Федеральное государственное автономное образовательное учреждение высшего образования

Университет ИТМО

Дисциплина: Компьютерные сети

Лабораторная работа 5

Выполнили:
Кривоносов Егор Дмитриевич
Марков Петр Денисович

Группа: P33111

Преподаватель:
Тропченко Андрей Александрович

2022 г.

Санкт-Петербург

Цель работы

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

Задание

1. Запустить Wireshark. В появившемся окне выбрать интерфейс, для которого необходимо осуществлять анализ проходящих через него пакетов. В качестве интерфейса, используемого для захвата трафика, выбрать физический адаптер, через который компьютер подключён к Интернету (обычно этот адаптер называется Local или “Подключение по локальной сети”). Если меню для выбора адаптера не появляется при запуске Wireshark, нужно запустить из “Меню” команду “Capture->Options”. После выбора адаптеру, нужно запустить процесс захвата трафика (кнопка Start).
2. Инициировать процесс передачи трафика по сети (например, в браузере открыть сайт, заданный по варианту, или запустить соответствующую сетевую утилиту).
3. Установить значение “Фильтра”, чтобы из всего множества перехватываемых пакетов Wireshark отобразил только те, которые имеют отношение к выполняемому заданию. Для корректного создания фильтра следует пользоваться всплывающими подсказками Wireshark, которые активизируются при наборе фильтра. В качестве альтернативного способа можно использовать интерактивный конструктор фильтра, нажав на кнопку “Expression” в правой части элемента “Фильтр”
4. Дождаться появления данных в списке захваченных пакетов и убедиться, что количество пакетов достаточно для выполнения задания.
5. Сохранить захваченный трафик в файл-трассу (pcap).
6. Описать в отчёте структуру наблюдаемых PDU (т.е. протокольных блоков данных: кадров, пакетов, сегментов) как для запросов, так и ответов. Указать название и назначение всех заголовков всех уровней OSI-модели в пакетах с учётом порядка инкапсуляции (для этого нужно раскрывать соответствующие значки «+» в поле с детальной информацией о выбранном пакете).
7. Написать в отчёте ответы на вопросы задания (для этого может потребоваться самостоятельно изучить назначение соответствующей заданию сетевой утилиты, использованной для создания трафика).
8. Поместить в отчёт скриншоты окна Wireshark, иллюстрирующие ответы из вышеуказанных п.6 и п.7.

URL - Адрес сайта, в названии которого лексически входит фамилия студента: https://krivonosov.ru/

Выполнение

Этап 1. Анализ трафика утилиты ping

Необходимо отследить и проанализировать трафик, создаваемый утилитой ping, запустив её следующим образом из командной строки:

“ping -l размер_пакета адрес_сайта_по_варианту”.

В качестве “размера_пакета” необходимо поочерёдно использовать различные значения от 100 до 10000, самостоятельно выбрав шаг изменения.

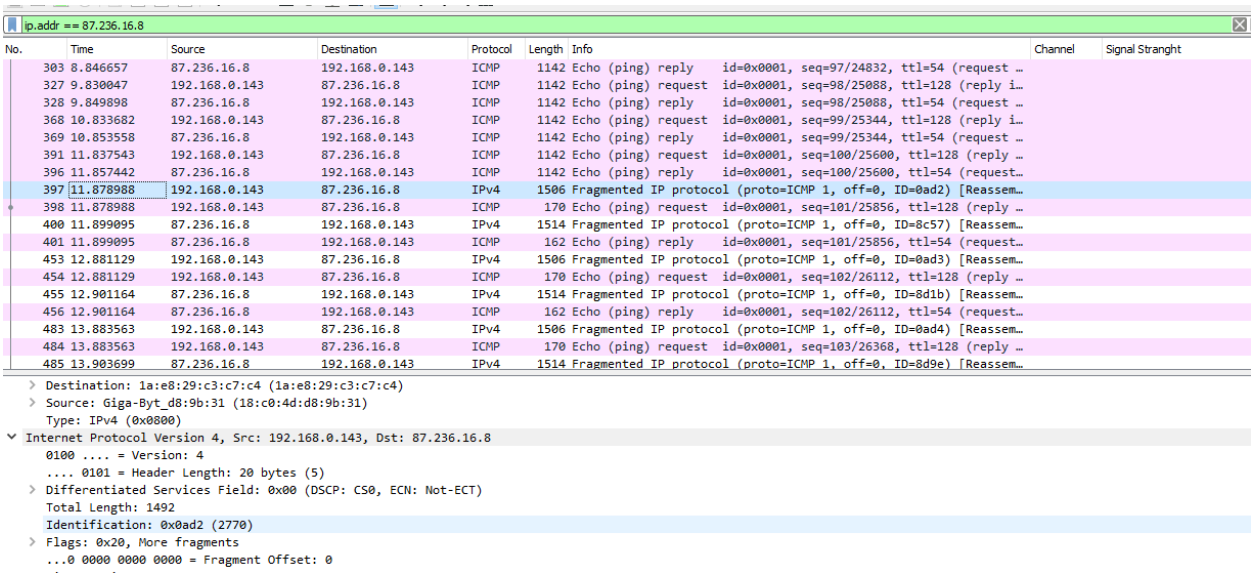
for /L %i in (100, 500, 10000) do ping -l %i krivonosov.ru

```
C:\Users\Egor>ping -l 6600 krivonosov.ru

Обмен пакетами с krivonosov.ru [87.236.16.8] с 6600 байтами данных:
Ответ от 87.236.16.8: число байт=6600 время=20мс TTL=54
Ответ от 87.236.16.8: число байт=6600 время=20мс TTL=54
Ответ от 87.236.16.8: число байт=6600 время=20мс TTL=54
Ответ от 87.236.16.8: число байт=6600 время=20мс TTL=54

Статистика Ping для 87.236.16.8:
    Пакетов: отправлено = 4, получено = 4, потеряно = 0
    (0% потерь)
Приблизительное время приема-передачи в мс:
    Минимальное = 20мсек, Максимальное = 20 мсек, Среднее = 20 мсек
```

Вопросы

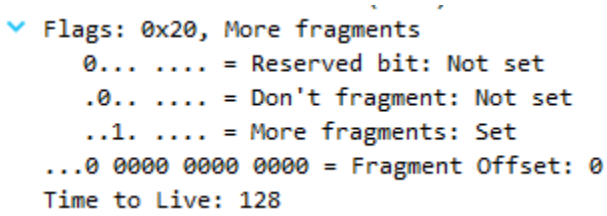
1. **Имеет ли место фрагментация исходного пакета, какое поле на это указывает?**

   Да, так как с увеличением размера пакета, мы можем видеть, что на уровне IP-протокола появляются фрагменты “Fragmented IP protocol”. Флаг More Fragments как раз указывает на наличие фрагментации исходного пакета.



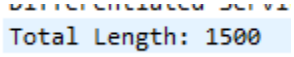
2. **Какая информация указывает, является ли фрагмент пакета последним или промежуточным?**

   Установленный бит MF (More Fragments) говорит о том, что данный пакет является промежуточным (не последним) фрагментом. Соответственно, если флаг MF не установлен, то пакет является последним.



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

   Количество фрагментов зависит от размера пакета и соответствующего соединению MTU (Maximum Transmission Unit – максимальный размер передаваемого блока), который обычно равен 1500 байт.

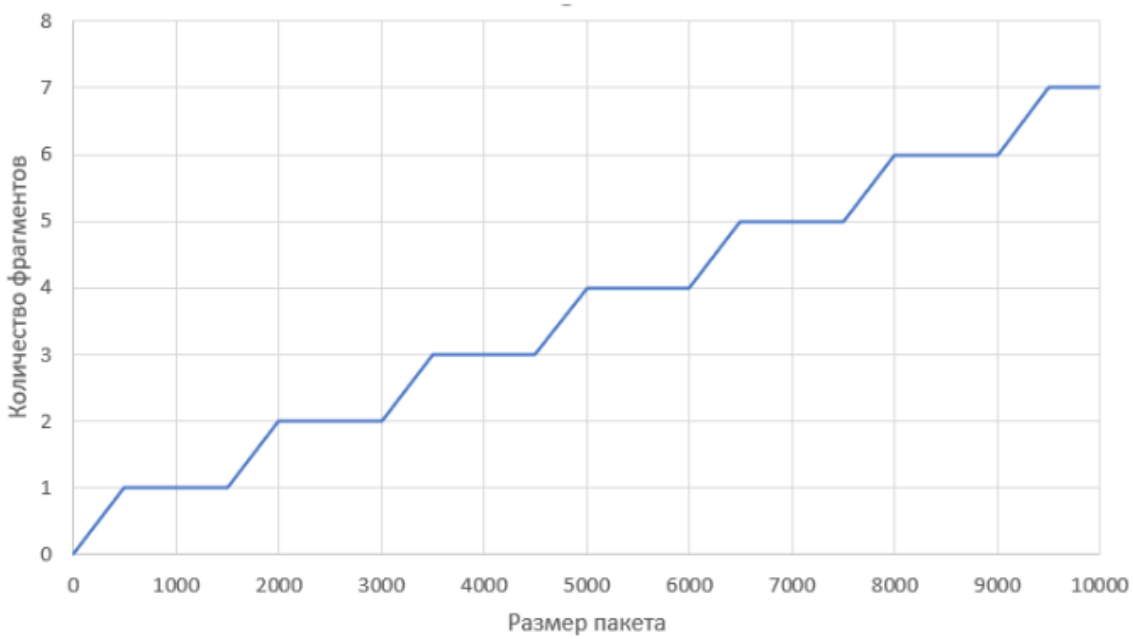


   Количество фрагментов = ceil(Длина сообщения (байт) / 1500 байт (фрагмент))

   ceil() - округление вверх.

4. **Построить график, в котором на оси абсцисс находится размер пакета, а по оси ординат – количество фрагментов, на которое был разделён каждый**

ping-пакет.

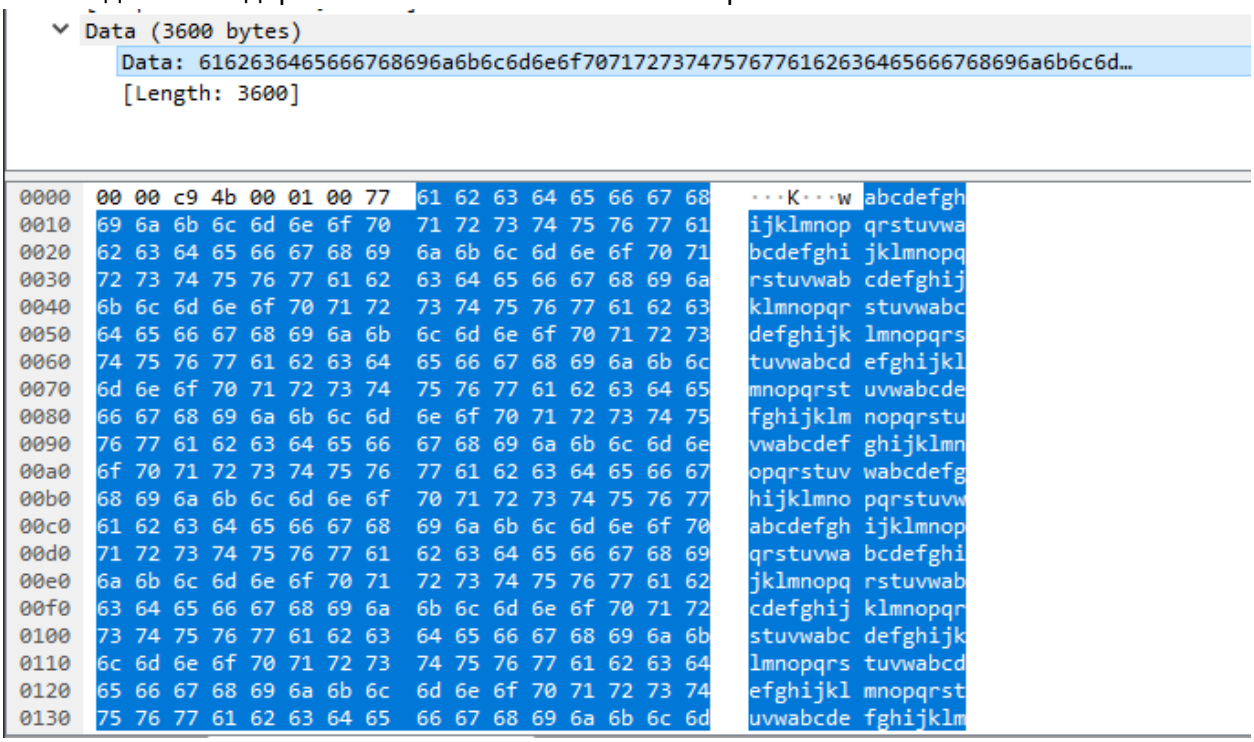


5. **Как изменить поле TTL с помощью утилиты ping?**

   Для изменения TTL нужно добавить ключ -i, его аргументом является срок жизни пакета в миллисекундах. (ping -i 15 krivonosov.ru)

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

В поле данных содержатся символы английского алфавита.



Этап 2. Анализ трафика утилиты tracert (traceroute)

Необходимо отследить и проанализировать трафик, создаваемый утилитой tracert (или traceroute в Linux), запустив её следующим образом из командной строки:

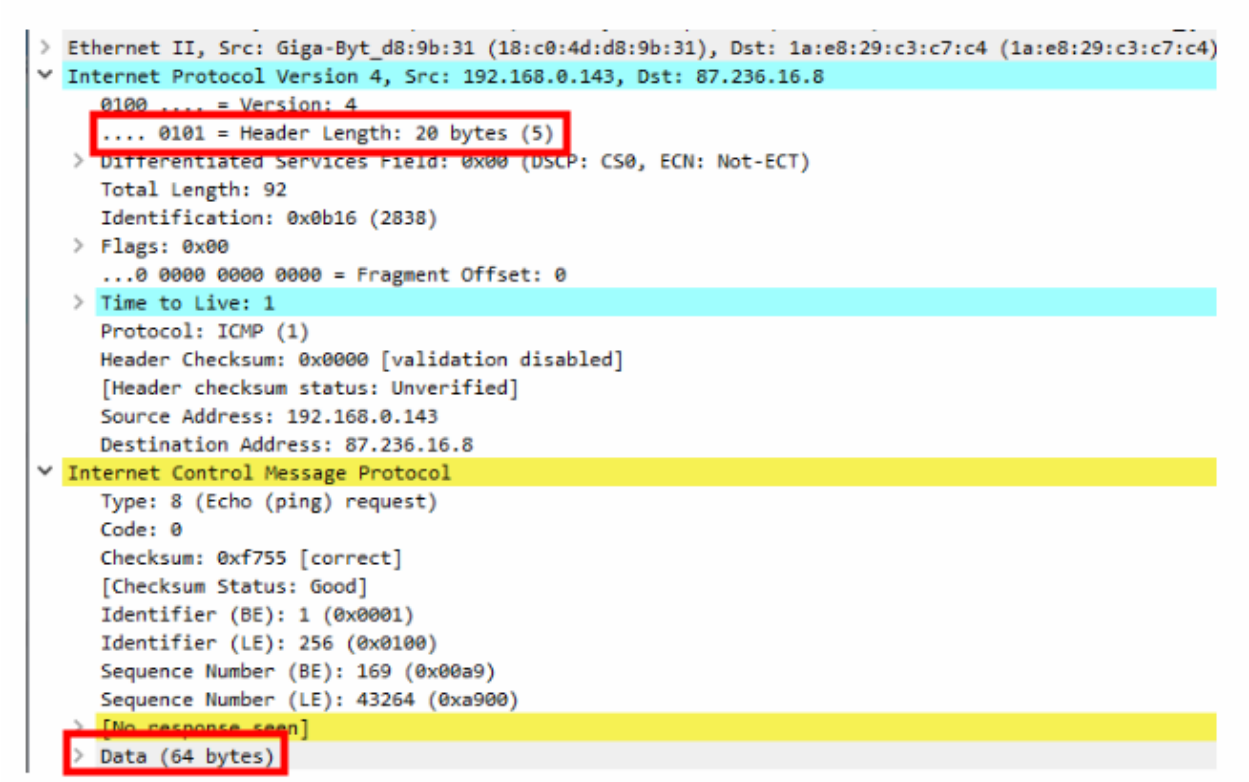
“tracert -d адрес_сайта_по_варианту”

```
C:\Users\Egor>tracert -d krivonosov.ru

Трассировка маршрута к krivonosov.ru [87.236.16.8]
с максимальным числом прыжков 30:

  1    <1 мс    <1 мс    <1 мс  192.168.0.1
  2    <1 мс    <1 мс    <1 мс  185.102.9.161
  3    <1 мс    <1 мс    <1 мс  172.29.192.136
  4    <1 мс    <1 мс    <1 мс  172.29.193.253
  5    <1 мс    <1 мс    <1 мс  172.29.194.177
  6    <1 мс    <1 мс    <1 мс  172.29.194.167
  7     1 ms    <1 мс    <1 мс  172.29.194.184
  8     1 ms    <1 мс    <1 мс  85.114.1.12
  9     1 ms     1 ms     1 ms  85.114.3.53
 10     1 ms     1 ms     1 ms  85.114.3.56
 11     4 ms     1 ms     1 ms  87.245.250.65
 12    11 ms    17 ms    11 ms  87.245.233.89
 13    10 ms    10 ms     9 ms  87.245.229.69
 14     *        *        *     Превышен интервал ожидания для запроса.
 15    19 ms    19 ms    19 ms  87.236.16.8

Трассировка завершена.
```

Вопросы

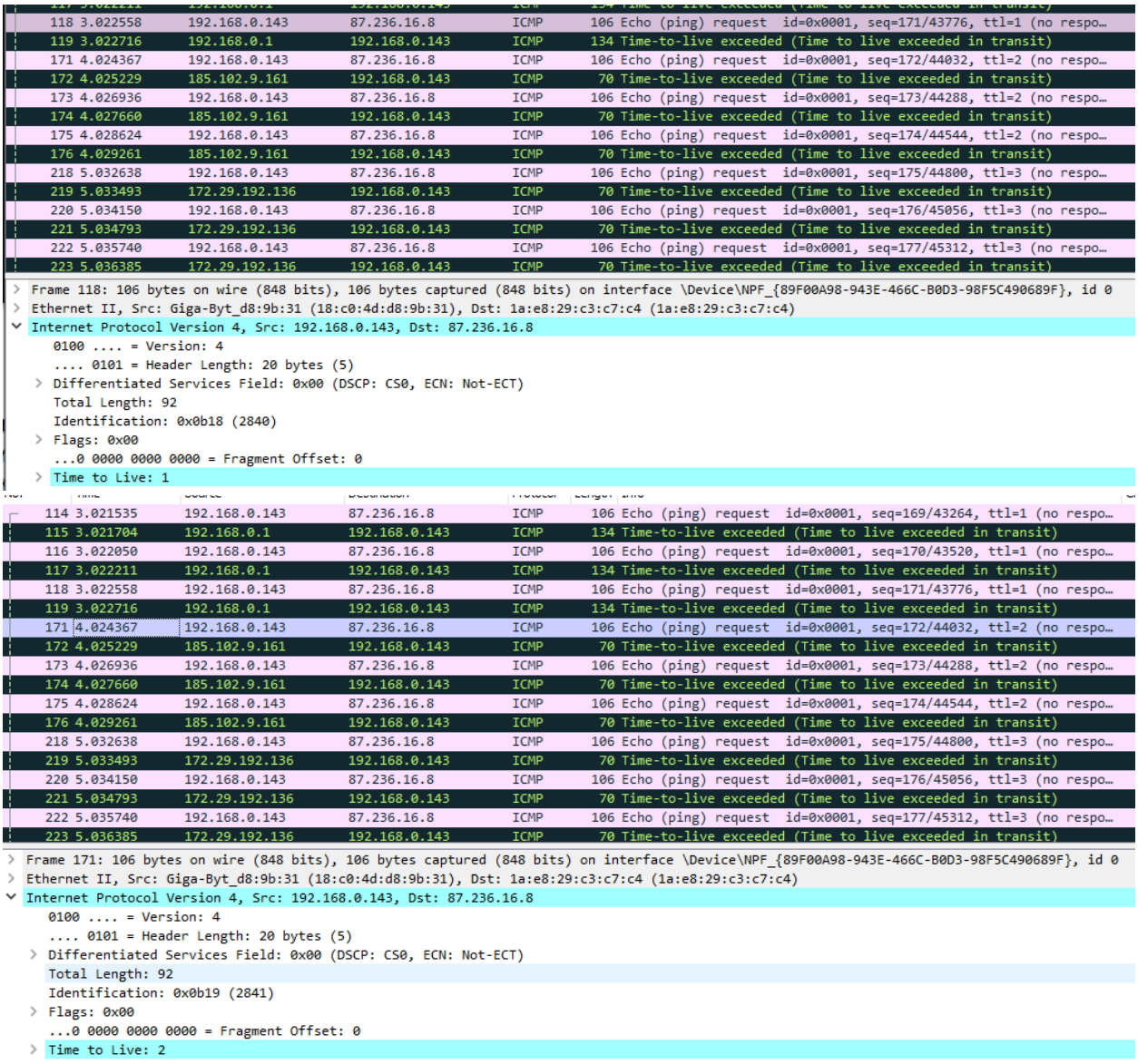
1. **Сколько байт содержится в заголовке IP? Сколько байт содержится в поле данных?**

   В заголовке - 20 байт.
   В поле данных - 64 байта.



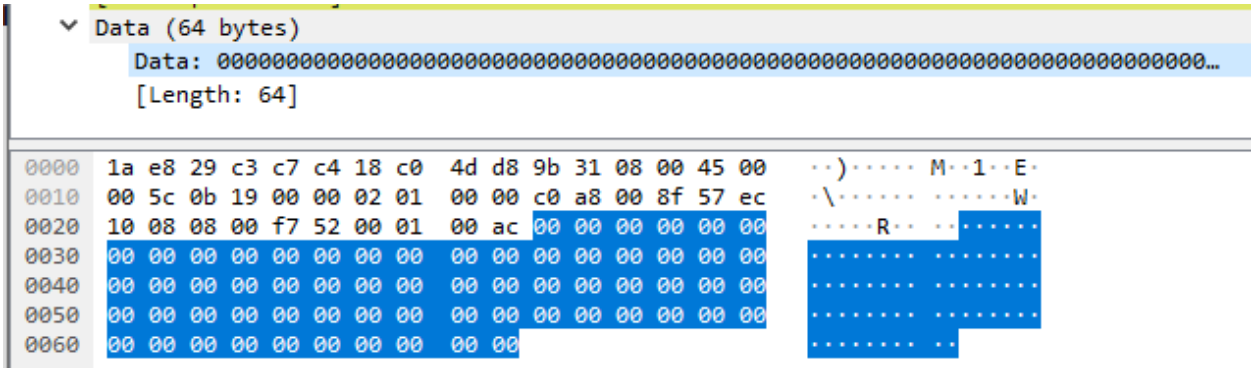
2. **Как и почему изменяется поле TTL в следующих друг за другом ICMP-пакетах tracert?**

   Утилита tracert отправляет первый пакет 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. Поскольку каждый маршрутизатор на пути обязан уменьшить значение поля TTL пакета, по крайней мере на 1 перед дальнейшей пересылкой пакета, значение TTL по сути является эффективным счетчиком переходов.



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

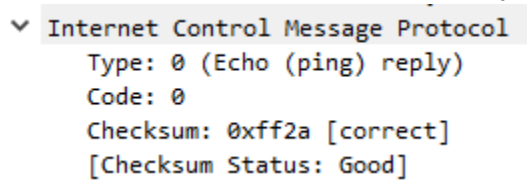
   В отличие от пакетов, генерируемых утилитой ping, пакеты, генерируемые утилитой tracert, в поле данных содержат нули.



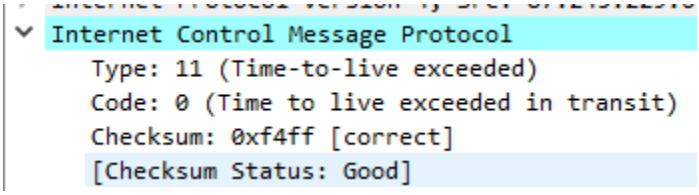
4. **Чем отличаются полученные пакеты «ICMP reply» от «ICMP error» и зачем нужны оба этих типа ответов?**

   Пакеты «ICMP reply» указывают на получение ответного сообщения.
   Пакеты «ICMP error» указывают на то, что произошла ошибка.
   Они используются чтобы различать причину истечения TTL.

   Значение поля TYPE = 0 соответствует сообщению Echo Reply



   Значение поля TYPE = 11 соответствует сообщению Time Exceeded (истекло время)



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, то есть в выводе будут отсутствовать имена хостов, через которые проходит IP-пакет.

Этап 3. Анализ HTTP-трафика

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):

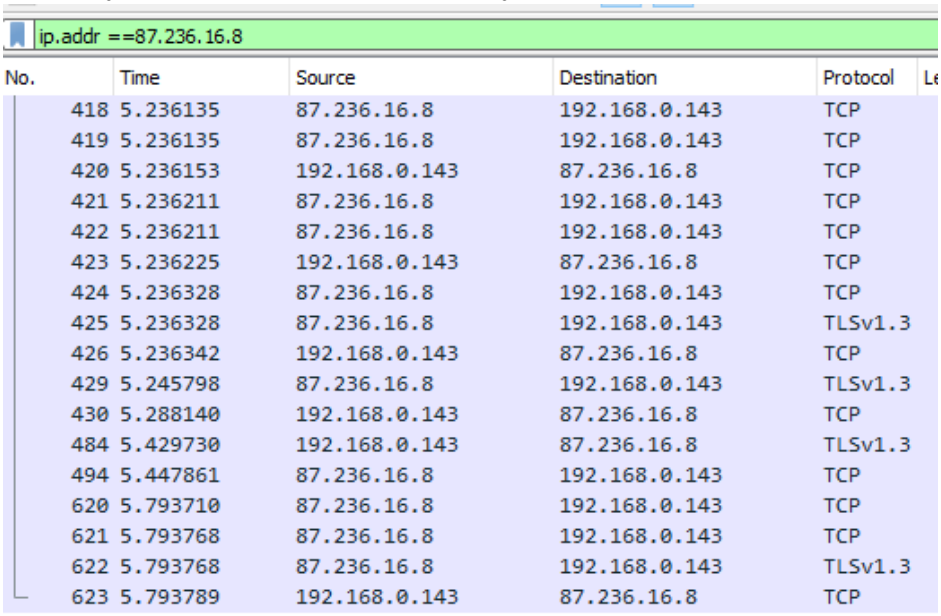
- GET-сообщение от клиента (браузера);
- ответ сервера.

Для этого в поле с детальной информацией о пакете нужно развернуть строку “HTTP”. Затем необходимо обновить страницу в браузере так, чтобы вместо «HTTP GET» был сгенерирован «HTTP CONDITIONAL GET» (так называемый «условный GET»). Условные запросы GET содержат поля If-Modified-Since, If-Match, If-Range и подобные, которые позволяют при повторном запросе не передавать редко изменяемые данные. В ответ на условный GET тело запрашиваемого ресурса передается только в том случае, если этот ресурс изменялся после даты «If-Modified- Since». Если ресурс не изменялся, сервер вернет код статуса «304 Not Modified».

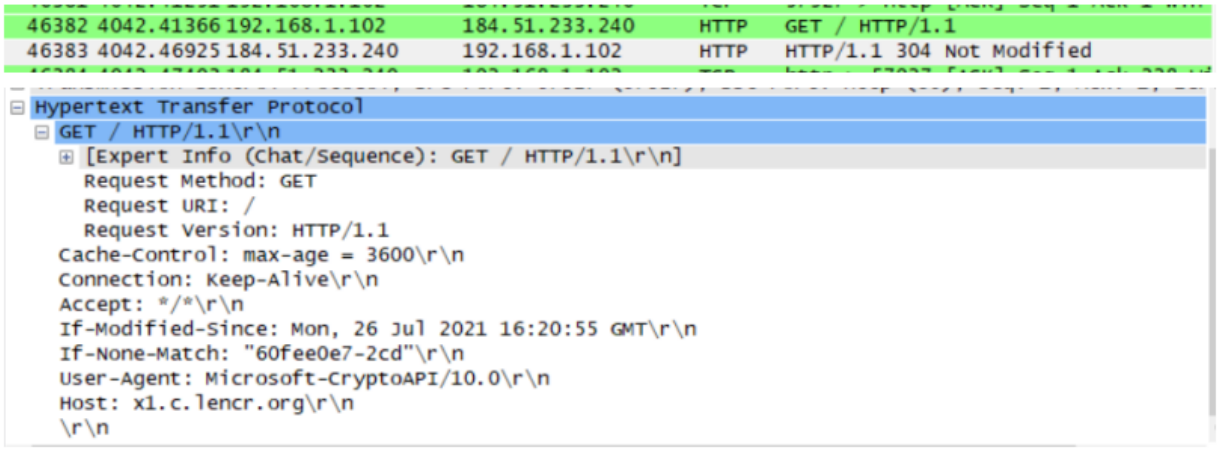
Вопросы

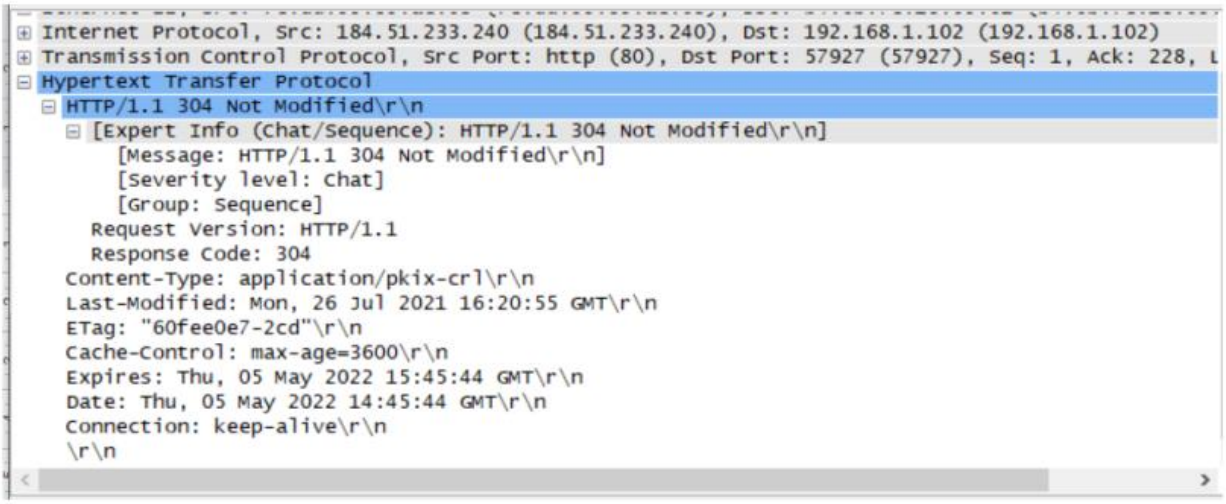
По результатам анализа собранной трассы покажите, каким образом протокол HTTP передавал содержимое страницы при первичном посещении страницы и при вторичном запросе-обновлении от браузера (т.е. при различных видах GET-запросов).

Так как сайт, выбранный в нашем варианте, имеет динамическое содержимое, как и большинство современных сайтов, использующих реактивные технологии, то заголовка not modified там быть не может (только статические страницы/файлы могут его иметь). Также HTTPS имеет сайт из-за этого HTML поля нет т.к. передача происходит по TCP. Поэтому рассмотрим на примере другого сайта.



No.	Time	Source	Destination	Protocol	Le
418	5.236135	87.236.16.8	192.168.0.143	TCP	
419	5.236135	87.236.16.8	192.168.0.143	TCP	
420	5.236153	192.168.0.143	87.236.16.8	TCP	
421	5.236211	87.236.16.8	192.168.0.143	TCP	
422	5.236211	87.236.16.8	192.168.0.143	TCP	
423	5.236225	192.168.0.143	87.236.16.8	TCP	
424	5.236328	87.236.16.8	192.168.0.143	TCP	
425	5.236328	87.236.16.8	192.168.0.143	TLSv1.3	
426	5.236342	192.168.0.143	87.236.16.8	TCP	
429	5.245798	87.236.16.8	192.168.0.143	TLSv1.3	
430	5.288140	192.168.0.143	87.236.16.8	TCP	
484	5.429730	192.168.0.143	87.236.16.8	TLSv1.3	
494	5.447861	87.236.16.8	192.168.0.143	TCP	
620	5.793710	87.236.16.8	192.168.0.143	TCP	
621	5.793768	87.236.16.8	192.168.0.143	TCP	
622	5.793768	87.236.16.8	192.168.0.143	TLSv1.3	
623	5.793789	192.168.0.143	87.236.16.8	TCP	





Заголовок «HTTP/1.1 304 Not Modified» означает, что не нужно отправлять тело вместе с ним (таким образом экономится пропускная способность). Как видно в 1 запросе, установлен Cache-Control (в секундах), соответственно содержимое страницы кэшируется и нет необходимости отправлять его снова, если изменений с последней загрузки не произошло.

Этап 4. Анализ DNS-трафика

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

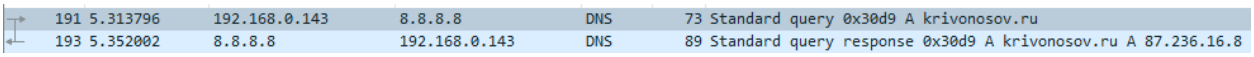
- настроить Wireshark-фильтр: “ip.addr == ваш_IP_адрес”;
- очистить кэш DNS с помощью команды ipconfig в командной строке: ipconfig /flushdns
- очистить кэш браузера;
- зайти на Интернет-сайт, заданный по варианту.

```
C:\Users\Egor>ipconfig /flushdns

Настройка протокола IP для Windows

Кэш сопоставителя DNS успешно очищен.
```

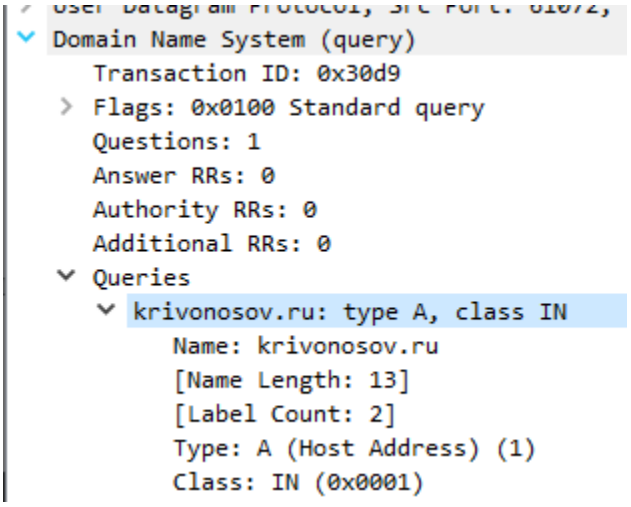
Вопросы

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

   Так как только что был очищен кэш, необходимого получить с DNS-сервера адрес запрашиваемого сайта. Там будет найдено совпадение доменного имени и сетевого адреса.



191	5.313796	192.168.0.143	8.8.8.8	DNS	73 Standard query 0x30d9 A krivonosov.ru
193	5.352002	8.8.8.8	192.168.0.143	DNS	89 Standard query response 0x30d9 A krivonosov.ru A 87.236.16.8



```
User Datagram Protocol, Src Port: 61072,
Domain Name System (query)
    Transaction ID: 0x30d9
  > Flags: 0x0100 Standard query
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
  v Queries
    v krivonosov.ru: type A, class IN
        Name: krivonosov.ru
        [Name Length: 13]
        [Label Count: 2]
        Type: A (Host Address) (1)
        Class: IN (0x0001)
```

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

   Прямой: преобразование домена в IP-адрес.
   Обратный: преобразование IP-адреса в домен.
   Рекурсивный: выполняется DNS-сервером, пока не будет найден домен или не будет получен ответ, что домен не существует. Рекурсия выполняется сервером.
   Итеративный: то же самое, что рекурсивный, но также допускается выполнение поиска клиентом.

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

   Если запрашиваемое изображение находится на другом сервере. Такое может

быть, если они, например, закэшированы на CDN-хостинге.

Этап 5. Анализ ARP-трафика

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

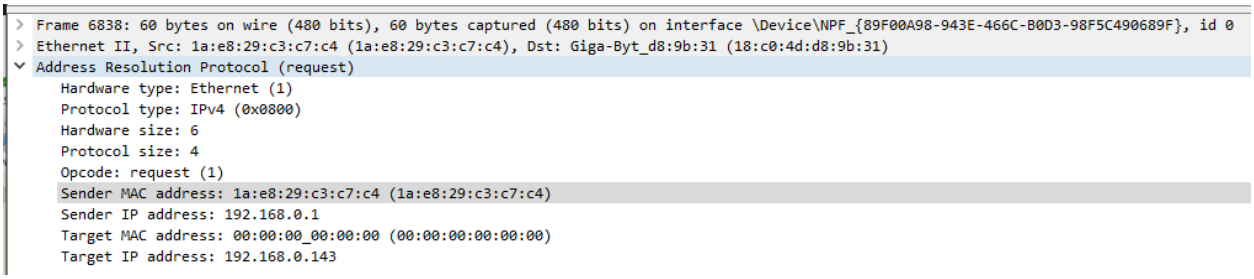
- очистить ARP-таблицу командой “netsh interface ip delete arpcache” (проверить очистилась ли таблица можно с помощью команды “arp -a”, выводящей таблицу на экран);
- очистить кэш браузера;
- зайти на Интернет-сайт, заданный по варианту.

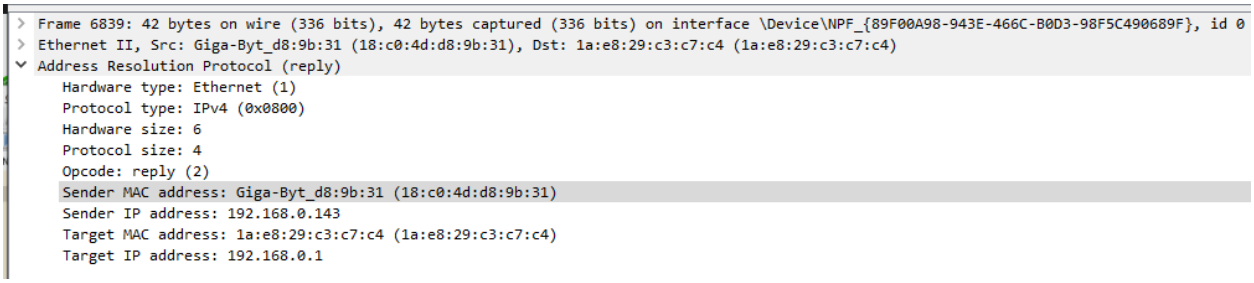
```
C:\Users\Egor>netsh interface ip delete arpcache
OK.

C:\Users\Egor>arp -a

Интерфейс: 192.168.0.143 --- 0x8
  адрес в Интернете      Физический адрес      Тип
  192.168.0.1           1a-e8-29-c3-c7-c4     динамический
  192.168.0.122         6c-e8-c6-60-c3-18     динамический
  192.168.0.255         ff-ff-ff-ff-ff-ff     статический
  224.0.0.2             01-00-5e-00-00-02     статический
  224.0.0.22            01-00-5e-00-00-16     статический
  224.0.0.251           01-00-5e-00-00-fb     статический

Интерфейс: 172.18.85.81 --- 0x13
  адрес в Интернете      Физический адрес      Тип
  172.18.85.95          ff-ff-ff-ff-ff-ff     статический
  224.0.0.22            01-00-5e-00-00-16     статический
  224.0.0.251           01-00-5e-00-00-fb     статический

C:\Users\Egor>
```

Вопросы

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

5965	14.705508	ASUSTekC_da:14:25	Broadcast	ARP	60 Who has 192.168.0.2? Tell 192.168.0.179
5974	15.056637	Giga-Byt_d8:9b:31	Broadcast	ARP	42 Who has 192.168.0.2? Tell 192.168.0.143
5999	15.705601	ASUSTekC_da:14:25	Broadcast	ARP	60 Who has 192.168.0.2? Tell 192.168.0.179
6327	23.287481	Giga-Byt_d8:9b:31	Broadcast	ARP	42 Who has 192.168.0.2? Tell 192.168.0.143
6366	24.050260	ASUSTekC_da:14:25	Broadcast	ARP	60 Who has 192.168.0.2? Tell 192.168.0.179
6368	24.056565	Giga-Byt_d8:9b:31	Broadcast	ARP	42 Who has 192.168.0.2? Tell 192.168.0.143
6410	24.705195	ASUSTekC_da:14:25	Broadcast	ARP	60 Who has 192.168.0.2? Tell 192.168.0.179
6422	25.056240	Giga-Byt_d8:9b:31	Broadcast	ARP	42 Who has 192.168.0.2? Tell 192.168.0.143
6448	25.705281	ASUSTekC_da:14:25	Broadcast	ARP	60 Who has 192.168.0.2? Tell 192.168.0.179
6838	31.465830	1a:e8:29:c3:c7:c4	Giga-Byt_d8:9b:31	ARP	60 Who has 192.168.0.143? Tell 192.168.0.1
6839	31.465840	Giga-Byt_d8:9b:31	1a:e8:29:c3:c7:c4	ARP	42 192.168.0.143 is at 18:c0:4d:d8:9b:31





1a:e8:29:c3:c7:c4 - адрес отправителя (маршрутизатор) (ip = 192.168.0.1)
18:c0:4d:d8:9b:31 - адрес устройства получателя (компьютер, с которого производится запрос на сайт) (ip = 192.168.0.143)
00:00:00:00:00:00 - broadcast-адрес

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

   В HTTP-пакетах присутствуют те же самые MAC-адреса, что и в ARP запросе. Они используются для идентификации отправителя и получателя HTTP-пакета.

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

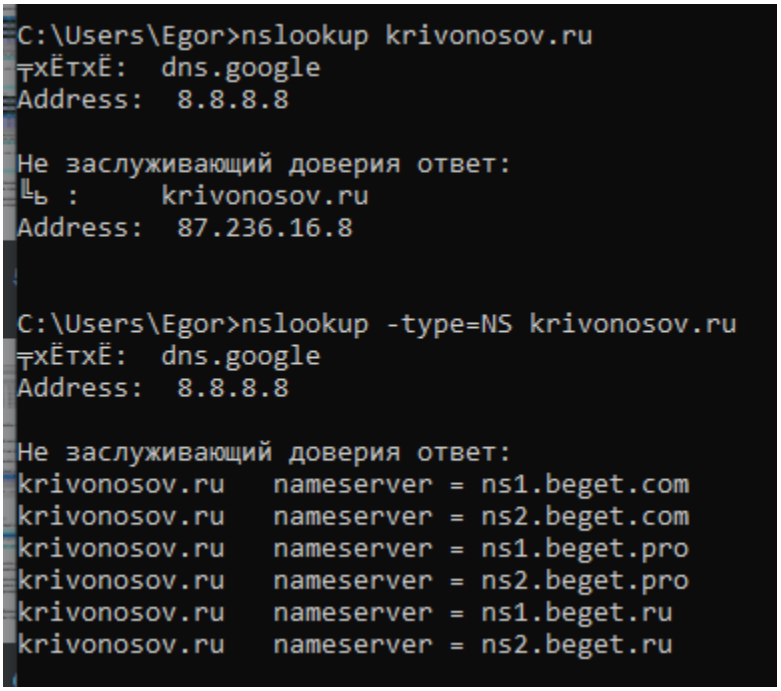
   Чтобы узел-получатель мог добавить информацию об узле-отправителе в свою ARP- таблицу.

Этап 6. Анализ трафика утилиты nslookup

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

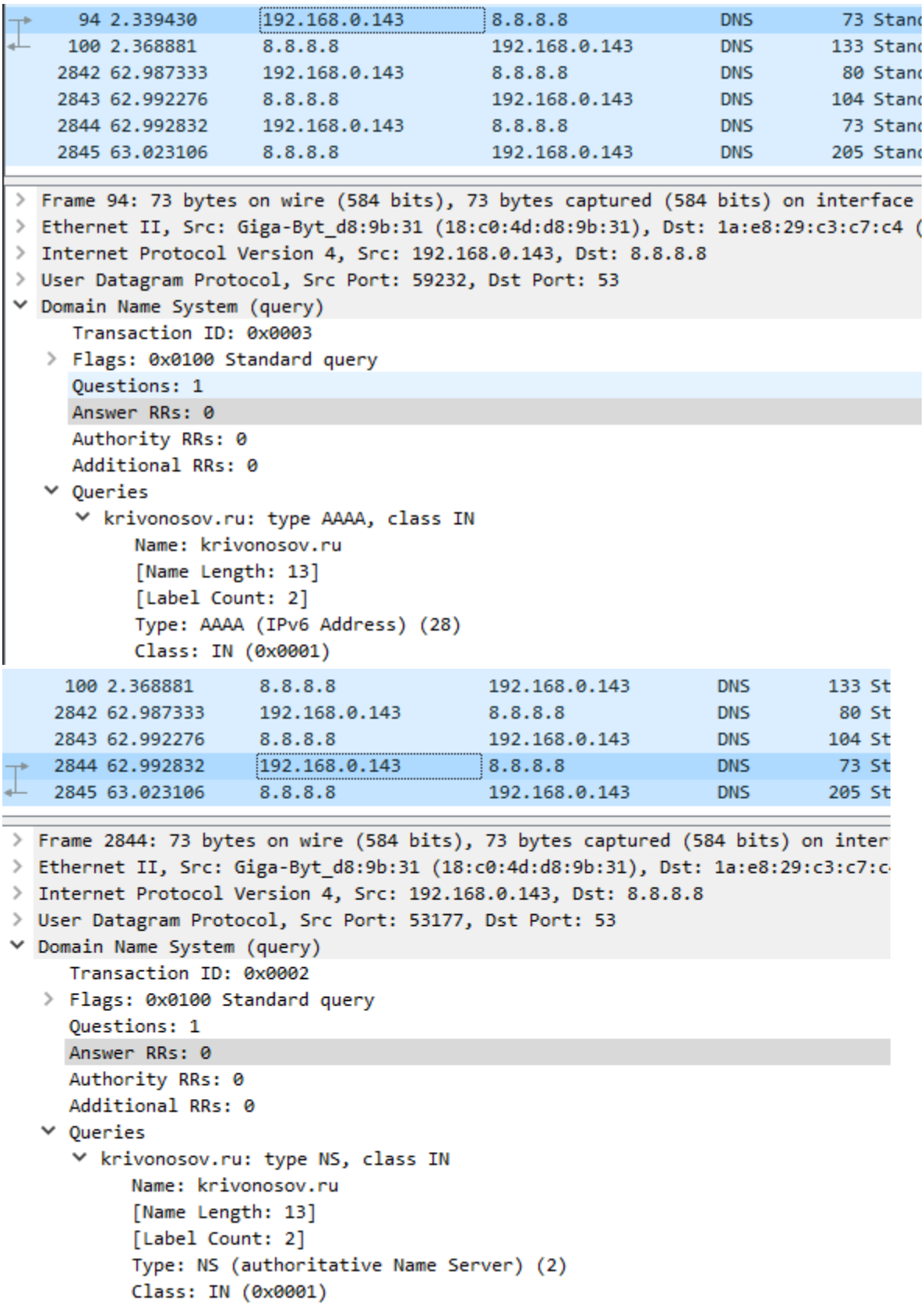
- Настроить Wireshark-фильтр: “ip.addr == ваш_IP_адрес”.
- Запустить в командной строке команду “nslookup адрес_сайта_по_варианту”.
- Дождаться отправки трёх DNS-запросов и трёх DNS-ответов (в работе нужно использовать только последние из них, т.к. первые два набора запросов/ответов специфичны для nslookup и не генерируются другими сетевыми приложениями).

- Повторить предыдущие два шага, используя команду: “nslookup -type=NS имя_сайта_по_варианту”.



Вопросы

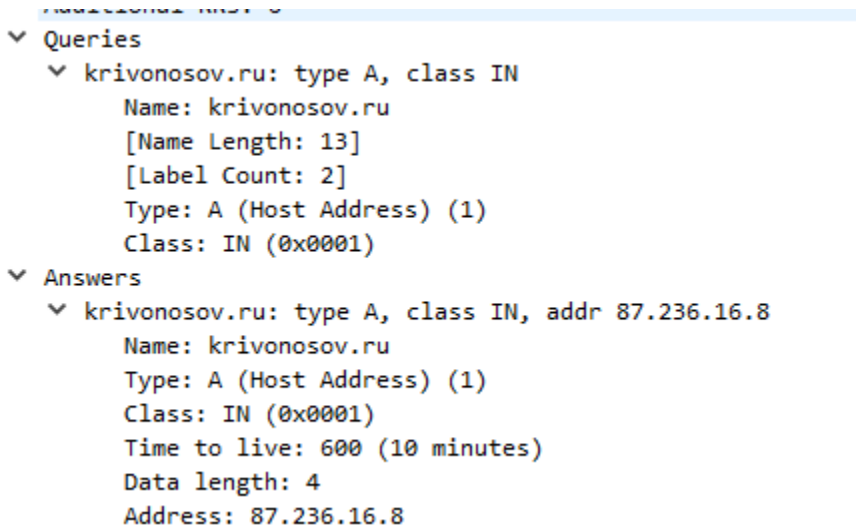
1. **Чем различается трасса трафика в п.2 и п.4, указанных выше?**


94 2.339430 192.168.0.143 8.8.8.8 DNS 73 Stand
100 2.368881 8.8.8.8 192.168.0.143 DNS 133 Stand
2842 62.987333 192.168.0.143 8.8.8.8 DNS 80 Stand
2843 62.992276 8.8.8.8 192.168.0.143 DNS 104 Stand
2844 62.992832 192.168.0.143 8.8.8.8 DNS 73 Stand
2845 63.023106 8.8.8.8 192.168.0.143 DNS 205 Stand
Frame 94: 73 bytes on wire (584 bits), 73 bytes captured (584 bits) on interface
Ethernet II, Src: Giga-Byt_d8:9b:31 (18:c0:4d:d8:9b:31), Dst: 1a:e8:29:c3:c7:c4 (
Internet Protocol Version 4, Src: 192.168.0.143, Dst: 8.8.8.8
User Datagram Protocol, Src Port: 59232, Dst Port: 53
Domain Name System (query)
Transaction ID: 0x0003
Flags: 0x0100 Standard query
Questions: 1
Answer RRs: 0
Authority RRs: 0
Additional RRs: 0
Queries
krivonosov.ru: type AAAA, class IN
Name: krivonosov.ru
[Name Length: 13]
[Label Count: 2]
Type: AAAA (IPv6 Address) (28)
Class: IN (0x0001)
100 2.368881 8.8.8.8 192.168.0.143 DNS 133 St
2842 62.987333 192.168.0.143 8.8.8.8 DNS 80 St
2843 62.992276 8.8.8.8 192.168.0.143 DNS 104 St
2844 62.992832 192.168.0.143 8.8.8.8 DNS 73 St
2845 63.023106 8.8.8.8 192.168.0.143 DNS 205 St
Frame 2844: 73 bytes on wire (584 bits), 73 bytes captured (584 bits) on inter
Ethernet II, Src: Giga-Byt_d8:9b:31 (18:c0:4d:d8:9b:31), Dst: 1a:e8:29:c3:c7:c
Internet Protocol Version 4, Src: 192.168.0.143, Dst: 8.8.8.8
User Datagram Protocol, Src Port: 53177, Dst Port: 53
Domain Name System (query)
Transaction ID: 0x0002
Flags: 0x0100 Standard query
Questions: 1
Answer RRs: 0
Authority RRs: 0
Additional RRs: 0
Queries
krivonosov.ru: type NS, class IN
Name: krivonosov.ru
[Name Length: 13]
[Label Count: 2]
Type: NS (authoritative Name Server) (2)
Class: IN (0x0001)

При запуске в п.2 утилита ищет IP-адрес хоста (запись типа A (IPv4) или AAAA (IPv6)).
При запуске в п.4 утилита ищет Name Server для запрашиваемого хоста.

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

   Данные запрашиваемого типа DNS-записи: для A - IPv4-адрес, для NS - список authoritative Name Server.



```
Queries
  krivonosov.ru: type NS, class IN
      Name: krivonosov.ru
      [Name Length: 13]
      [Label Count: 2]
      Type: NS (authoritative Name Server) (2)
      Class: IN (0x0001)
Answers
  krivonosov.ru: type NS, class IN, ns ns1.beget.com
      Name: krivonosov.ru
      Type: NS (authoritative Name Server) (2)
      Class: IN (0x0001)
      Time to live: 300 (5 minutes)
      Data length: 15
      Name Server: ns1.beget.com
  krivonosov.ru: type NS, class IN, ns ns2.beget.com
      Name: krivonosov.ru
      Type: NS (authoritative Name Server) (2)
      Class: IN (0x0001)
      Time to live: 300 (5 minutes)
      Data length: 6
      Name Server: ns2.beget.com
  krivonosov.ru: type NS, class IN, ns ns1.beget.pro
      Name: krivonosov.ru
      Type: NS (authoritative Name Server) (2)
      Class: IN (0x0001)
      Time to live: 300 (5 minutes)
      Data length: 15
      Name Server: ns1.beget.pro
```

NAME — имя хоста.
TYPE — тип ресурсной записи. Определяет формат и назначение данной ресурсной записи.
CLASS — класс ресурсной записи. Обычно IN для Internet (Код 0x0001)
TTL — (Time To Live) — допустимое время хранения данной ресурсной записи в кэше неответственного DNS-сервера.
RDLENGTH — длина поля данных.
RDATA — поле данных, формат и содержание которого зависит от типа записи.

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

```
Answers
  krivonosov.ru: type NS, class IN, ns ns1.beget.com
  krivonosov.ru: type NS, class IN, ns ns2.beget.com
  krivonosov.ru: type NS, class IN, ns ns1.beget.pro
  krivonosov.ru: type NS, class IN, ns ns2.beget.pro
  krivonosov.ru: type NS, class IN, ns ns1.beget.ru
  krivonosov.ru: type NS, class IN, ns ns2.beget.ru
```

Авторитативный отклик возвращают серверы, которые являются ответственными за зону, в которой описана информация, необходимая DNS-клиенту.

Этап 7. Анализ FTP-трафика

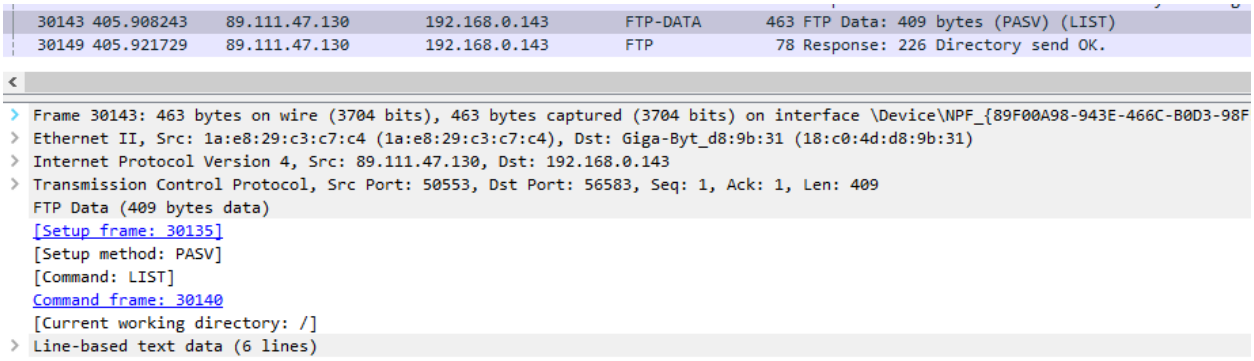
Необходимо отследить и проанализировать трафик протокола FTP, сгенерированный в результате выполнения следующих действий:

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

Вопросы

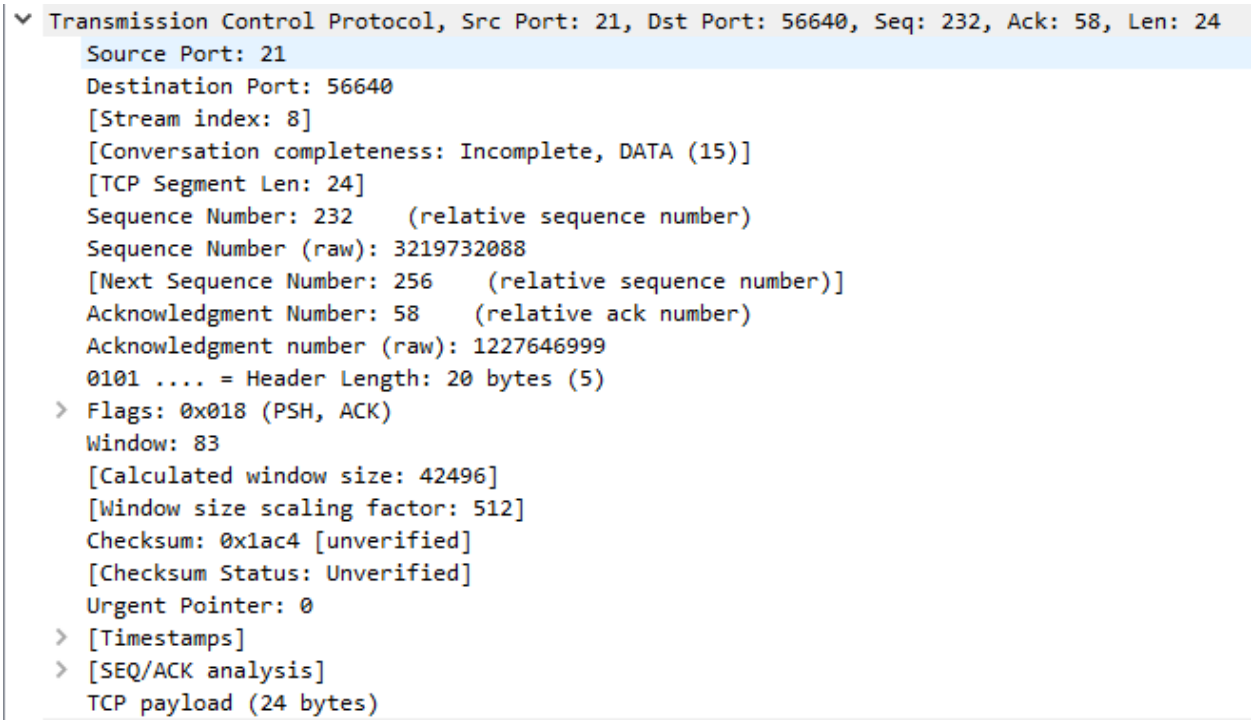
1. **Сколько байт данных содержится в пакете FTP-DATA?**

   Размер может быть любы, но не больше MTU.
   В данном случае 409 байт.



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

   Для потока управления на сервере используется порт 21. Для передачи данных используется порт 20, если передача идет в активном режиме, либо с любого порта клиента к любому порту сервера в пассивном режиме.



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

   FTP используется для выполнения команд (request/response), а FTP-DATA работает с файлами.

Этап 8. Анализ DHCP-трафика

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

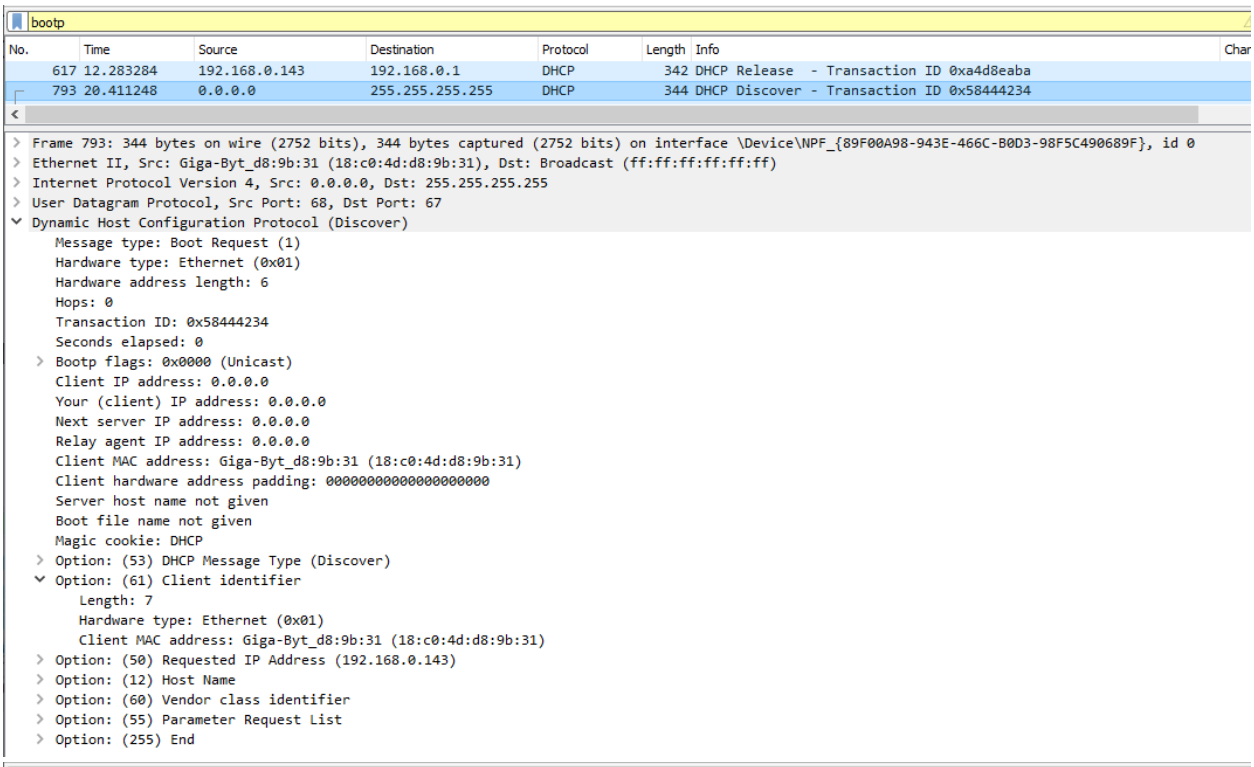
- Убедиться, что для назначения IP-адреса на компьютере был использован DHCP и что компьютеру был назначен IP-адрес.
- Настроить Wireshark-фильтр «bootp» (во время защиты УИР следует объяснить, почему именно такой фильтр используется для анализа DHCP-трафика).
- Сбросить текущий IP-адрес, выданный накануне перед этим DHCP-сервером, с помощью команды: “ipconfig /release“.
- Запросить новый IP-адрес с помощью команды: “ipconfig /renew“.
- Повторить п.3 и п.4.

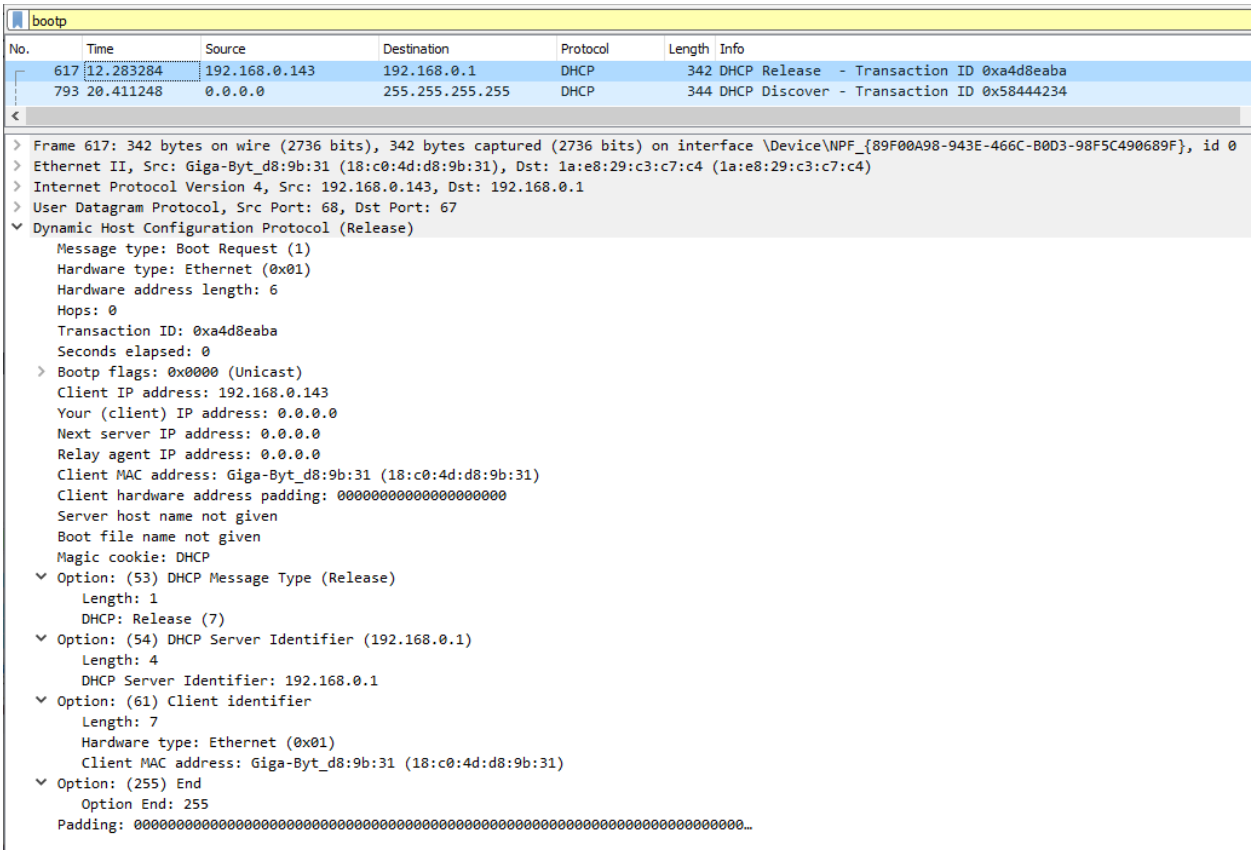
```
C:\Users\Egor>ipconfig /release

Настройка протокола IP для Windows


Адаптер Ethernet Ethernet 3:

   DNS-суффикс подключения . . . . . :
   Локальный IPv6-адрес канала . . . : fe80::d9f5:8153:a96f:61aa%8
   Основной шлюз. . . . . :

Адаптер Ethernet vEthernet (Default Switch):

   DNS-суффикс подключения . . . . . :
   Локальный IPv6-адрес канала . . . : fe80::d0ee:6f32:6212:7a14%19
   IPv4-адрес. . . . . : 172.18.85.81
   Маска подсети . . . . . : 255.255.255.240
   Основной шлюз. . . . . :

C:\Users\Egor>ipconfig /renew

Настройка протокола IP для Windows


Адаптер Ethernet Ethernet 3:

   DNS-суффикс подключения . . . . . :
   Локальный IPv6-адрес канала . . . : fe80::d9f5:8153:a96f:61aa%8
   IPv4-адрес. . . . . : 192.168.0.143
   Маска подсети . . . . . : 255.255.255.0
   Основной шлюз. . . . . : 192.168.0.1

Адаптер Ethernet vEthernet (Default Switch):

   DNS-суффикс подключения . . . . . :
   Локальный IPv6-адрес канала . . . : fe80::d0ee:6f32:6212:7a14%19
   IPv4-адрес. . . . . : 172.18.85.81
   Маска подсети . . . . . : 255.255.255.240
   Основной шлюз. . . . . :
```

Вопросы

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

   Оба запроса выполняются клиентом, DHCP Discover ищет DHCP-сервер в своей канальной среде, а DHCP Request принимает предлагаемый адрес и уведомляет DHCP-сервер об этом.


bootp
No. Time Source Destination Protocol Length Info
617 12.283284 192.168.0.143 192.168.0.1 DHCP 342 DHCP Release - Transaction ID 0xa4d8eaba
793 20.411248 0.0.0.0 255.255.255.255 DHCP 344 DHCP Discover - Transaction ID 0x58444234
Frame 793: 344 bytes on wire (2752 bits), 344 bytes captured (2752 bits) on interface \Device\NPF_{89F00A98-943E-466C-B0D3-98F5C490689F}, id 0
Ethernet II, Src: Giga-Byt_d8:9b:31 (18:c0:4d:d8:9b:31), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
Internet Protocol Version 4, Src: 0.0.0.0, Dst: 255.255.255.255
User Datagram Protocol, Src Port: 68, Dst Port: 67
Dynamic Host Configuration Protocol (Discover)
Message type: Boot Request (1)
Hardware type: Ethernet (0x01)
Hardware address length: 6
Hops: 0
Transaction ID: 0x58444234
Seconds elapsed: 0
Bootp flags: 0x0000 (Unicast)
Client IP address: 0.0.0.0
Your (client) IP address: 0.0.0.0
Next server IP address: 0.0.0.0
Relay agent IP address: 0.0.0.0
Client MAC address: Giga-Byt_d8:9b:31 (18:c0:4d:d8:9b:31)
Client hardware address padding: 00000000000000000000
Server host name not given
Boot file name not given
Magic cookie: DHCP
Option: (53) DHCP Message Type (Discover)
Option: (61) Client identifier
Length: 7
Hardware type: Ethernet (0x01)
Client MAC address: Giga-Byt_d8:9b:31 (18:c0:4d:d8:9b:31)
Option: (50) Requested IP Address (192.168.0.143)
Option: (12) Host Name
Option: (60) Vendor class identifier
Option: (55) Parameter Request List
Option: (255) End

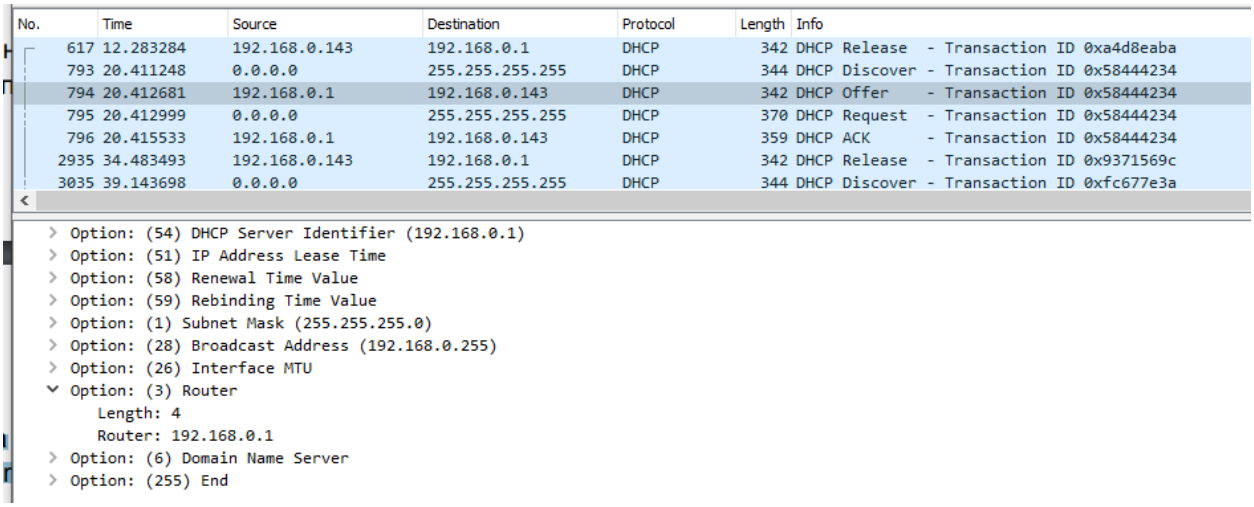

bootp
No. Time Source Destination Protocol Length Info
617 12.283284 192.168.0.143 192.168.0.1 DHCP 342 DHCP Release - Transaction ID 0xa4d8eaba
793 20.411248 0.0.0.0 255.255.255.255 DHCP 344 DHCP Discover - Transaction ID 0x58444234
Frame 617: 342 bytes on wire (2736 bits), 342 bytes captured (2736 bits) on interface \Device\NPF_{89F00A98-943E-466C-B0D3-98F5C490689F}, id 0
Ethernet II, Src: Giga-Byt_d8:9b:31 (18:c0:4d:d8:9b:31), Dst: 1a:e8:29:c3:c7:c4 (1a:e8:29:c3:c7:c4)
Internet Protocol Version 4, Src: 192.168.0.143, Dst: 192.168.0.1
User Datagram Protocol, Src Port: 68, Dst Port: 67
Dynamic Host Configuration Protocol (Release)
Message type: Boot Request (1)
Hardware type: Ethernet (0x01)
Hardware address length: 6
Hops: 0
Transaction ID: 0xa4d8eaba
Seconds elapsed: 0
Bootp flags: 0x0000 (Unicast)
Client IP address: 192.168.0.143
Your (client) IP address: 0.0.0.0
Next server IP address: 0.0.0.0
Relay agent IP address: 0.0.0.0
Client MAC address: Giga-Byt_d8:9b:31 (18:c0:4d:d8:9b:31)
Client hardware address padding: 00000000000000000000
Server host name not given
Boot file name not given
Magic cookie: DHCP
Option: (53) DHCP Message Type (Release)
Length: 1
DHCP: Release (7)
Option: (54) DHCP Server Identifier (192.168.0.1)
Length: 4
DHCP Server Identifier: 192.168.0.1
Option: (61) Client identifier
Length: 7
Hardware type: Ethernet (0x01)
Client MAC address: Giga-Byt_d8:9b:31 (18:c0:4d:d8:9b:31)
Option: (255) End
Option End: 255
Padding: 00000000000000000000000000000000000000000000000000000000000000000000000000000000…

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

   В качестве MAC-адреса источника клиент изначально подставил свой MAC-адрес, а MAC-адрес сервера он не знает, поэтому использует широковещательный MAC-адрес. Соответственно, в заголовке IP-пакета в качестве адреса источника клиент использовал 0.0.0.0. При отправке Offer или ACK пакетов, адреса источника соответствуют адресам DHCP-сервера, адреса назначения широковещательные.

3. **Каков IP-адрес DHCP-сервера?**

   192.168.0.1 - адрес роутера



4. **Что произойдёт, если очистить использованный фильтр “bootp”?**

   Будут отображаться все пакеты.

Этап 9. Анализ Telegram-трафика

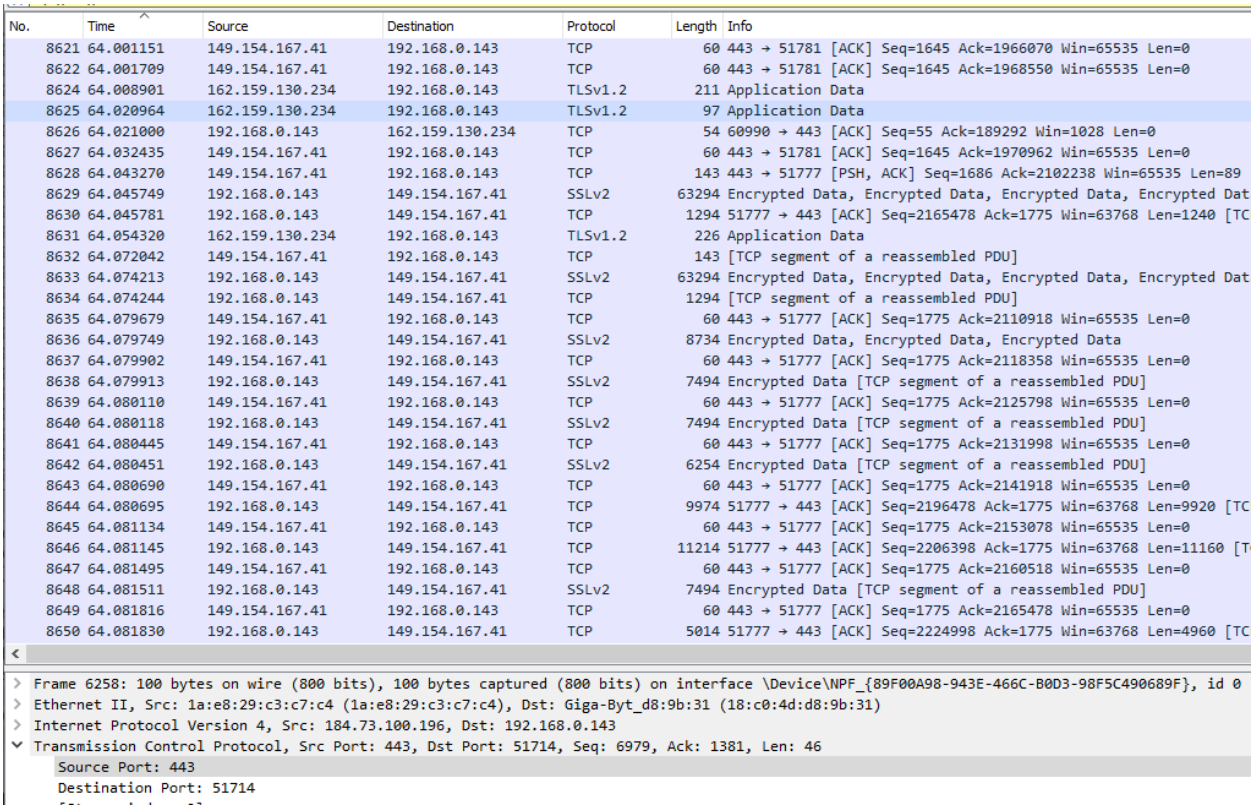
Необходимо отследить и проанализировать трафик Telegram, сгенерированный в результате выполнения следующих действий:

- отправить текстовое сообщение и получить ответ;
- осуществить короткий сеанс аудио-общения;
- осуществить короткий сеанс видео-общения.

Вопросы

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

   Текстовые данные передаются с помощью **TSL**
   При загрузке аудио и видео используются **TCP** и **SSL v2**
   Во время аудио-звонков используется **UDP**



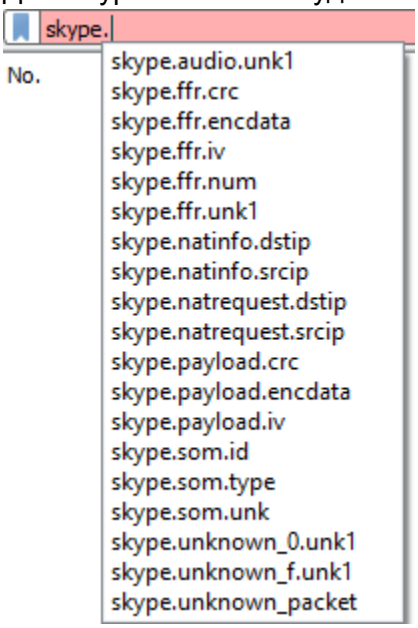
No.	Time	Source	Destination	Protocol	Length	Info
8621	64.001151	149.154.167.41	192.168.0.143	TCP	60	443 → 51781 [ACK] Seq=1645 Ack=1966070 Win=65535 Len=0
8622	64.001709	149.154.167.41	192.168.0.143	TCP	60	443 → 51781 [ACK] Seq=1645 Ack=1968550 Win=65535 Len=0
8624	64.008901	162.159.130.234	192.168.0.143	TLSv1.2	211	Application Data
8625	64.020964	162.159.130.234	192.168.0.143	TLSv1.2	97	Application Data
8626	64.021000	192.168.0.143	162.159.130.234	TCP	54	60990 → 443 [ACK] Seq=55 Ack=189292 Win=1028 Len=0
8627	64.032435	149.154.167.41	192.168.0.143	TCP	60	443 → 51781 [ACK] Seq=1645 Ack=1970962 Win=65535 Len=0
8628	64.043270	149.154.167.41	192.168.0.143	TCP	143	443 → 51777 [PSH, ACK] Seq=1686 Ack=2102238 Win=65535 Len=89
8629	64.045749	192.168.0.143	149.154.167.41	SSLv2	63294	Encrypted Data, Encrypted Data, Encrypted Data, Encrypted Dat
8630	64.045781	192.168.0.143	149.154.167.41	TCP	1294	51777 → 443 [ACK] Seq=2165478 Ack=1775 Win=63768 Len=1240 [TC
8631	64.054320	162.159.130.234	192.168.0.143	TLSv1.2	226	Application Data
8632	64.072042	149.154.167.41	192.168.0.143	TCP	143	[TCP segment of a reassembled PDU]
8633	64.074213	192.168.0.143	149.154.167.41	SSLv2	63294	Encrypted Data, Encrypted Data, Encrypted Data, Encrypted Dat
8634	64.074244	192.168.0.143	149.154.167.41	TCP	1294	[TCP segment of a reassembled PDU]
8635	64.079679	149.154.167.41	192.168.0.143	TCP	60	443 → 51777 [ACK] Seq=1775 Ack=2110918 Win=65535 Len=0
8636	64.079749	192.168.0.143	149.154.167.41	SSLv2	8734	Encrypted Data, Encrypted Data, Encrypted Data
8637	64.079902	149.154.167.41	192.168.0.143	TCP	60	443 → 51777 [ACK] Seq=1775 Ack=2118358 Win=65535 Len=0
8638	64.079913	192.168.0.143	149.154.167.41	SSLv2	7494	Encrypted Data [TCP segment of a reassembled PDU]
8639	64.080110	149.154.167.41	192.168.0.143	TCP	60	443 → 51777 [ACK] Seq=1775 Ack=2125798 Win=65535 Len=0
8640	64.080118	192.168.0.143	149.154.167.41	SSLv2	7494	Encrypted Data [TCP segment of a reassembled PDU]
8641	64.080445	149.154.167.41	192.168.0.143	TCP	60	443 → 51777 [ACK] Seq=1775 Ack=2131998 Win=65535 Len=0
8642	64.080451	192.168.0.143	149.154.167.41	SSLv2	6254	Encrypted Data [TCP segment of a reassembled PDU]
8643	64.080690	149.154.167.41	192.168.0.143	TCP	60	443 → 51777 [ACK] Seq=1775 Ack=2141918 Win=65535 Len=0
8644	64.080695	192.168.0.143	149.154.167.41	TCP	9974	51777 → 443 [ACK] Seq=2196478 Ack=1775 Win=63768 Len=9920 [TC
8645	64.081134	149.154.167.41	192.168.0.143	TCP	60	443 → 51777 [ACK] Seq=1775 Ack=2153078 Win=65535 Len=0
8646	64.081145	192.168.0.143	149.154.167.41	TCP	11214	51777 → 443 [ACK] Seq=2206398 Ack=1775 Win=63768 Len=11160 [T
8647	64.081495	149.154.167.41	192.168.0.143	TCP	60	443 → 51777 [ACK] Seq=1775 Ack=2160518 Win=65535 Len=0
8648	64.081511	192.168.0.143	149.154.167.41	SSLv2	7494	Encrypted Data [TCP segment of a reassembled PDU]
8649	64.081816	149.154.167.41	192.168.0.143	TCP	60	443 → 51777 [ACK] Seq=1775 Ack=2165478 Win=65535 Len=0
8650	64.081830	192.168.0.143	149.154.167.41	TCP	5014	51777 → 443 [ACK] Seq=2224998 Ack=1775 Win=63768 Len=4960 [TC

```
> Frame 6258: 100 bytes on wire (800 bits), 100 bytes captured (800 bits) on interface \Device\NPF_{89F00A98-943E-466C-B0D3-98F5C490689F}, id 0
> Ethernet II, Src: 1a:e8:29:c3:c7:c4 (1a:e8:29:c3:c7:c4), Dst: Giga-Byt_d8:9b:31 (18:c0:4d:d8:9b:31)
> Internet Protocol Version 4, Src: 184.73.100.196, Dst: 192.168.0.143
v Transmission Control Protocol, Src Port: 443, Dst Port: 51714, Seq: 6979, Ack: 1381, Len: 46
    Source Port: 443
    Destination Port: 51714
```

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

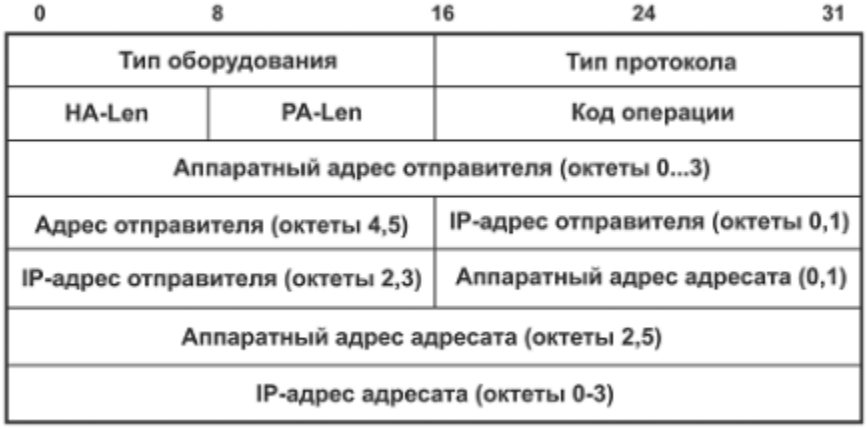
   В Wireshark можно установить фильтр по протоколу, соответственно нужно установить tsl для текстовых данных, tcp для просмотра загрузки аудио/видео, udp для аудио.

Для Skype есть более удобные фильтры в wireshark

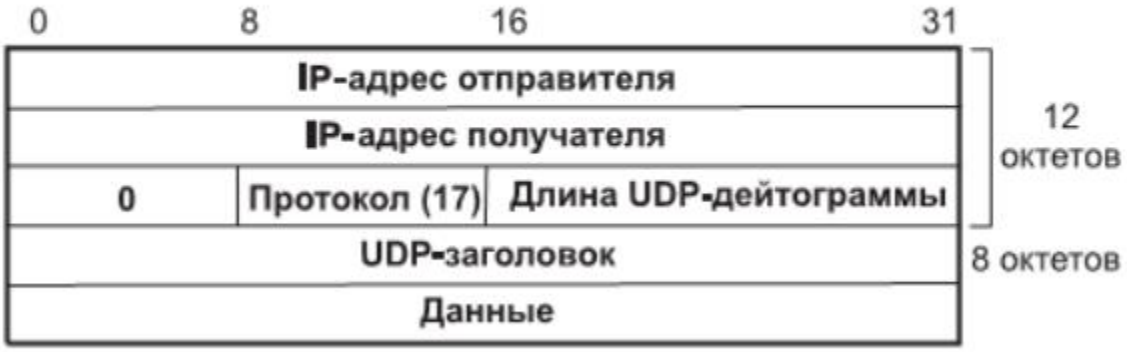


Структура наблюдаемых пакетов

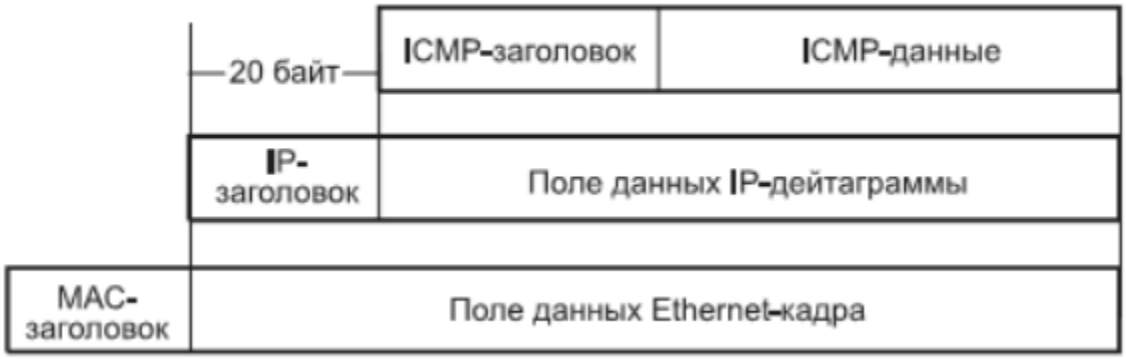
ARP



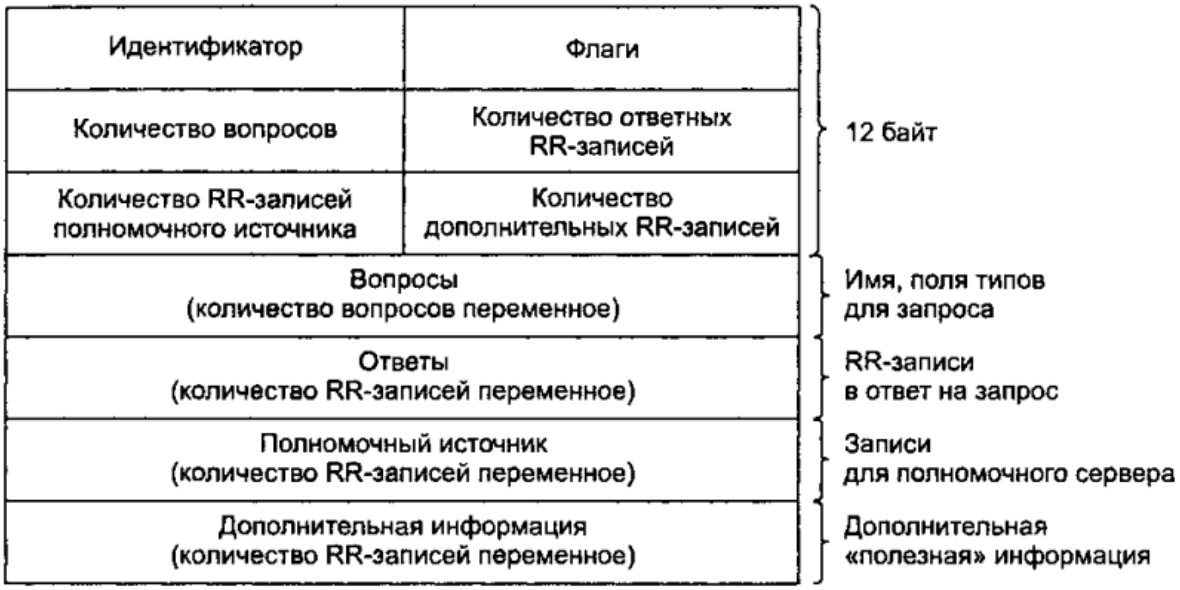
UDP


0
8
16
31
IP-адрес отправителя
IP-адрес получателя
0
Протокол (17)
Длина UDP-дейтограммы
UDP-заголовок
Данные
12 октетов
8 октетов

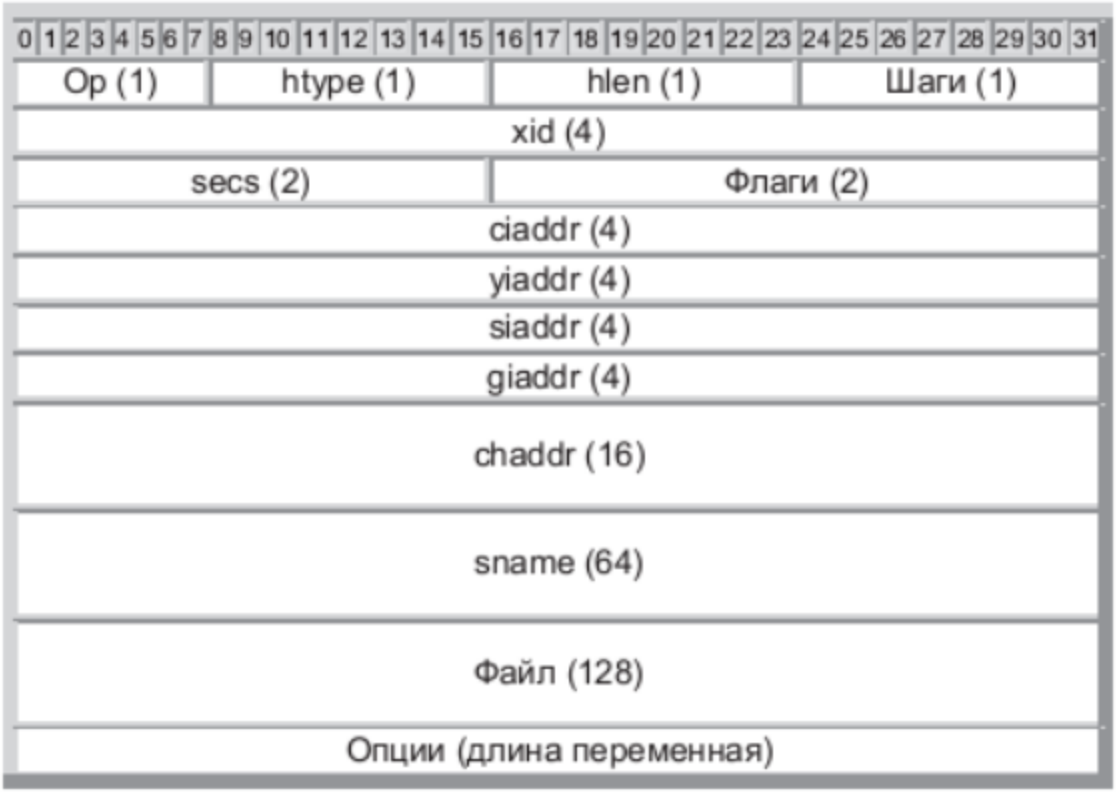
ICMP


20 байт
ICMP-заголовок
ICMP-данные
IP-заголовок
Поле данных IP-дейтаграммы
MAC-заголовок
Поле данных Ethernet-кадра

DNS


Идентификатор
Флаги
Количество вопросов
Количество ответных RR-записей
Количество RR-записей полномочного источника
Количество дополнительных RR-записей
12 байт
Вопросы (количество вопросов переменное)
Имя, поля типов для запроса
Ответы (количество RR-записей переменное)
RR-записи в ответ на запрос
Полномочный источник (количество RR-записей переменное)
Записи для полномочного сервера
Дополнительная информация (количество RR-записей переменное)
Дополнительная «полезная» информация

DHCP



Вывод

В ходе лабораторной работы был проанализирован сетевой трафик с помощью программы Wireshark. Мы изучили, какие пакеты передаются при работе утилит ping, и tracert и какую информацию они содержат. Также был проведен анализ трафика HTTP-запросов и влияние на него кэширования данных. Кэширование также влияет на работу DNS, во время выполнения работы нам необходимо было очистить кэши и посмотреть на работу DNS-запросов. Далее был рассмотрен трафик при выполнении arp-запросов, для этого нужно было очистить arp-таблицу. Кроме того был проанализирован трафик при работе с FTP, самой сложной частью этого задания оказалось найти действующий ftp-сервер. Для анализа DHCP пришлось вспомнить 2 лабораторную и работу этого протокола, чтобы описать типы запросов и данные, содержащиеся в соответствующих пакетах. И последним был рассмотрен трафик Telegram, мы узнали, какие протоколы используются в нем для передачи разных типов данных.