Санкт-Петербургский национально исследовательский университет

информационных технологий, механики и оптики

Факультет программной инженерии и компьютерной техники



**Лабораторная работа № 4 по дисциплине**

**«Компьютерные сети»**

Выполнил:

Студент группы: P33121

Нуруллаев Даниил

Преподаватель: Тропченко А. А.

Санкт-Петербург

2023

**Задание**

Запустить Wireshark (иногда для этого требуются права Администратора). 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).

Инициировать процесс передачи трафика по сети (например, в браузере открыть сайт, заданный по варианту, или запустить соответствующую сетевую утилиту – см. ниже);

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”.

Дождаться появления данных в списке захваченных пакетов и убедиться, что количество пакетов достаточно для выполнения задания.

Сохранить захваченный трафик в файл-трассу (pcap). Указанный файл нужно предъявить по первому требованию преподавателя во время защиты, если в этом возникнет необходимость.

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).

Написать в отчёте ответы на вопросы задания (для этого может потребоваться самостоятельно изучить назначение соответствующей заданию сетевой утилиты, использованной для создания трафика).

Поместить в отчёт скриншоты окна Wireshark, иллюстрирующие ответы из вышеуказанных п.6 и п.7.

**URL: godbolt.org**

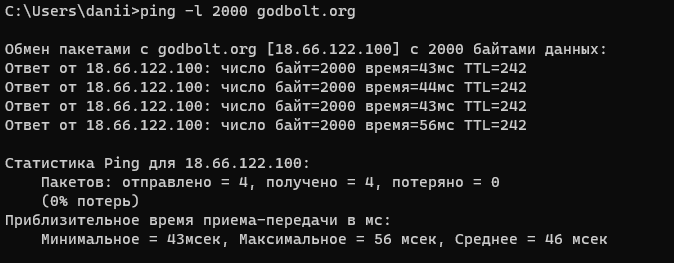
**Выполнение:**

**Этап 1. Анализ трафика утилиты ping**

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

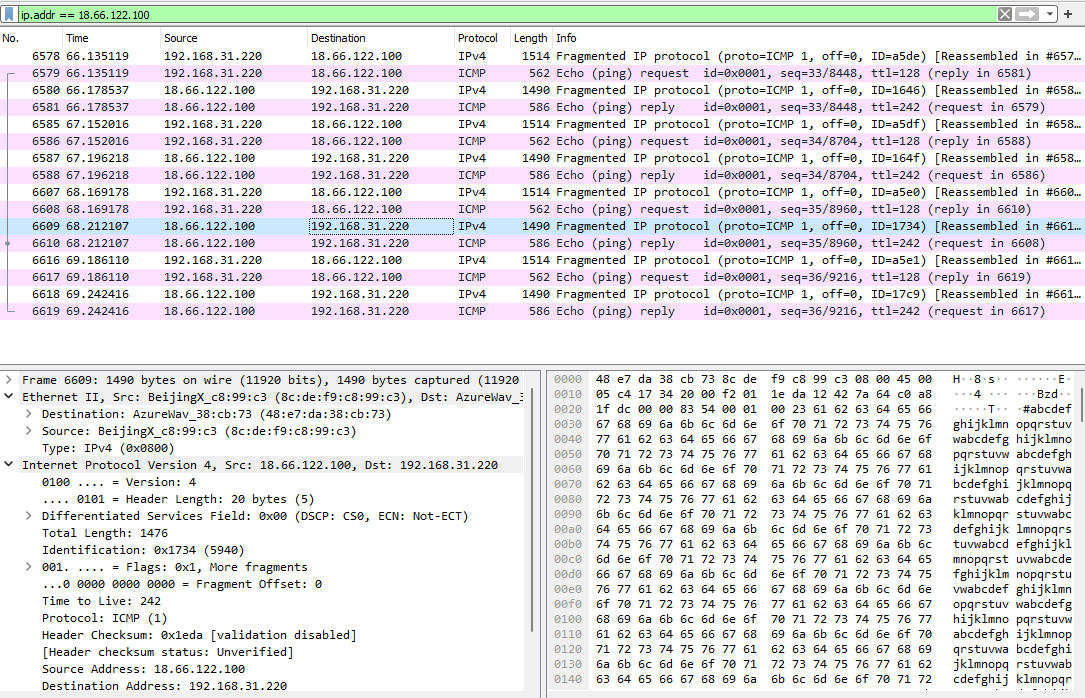
“ping -l 2000 wireshark.org”.

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

****

**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 IPпротокола появляются фрагменты “Fragmented IP protocol”. Флаг More Fragments как раз указывает на наличие фрагментации исходного пакета.

****

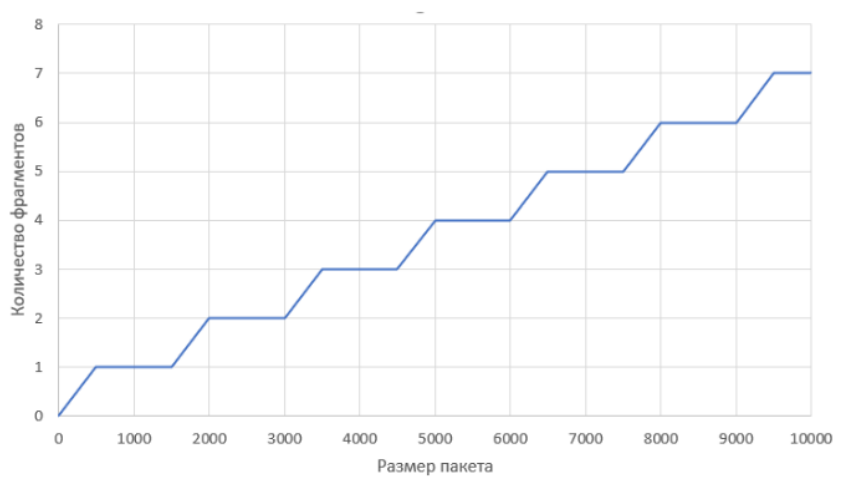
**2) Какая информация указывает, является ли фрагмент пакета последним или промежуточным?**

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?**

Ping передает данные по 32 байта, так что фрагментации для них нет, т.е. 0 фрагментов

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

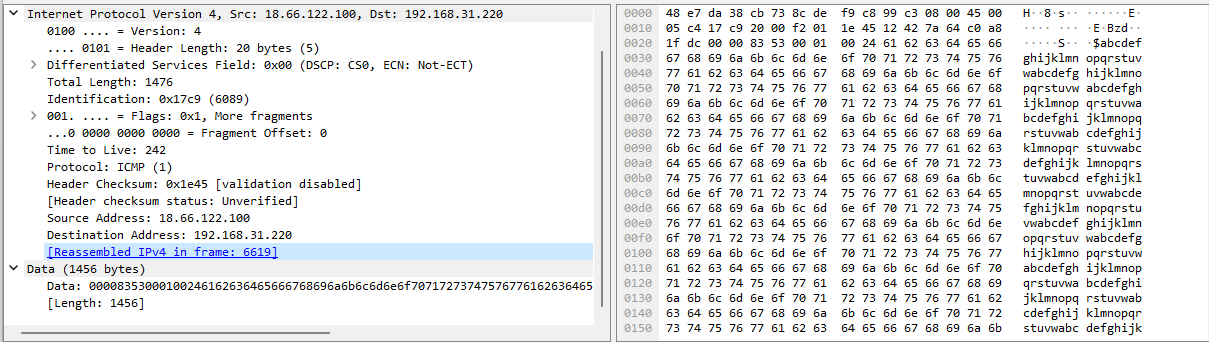
****

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

Для изменения TTL нужно добавить ключ -i или -t(для linux), его аргументом является срок жизни пакета в миллисекундах.

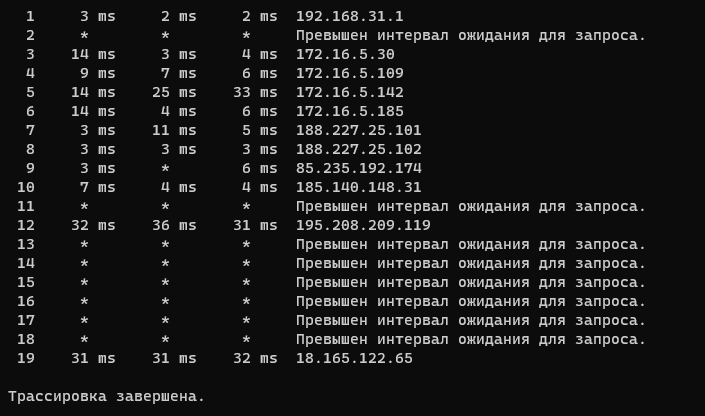
**6) Что содержится в поле данных ping-пакета?**

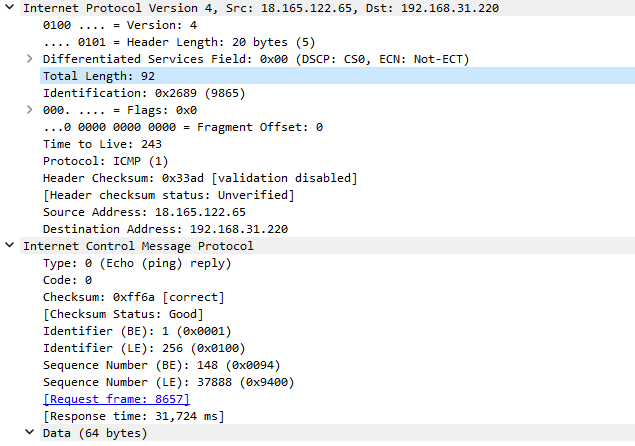
В поле данных содержатся символы английского алфавита.

****

# **Этап 2. Анализ трафика утилиты tracert**

Необходимо отследить и проанализировать трафик, создаваемый утилитой tracert (или traceroute в Linux)

****

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

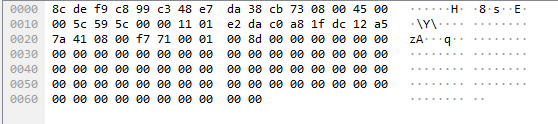
Заголовок: 20 байт, Данные: 64 байта

**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 ICMPпакетов, генерируемых утилитой ping (см. предыдущее задание).**

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

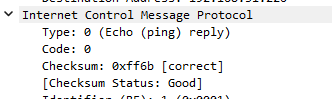
****

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

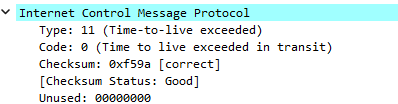
Пакеты «ICMP reply» указывают на получение ответного сообщения.

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.

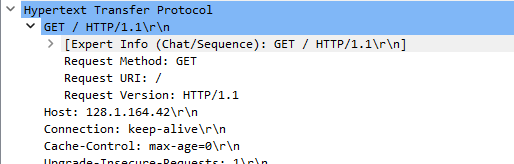
Для этого в поле с детальной информацией о пакете нужно развернуть строку “HTTP”. Затем необходимо обновить страницу в браузере так, чтобы вместо «HTTP GET» был сгенерирован «HTTP CONDITIONAL GET» (так называемый «условный GET»). Условные запросы GET содержат поля If-Modified-Since, If-Match, If-Range 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».

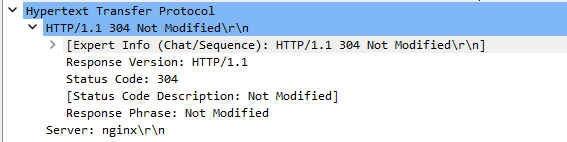
**1) По результатам анализа собранной трассы покажите, каким образом протокол HTTP передавал содержимое страницы при первичном посещении страницы и при вторичном запросе-обновлении от браузера (т.е. при различных видах GET-запросов).**

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.



****

****

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.

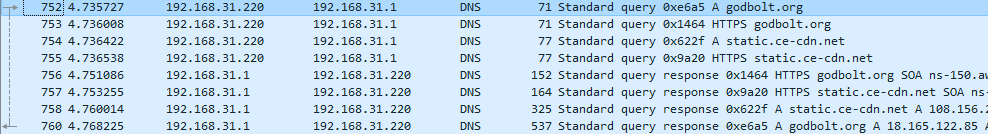
**Этап 4. Анализ DNS-трафика**

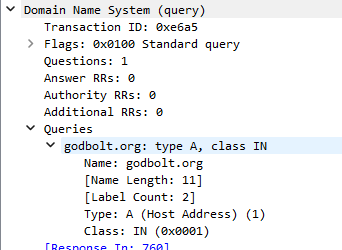
Необходимо отследить и проанализировать трафик протокола DNS, сгенерированный в результате выполнения следующих действий:

* + - * 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.

**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.

****

****

**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.

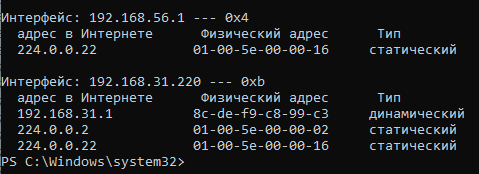
**3) В какой ситуации нужно выполнять независимые DNS-запросы для получения содержащихся на сайте изображений?**

Если запрашиваемое изображение находится на другом сервере. Такое может быть, если они, например, закэшированы на CDN-хостинге.

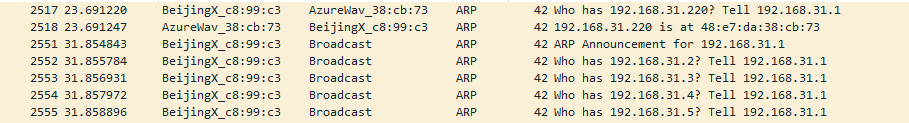
**Этап 5. Анализ ARP-трафика**

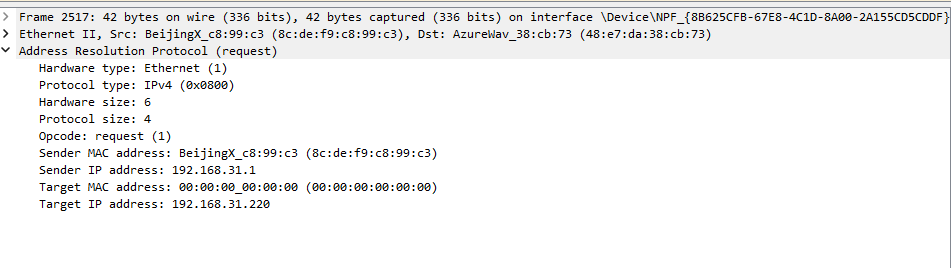
Необходимо отследить и проанализировать трафик протокола ARP, сгенерированный в результате выполнения следующих действий:

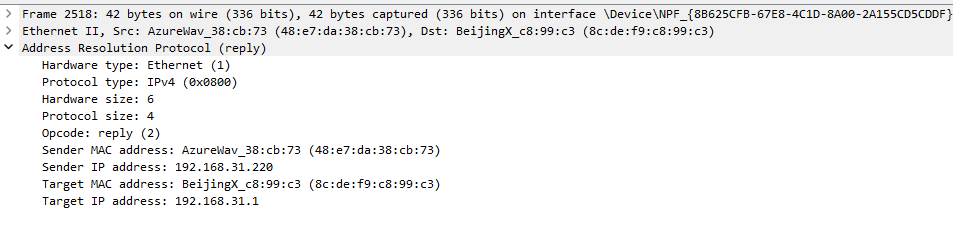
* + - * очистить ARP-таблицу командой “netsh interface ip delete arpcache” (проверить очистилась ли таблица можно с помощью команды команды “arp - a”, выводящей таблицу на экран);
      * очистить кэш браузера;
      * зайти на Интернет-сайт, заданный по варианту.

****

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

****

****

****

8c:de:f9:c8:99:c3 – адрес отправителя

48:e7:da:38:cb:73 – адрес получателя

00:00:00:00:00:00 – broadcast-адрес

**2) Какие МАС-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 МАС-адреса, что и в ARP запросе. Они используются для идентификации отправителя и получателя HTTP-пакета.

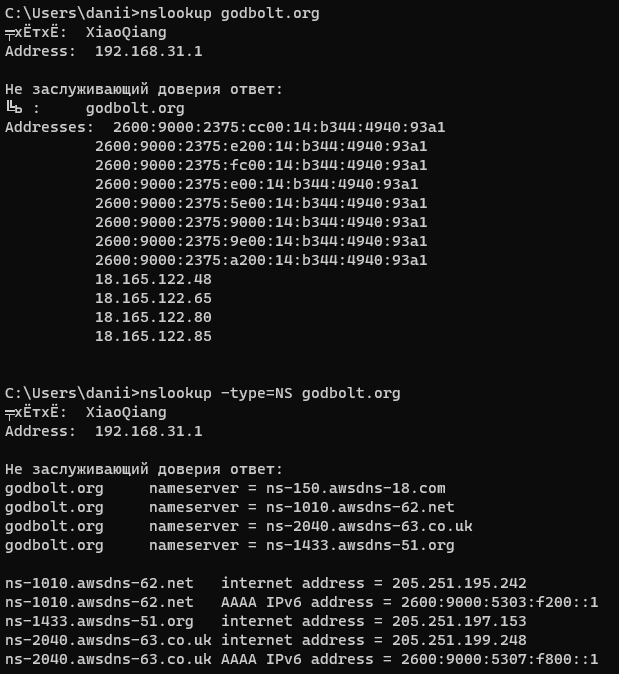
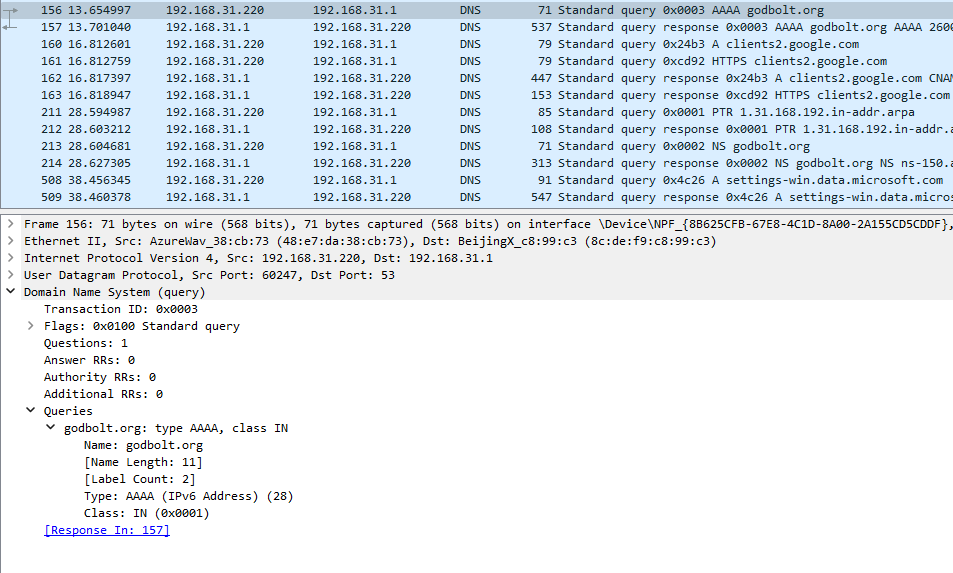
**3) Для чего ARP-запрос содержит IP-адрес источника?**

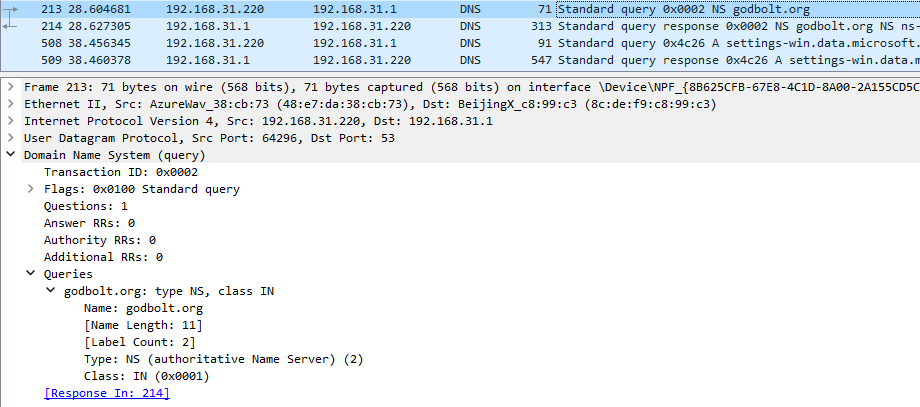
Чтобы узел-получатель мог добавить информацию об узле-отправителе в свою ARP- таблицу.

**Этап 6. Анализ трафика утилиты nslookup**

Это задание является необязательным, его необходимо выполнить только для желающих получить оценку «**хорошо**» или «**отлично**». Необходимо отследить и проанализировать трафик протокола DNS, сгенерированный в результате выполнения следующих действий:

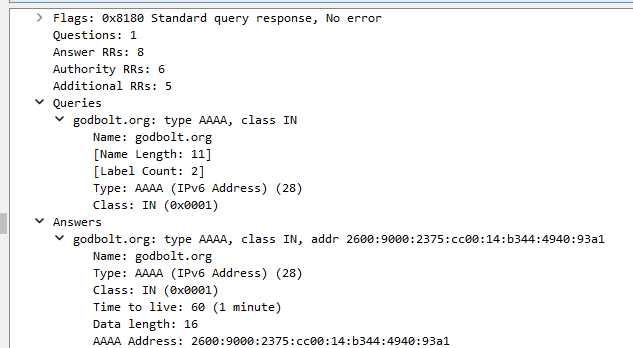
1. Настроить Wireshark-фильтр: “ip.addr == ваш\_IP\_адрес”.
2. Запустить в командной строке команду “nslookup адрес\_сайта\_по\_варианту”.
3. Дождаться отправки трёх DNS-запросов и трёх DNS-ответов (в работе нужно использовать только последние из них, т.к. первые два набора запросов/ответов специфичны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

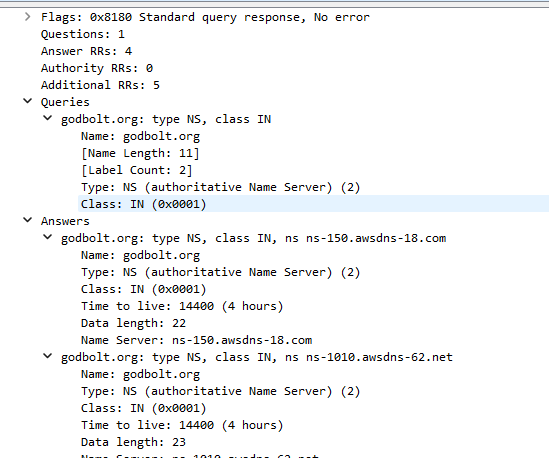
**  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
1) Чем различается трасса трафика в п.2 и п.4, указанных выше?  
**

****

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.  
  
  




NAME — имя хоста.

TYPE — тип ресурсной записи.

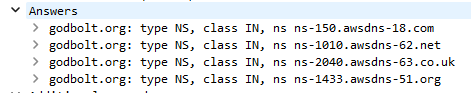
Определяет формат и назначение данной ресурсной записи.

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?  
**

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

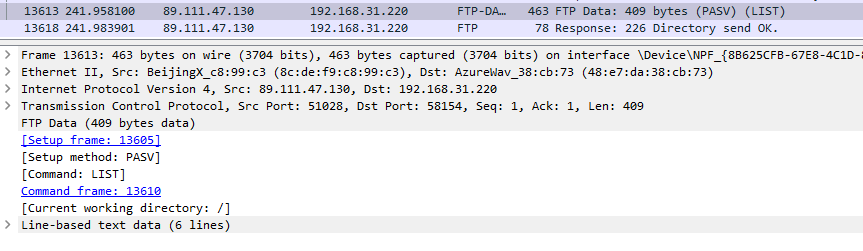
**Этап 7. Анализ FTP-трафика**

Это задание является необязательным, его необходимо выполнить только для желающих получить оценку «**хорошо**» или «**отлично**». 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.

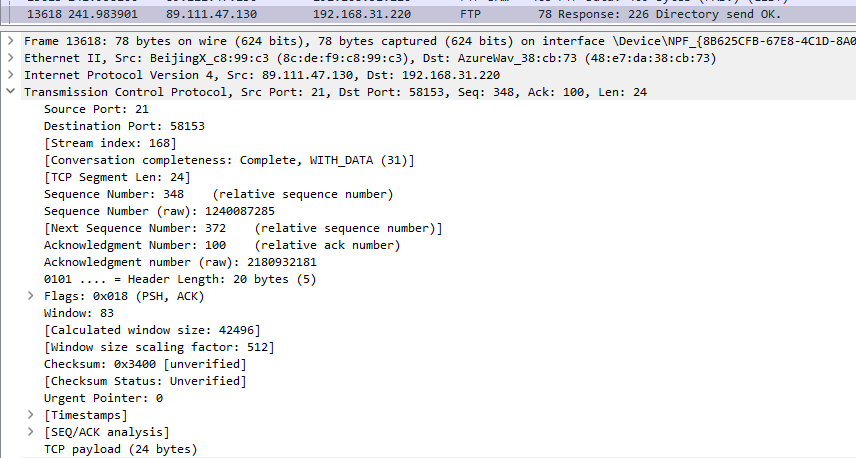
**1) Сколько байт данных содержится в пакете FTP-DATA?**

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**

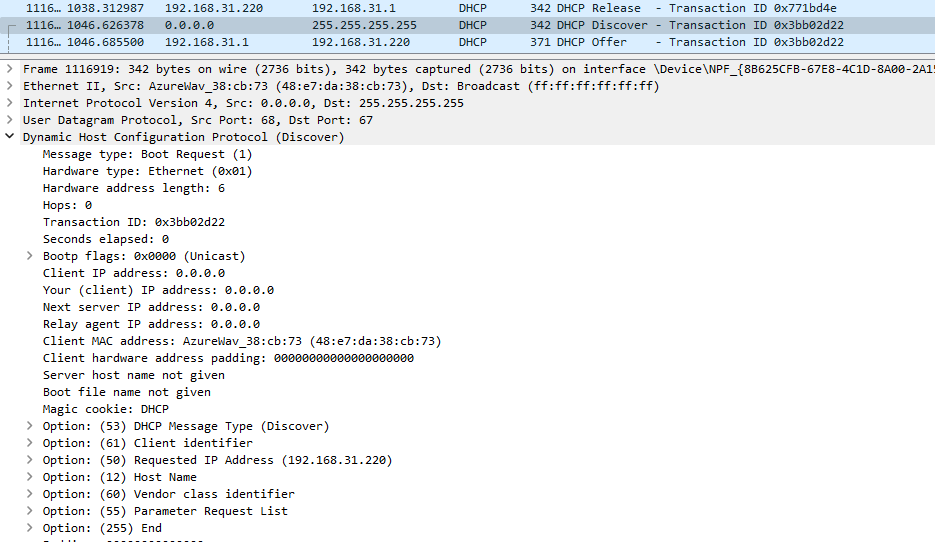
Это задание является необязательным, его необходимо выполнить для желающих получить оценку «отлично». Необходимо отследить и проанализировать трафик протокола DHCP, сгенерированный в результате выполнения следующих действий:

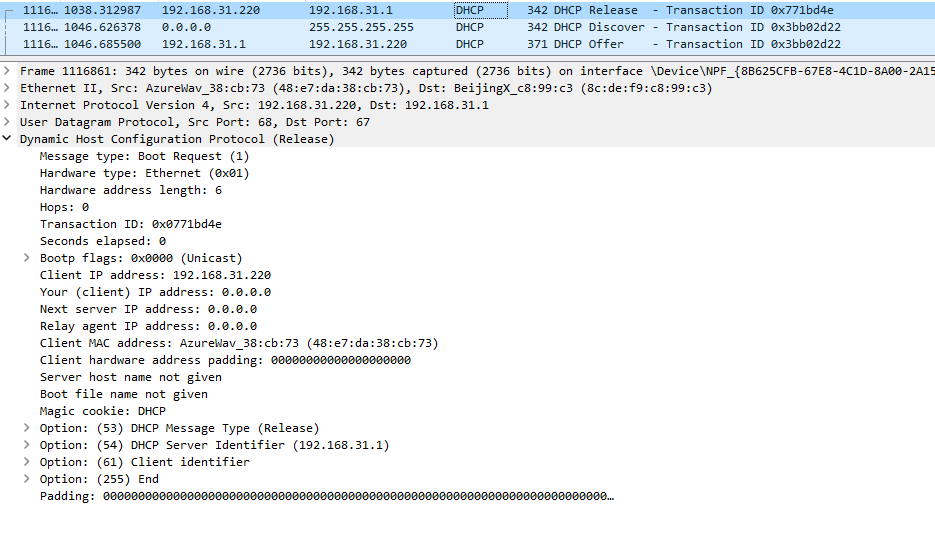
1. Убедиться, что для назначения IP-адреса на компьютере был использован DHCP и что компьютеру был назначен IP-адрес.
2. Настроить Wireshark-фильтр «bootp» (во время защиты УИР следует объяснить, почему именно такой фильтр используется для анализа DHCP-трафика).
3. Сбросить текущий IP-адрес, выданный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

“ipconfig /release“.

1. Запросить новый IP-адрес с помощью команды: “ipconfig /renew“.
2. Повторить п.3 и п.4.
3. **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.





**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 MACадрес. 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?**

192.168.31.1 – адрес роутера

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

Будут отображаться все пакеты.

**Этап 9. Анализ Skype-трафика**

Это задание является необязательным, его необходимо выполнить для желающих получить оценку «отлично». Необходимо отследить и проанализировать трафик Skype (или любой другой аналогичной по функциональности программы), сгенерированный в результате выполнения следующих действий:

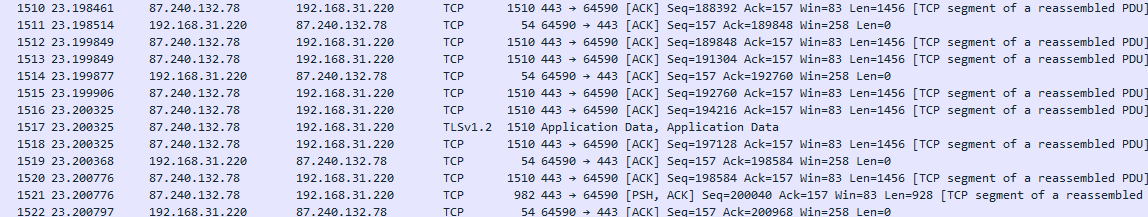
* + отправить текстовое сообщение и получить ответ;
  + осуществить короткий сеанс аудио-общения;
  + осуществить короткий сеанс видео-общения.

**1) Чем различаются пакета разных видов Skype-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



****

**2) Какой Wireshark-фильтр следует использовать для независимой идентификации Skype-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.  
  
**Вывод:**

Во время выполнения лабораторной работы мы познакомились с работой различных протоколов передачи данных, проанализировали переданные пакеты с помощью программы Wireshark и протестировали соединения через разные утилиты.