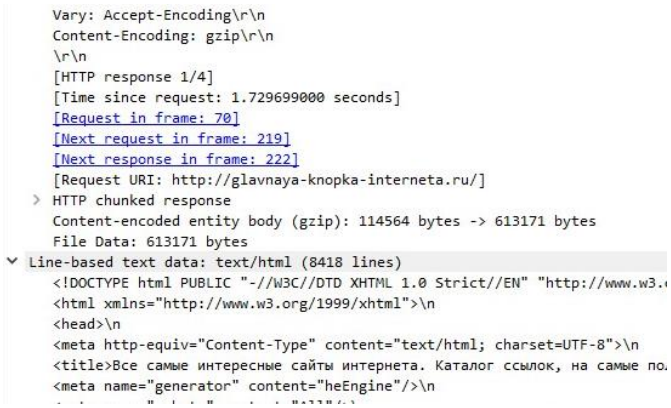
Анализ HTTP-трафика

При анализе трафика мною было выяснено, что при обращении к сайту происходит посылка http запросов на удаленный ресурс, который в свою очередь присылает ответ (в моем случае с состоянием OK и необходимыми данными).

Запрос



Ответ

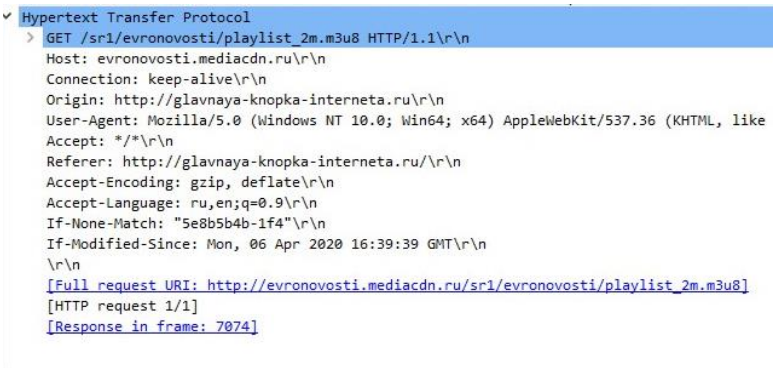


При первом обращении:

браузер посылает get-запрос

сервер присылает данные: html код страницы, картинки, стили.

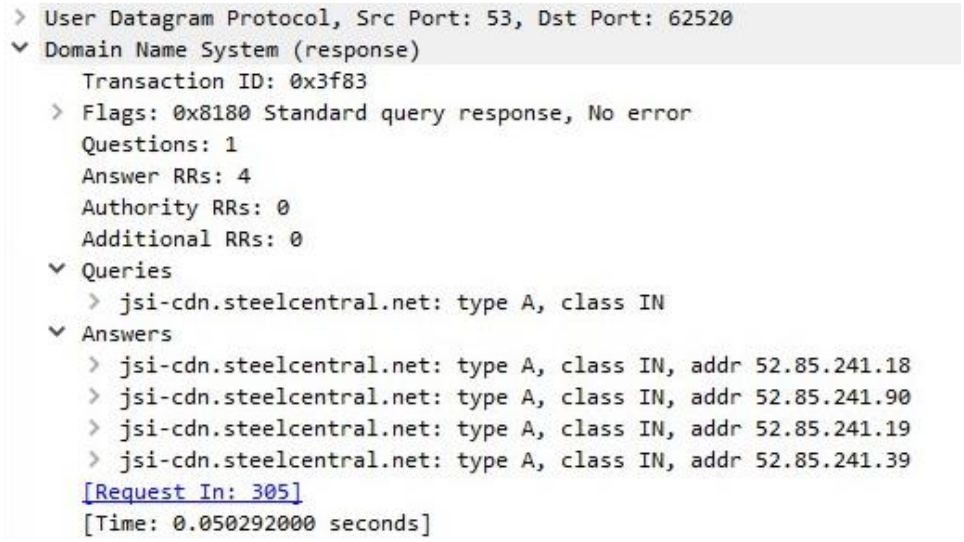
Повторный запрос:



Повторный ответ:

При повторном обращении:
браузер посылает get-запрос с заголовком if-Modified-Since
сервер на основании этого заголовка определяет, что информацию обновлять не нужно и присылает ответ со статусом 304 Not Modified.

Анализ DNS-трафика



В DNS трафике, как и в http, появился заголовок транспортного уровня, который используется при обмене датаграммами.

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

DNS-запрос отправляется на DNS сервер, с целью узнать IP-адрес нужного сайта.

2. **Какие бывают типы DNS-запросов?**

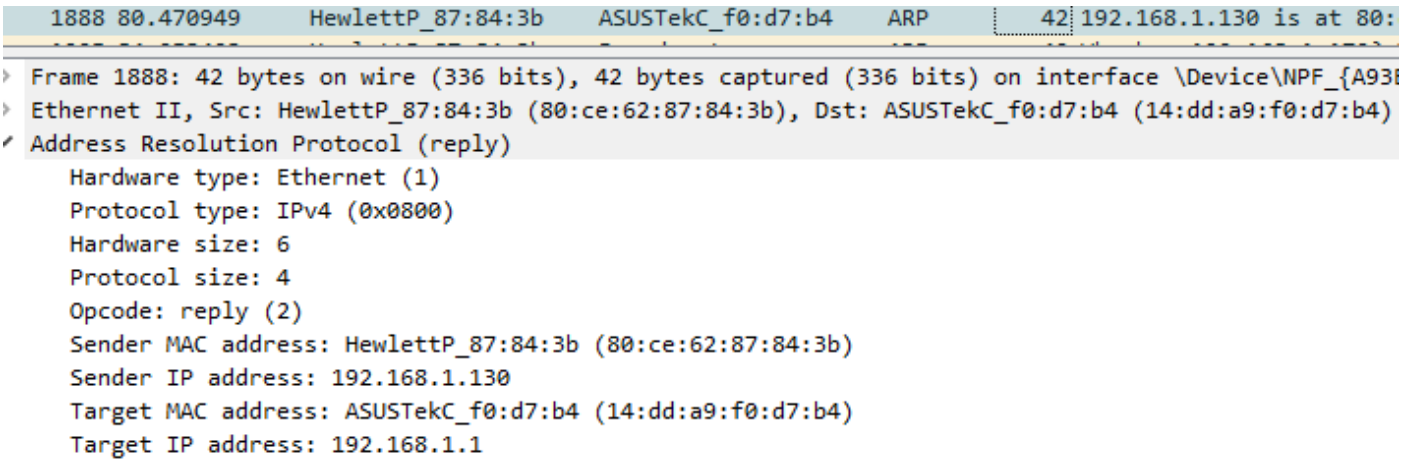
- Рекурсивный запрос. Отправляет доменное имя DNS серверу, просит вернуть либо IP адрес этого домена, либо имя DNS сервера, на котором можно получить IP адрес или следующее имя DNS сервера.
- Итеративный запрос. Отправляет доменное имя DNS серверу, просит вернуть либо IP адрес этого домена, либо имя DNS сервера, к которому можно обратиться.

- Обратный запрос. Посылает IP, просит вернуть доменное имя

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

Когда изображения расположены на ресурсах с неизвестным IP адресом.

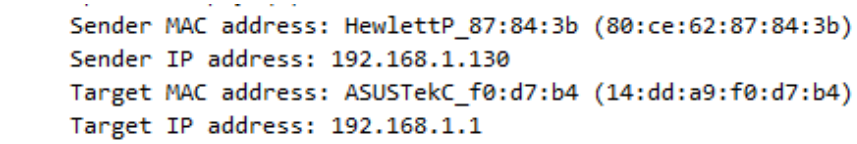
Анализ ARP-трафика



При обращении к сайту происходила посылка ARP запросов. Основная цель данного протокола – определение MAC-адреса по IP-адресу компьютера.

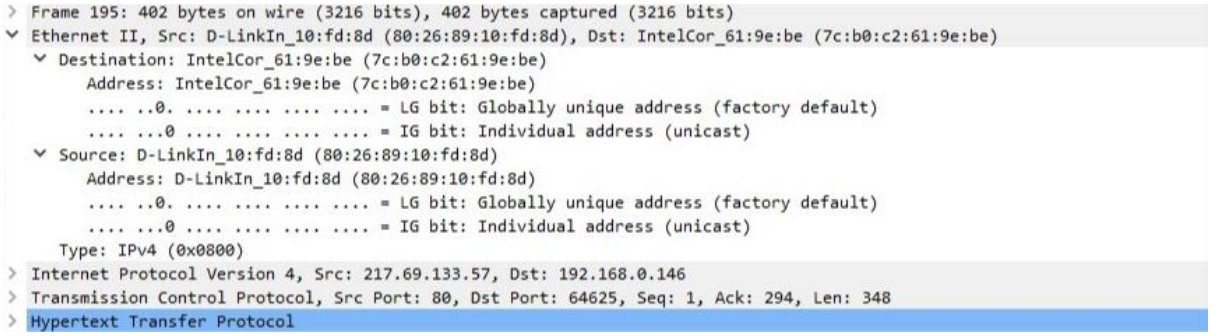
1. **Какие MAC-адреса присутствуют в захваченных пакетах ARP-протокола? Что означают эти адреса? Какие устройства они идентифицируют?**

Адреса роутера (14:dd:a9:f0:d7:b4) и машины(80:ce:62:87:84:3b), с которой отправляется запрос. Они означают уникальные физические адреса устройств.



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

Эти адреса идентифицируют отправителя и получателя ресурса. В данном случае это роутер и ноутбук.



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

Это требуется для корректного динамического обновления arp таблицы, которая хранит IP, MAC-адрес и тип записи. Если в таблице для нужного IP отсутствует MAC-адрес, то в бродкаст будет отправлен запрос, ответит только обладатель этого IP и в ответ отправит свой MAC адрес, который будет занесён в ARP-таблицу.

Анализ трафика утилиты nslookup

`nslookup`

No.	Time	Source	Destination	Protocol	Length	Info
112	10.012146	192.168.1.130	192.168.1.1	DNS	77	Standard query 0x4cef A allacee.github.io
114	10.029086	192.168.1.1	192.168.1.130	DNS	444	Standard query response 0x4cef A allacee.github.io A 185.199.108.153 A 185.199.109.…
115	10.035964	192.168.1.130	185.199.108.153	DNS	88	Standard query 0x0001 PTR 153.108.199.185.in-addr.arpa
125	12.067040	192.168.1.130	185.199.108.153	DNS	68	Standard query 0x0002 A nslookup
134	14.069612	192.168.1.130	185.199.108.153	DNS	68	Standard query 0x0003 AAAA nslookup
224	27.145824	192.168.1.130	192.168.1.1	DNS	84	Standard query 0x0001 PTR 1.1.168.192.in-addr.arpa
225	27.147084	192.168.1.1	192.168.1.130	DNS	113	Standard query response 0x0001 PTR 1.1.168.192.in-addr.arpa PTR router.asus.com
226	27.148592	192.168.1.130	192.168.1.1	DNS	66	Standard query 0x0002 A vk.com
227	27.151952	192.168.1.1	192.168.1.130	DNS	422	Standard query response 0x0002 A vk.com A 87.240.190.67 A 87.240.190.72 A 87.240.19…
228	27.154731	192.168.1.130	192.168.1.1	DNS	66	Standard query 0x0003 AAAA vk.com
229	27.157534	192.168.1.1	192.168.1.130	DNS	122	Standard query response 0x0003 AAAA vk.com SOA ns1.vkontakte.ru
419	47.192032	192.168.1.130	192.168.1.1	DNS	89	Standard query 0x3cd7 A d3cv4a9a9wh0bt.cloudfront.net
420	47.195176	192.168.1.1	192.168.1.130	DNS	466	Standard query response 0x3cd7 A d3cv4a9a9wh0bt.cloudfront.net A 13.32.42.196 A 13.…
461	50.755493	192.168.1.130	192.168.1.1	DNS	87	Standard query 0x06d8 A roaming.officeapps.live.com
462	50.759035	192.168.1.1	192.168.1.130	DNS	455	Standard query response 0x06d8 A roaming.officeapps.live.com CNAME prod.roaming1.li…
529	52.701356	192.168.1.130	192.168.1.1	DNS	97	Standard query 0xc3f3 A array507.prod.do.dsp.mp.microsoft.com
530	52.704665	192.168.1.1	192.168.1.130	DNS	311	Standard query response 0xc3f3 A array507.prod.do.dsp.mp.microsoft.com A 40.79.67.1…

`nslookup -type=NS`

22	1.462919	192.168.1.130	192.168.1.1	DNS	84	Standard query 0x0001 PTR 1.1.168.192.in-addr.arpa
23	1.464142	192.168.1.1	192.168.1.130	DNS	113	Standard query response 0x0001 PTR 1.1.168.192.in-addr.arpa PTR router.asus.com
24	1.465768	192.168.1.130	192.168.1.1	DNS	66	Standard query 0x0002 NS vk.com
25	1.468772	192.168.1.1	192.168.1.130	DNS	242	Standard query response 0x0002 NS vk.com NS ns1.vkontakte.ru NS ns2.vkontakte.ru NS…
66	8.191930	192.168.1.130	192.168.1.1	DNS	66	Standard query 0x86ca A vk.com
67	8.193084	192.168.1.1	192.168.1.130	DNS	162	Standard query response 0x86ca A vk.com A 93.186.225.208 A 87.240.139.194 A 87.240.…

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

При посылке в п.2 мы получаем 3 пары DNS вопрос-ответ, а при использовании ключа -type=NS только 2. При этом отличие в типе запроса (NS - указывает на имя сервера, на котором расположены данные, A – адрес ipv4, AAAA - адрес ipv6)

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

```
Answers
  d3cv4a9a9wh0bt.cloudfront.net: type A, class IN, addr 13.32.42.196
  d3cv4a9a9wh0bt.cloudfront.net: type A, class IN, addr 13.32.42.15
  d3cv4a9a9wh0bt.cloudfront.net: type A, class IN, addr 13.32.42.74
  d3cv4a9a9wh0bt.cloudfront.net: type A, class IN, addr 13.32.42.133
Answers
  vk.com: type A, class IN, addr 93.186.225.208
  vk.com: type A, class IN, addr 87.240.139.194
  vk.com: type A, class IN, addr 87.240.137.158
  vk.com: type A, class IN, addr 87.240.190.78
  vk.com: type A, class IN, addr 87.240.190.72
  vk.com: type A, class IN, addr 87.240.190.67
[Request In: 66]
```

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

```
Authoritative nameservers
  d3cv4a9a9wh0bt.cloudfront.net: type NS, class IN, ns ns-1859.awsdns-40.co.uk
  d3cv4a9a9wh0bt.cloudfront.net: type NS, class IN, ns ns-189.awsdns-23.com
  d3cv4a9a9wh0bt.cloudfront.net: type NS, class IN, ns ns-574.awsdns-07.net
  d3cv4a9a9wh0bt.cloudfront.net: type NS, class IN, ns ns-1438.awsdns-51.org
```

Анализ FTP-трафика

Для исследования выбран сайт ftp://ftp.novaworld.com/

```
    54 3.423243    207.178.209.208   192.168.1.130     FTP      81 Response: 220 Microsoft FTP Service
    55 3.423675    192.168.1.130     207.178.209.208   FTP      70 Request: USER anonymous
    56 3.613655    207.178.209.208   192.168.1.130     FTP     126 Response: 331 Anonymous access allowed, send identity (e-mail name) as password.
    57 3.614090    192.168.1.130     207.178.209.208   FTP      80 Request: PASS mozilla@example.com
    59 3.804110    207.178.209.208   192.168.1.130     FTP      85 Response: 230 Anonymous user logged in.
    60 3.804608    192.168.1.130     207.178.209.208   FTP      60 Request: SYST
    67 3.994550    207.178.209.208   192.168.1.130     FTP      70 Response: 215 Windows_NT
    68 3.994908    192.168.1.130     207.178.209.208   FTP      60 Request: FEAT
    74 4.184802    207.178.209.208   192.168.1.130     FTP      64 Response: 211-FEAT
    76 4.417624    207.178.209.208   192.168.1.130     FTP      83 Response:     SIZE
    77 4.418307    192.168.1.130     207.178.209.208   FTP      59 Request: PWD
    80 4.607781    207.178.209.208   192.168.1.130     FTP      85 Response: 257 "/" is current directory.
    81 4.608203    192.168.1.130     207.178.209.208   FTP      62 Request: TYPE I
    82 4.797908    207.178.209.208   192.168.1.130     FTP      74 Response: 200 Type set to I.
    83 4.798425    192.168.1.130     207.178.209.208   FTP      60 Request: PASV
    84 4.989212    207.178.209.208   192.168.1.130     FTP     107 Response: 227 Entering Passive Mode (207,178,209,208,10,173).
    85 4.990969    192.168.1.130     207.178.209.208   FTP      61 Request: CWD /
    93 5.180755    207.178.209.208   192.168.1.130     FTP      83 Response: 250 CWD command successful.
    94 5.181144    192.168.1.130     207.178.209.208   FTP      60 Request: LIST
   101 5.371463    207.178.209.208   192.168.1.130     FTP     108 Response: 125 Data connection already open; Transfer starting.
   102 5.372828    207.178.209.208   192.168.1.130     FTP-DA… 301 FTP Data: 247 bytes (PASV) (CWD /)
   111 5.564927    207.178.209.208   192.168.1.130     FTP      78 Response: 226 Transfer complete.
```

```
> Frame 55: 70 bytes on wire (560 bits), 70 bytes captured (560 bits) on interface \Device\NPF_{A93EE18D-B45B-48F5-A226-6AEBAC5778C7}, id 0
> Ethernet II, Src: HewlettP_87:84:3b (80:ce:62:87:84:3b), Dst: ASUSTekC_f0:d7:b4 (14:dd:a9:f0:d7:b4)
> Internet Protocol Version 4, Src: 192.168.1.130, Dst: 207.178.209.208
> Transmission Control Protocol, Src Port: 55598, Dst Port: 21, Seq: 1, Ack: 28, Len: 16
v File Transfer Protocol (FTP)
  v USER anonymous\r\n
      Request command: USER
      Request arg: anonymous
  [Current working directory: ]
```

Добавлен заголовок FTP – протокол прикладного уровня.

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

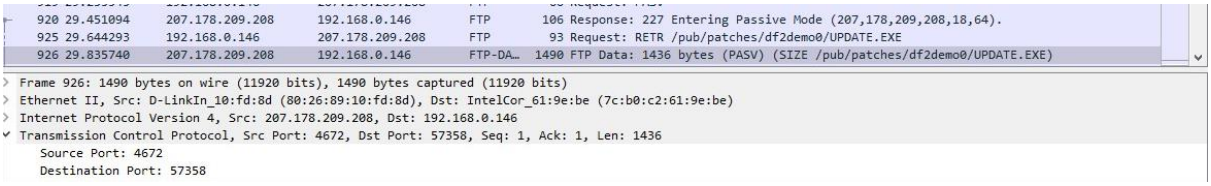
```
Transmission Control Protocol, Src Port: 2733, Dst Port: 55601, Seq: 1, Ack: 1, Len: 247
FTP Data (247 bytes data)
[Setup frame: 84]
[Setup method: PASV]
[Command: CWD /]
Command frame: 85
[Current working directory: /]
Line-based text data (5 lines)
    03-29-07  05:43AM                    0 1175172184.tst\r\n
    06-01-06  11:28AM                 8192 LOAD.TST\r\n
    04-19-06  11:13AM       <DIR>          pub\r\n
    01-13-15  02:21AM                   21 SERVER.LST\r\n
```

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

```
   918 29.255301    207.178.209.208   192.168.0.146     FTP     125 Response: 550 /pub/patches/df2demo0/UPDATE.EXE: The directory name is invalid.
   919 29.255545    192.168.0.146     207.178.209.208   FTP      60 Request: PASV
   920 29.451094    207.178.209.208   192.168.0.146     FTP     106 Response: 227 Entering Passive Mode (207,178,209,208,18,64)
```

```
> Frame 919: 60 bytes on wire (480 bits), 60 bytes captured (480 bits)
> Ethernet II, Src: IntelCor_61:9e:be (7c:b0:c2:61:9e:be), Dst: D-LinkIn_10:fd:8d (80:26:89:10:fd:8d)
> Internet Protocol Version 4, Src: 192.168.0.146, Dst: 207.178.209.208
v Transmission Control Protocol, Src Port: 57357, Dst Port: 21, Seq: 138, Ack: 280, Len: 6
    Source Port: 57357
    Destination Port: 21
    [Stream index: 30]
```

Для FTP – DATA порт находится из ответа сервера на запрос PASV, приходит 6 чисел, предпоследние * 256 + последняя.



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

- FTP для управления соединением (важной особенностью можно выделить то, что сервер присылает ip адрес и порт в ответе для клиента на запрос PASV – переход в пассивный режим)
- FTP-DATA для передачи данных, преобразованных в определенный тип, например бинарный.

Анализ Skype-трафика

1. **Чем различаются пакета разных видов Skype-трафика (текст, аудио, видео)?**
   Для передачи текстовых данных Скайп использует протокол TCP.
   Для обеспечения безопасности при передаче по сети на транспортном уровне используется криптографический протокол TLS.
   Для передачи медиаинформации используется протокол. Не обеспечивает контроль доставки, чтобы была возможность передать большой поток данных
2. **Какой Wireshark-фильтр следует использовать для независимой идентификации Skype-трафика разных видов (текст, аудио, видео)?**
   Текст: ip.addr == 192.168.1.130 and (tcp or tls) and (tcp.port == 64958)

   Видео: ip.addr == 192.168.1.130 and (udp) and (udp.port == 32132)

Вывод

В результате выполнения работы мной были исследованы на практике различные протоколы передачи данных – выявлены особенности каждого протокола, а также сферы применения и изучены инструменты для анализа сети (nslookup, tracrert).