

Факультет программной инженерии и компьютерной техники

Компьютерные сети

Лабораторная работа №3

Преподаватель: Маркина Татьяна Анатольевна

Выполнил: Кульбако Артемий Юрьевич Р33112

# Цель работы

Цель работы – изучить структуру протокольных блоков данных, анализируя реальный трафик на компьютере студента с помощью бесплатно распространяемой утилиты Wireshark. В процессе выполнения домашнего задания выполняются наблюдения за передаваемым трафиком с компьютера пользователя в Интернет и в обратном направлении. Применение специализированной утилиты Wireshark позволяет наблюдать структуру передаваемых кадров, пакетов и сегментов данных различных сетевых протоколов. При выполнении УИР требуется анализировать последовательности команд и назначение служебных данных, используемых для организации обмена данными в следующих протоколах: ARP, DNS, FTP, HTTP, DHCP.

# Выполнение

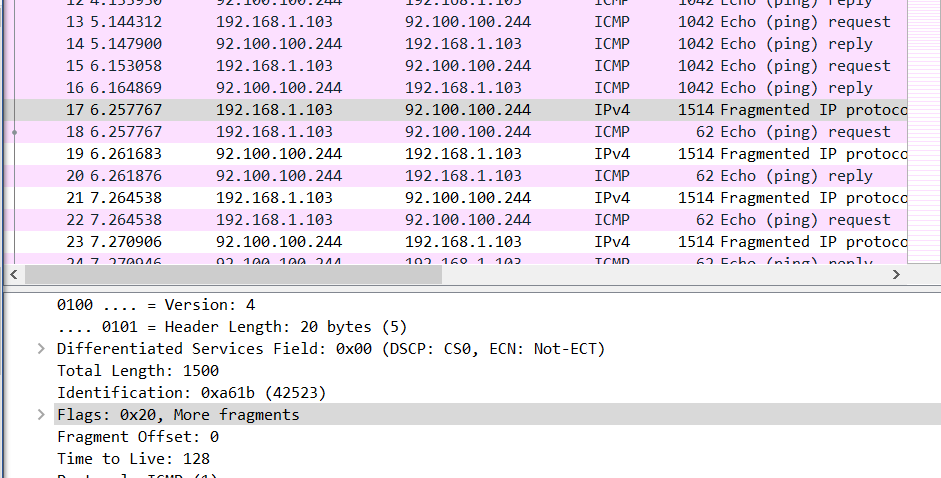
Для выполнения лабораторной работы мной был получен уникальный бесплатный адрес в сервисе noip.com: **kulbakoay.ddns.net**, однозначно идентифицирующий меня по фамилии и инициалам.

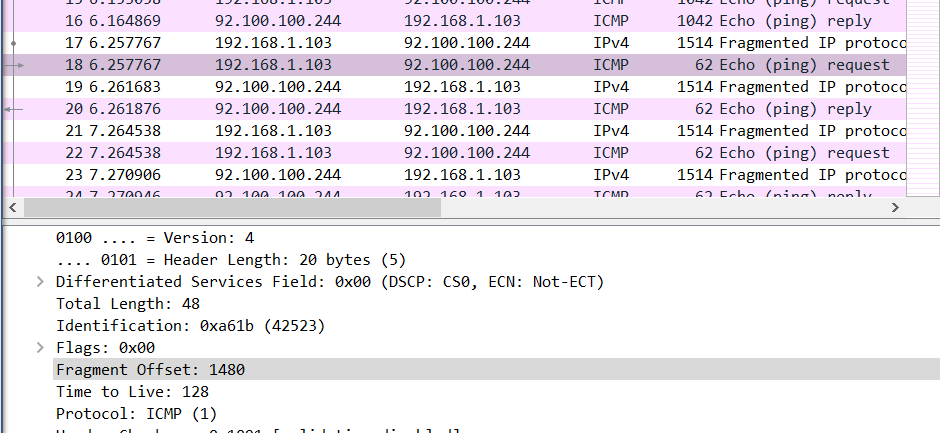
## Анализ трафика утилиты ping

for /l %%i in (500, 500, 10000) do ping -l %%i kulbakoay.ddns.net

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

Да, имеет. С увеличением размера пакета, мы можем видеть, что на уровне IP-протокола появляются фрагменты с флагом More Fragments, одинаковым идентификатором и смещением != 0

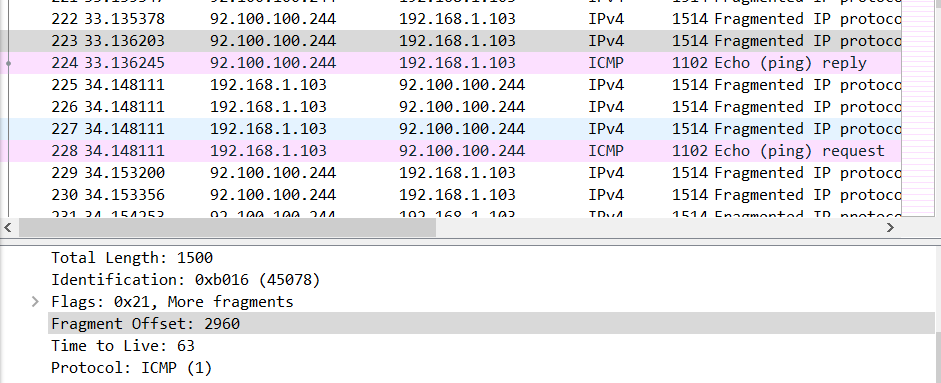




1. Какая информация указывает, является ли фрагмент пакета последним или промежуточным?

Последний фрагмент можно идентифицировать по ненулевому смещению (это значит, что в пакеты были ещё фрагменты до него), но флаг MF отсутствует (значит, что больше фрагментов пакета не будет (пример на скриншоте выше).

Промежуточный пакет иметь и ненулевое смещение, и установленный флаг MF.

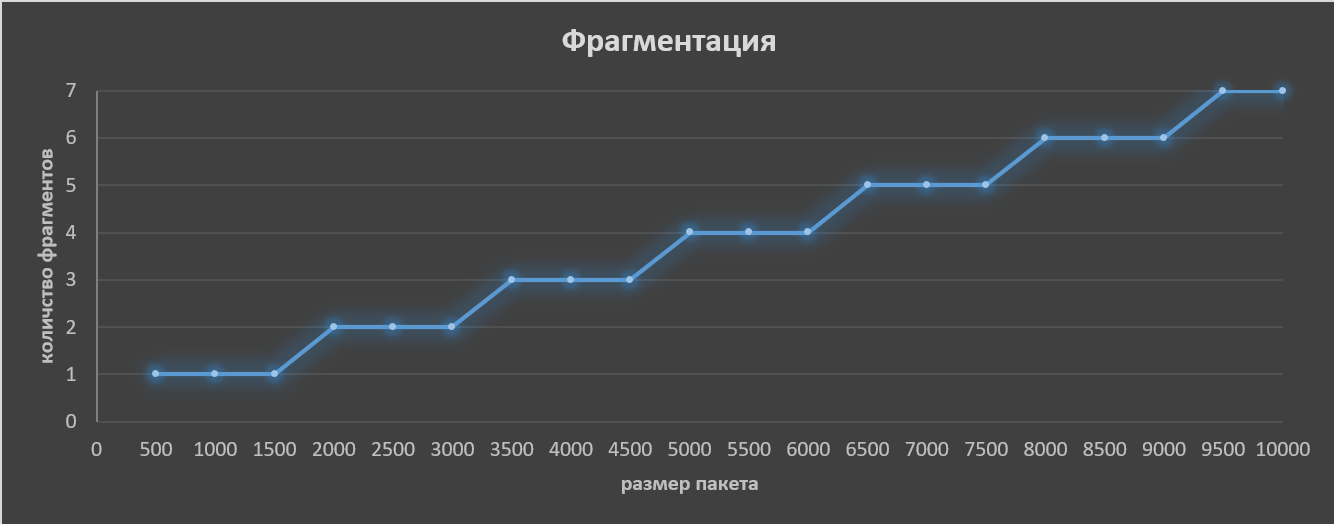


1. Чему равно количество фрагментов при передаче ping-пакетов?

Количество фрагментов зависит от соответствующего соединению MTU (Maximum Transmission Unit – максимальный размер передаваемого блока), а также от размера пакетов.

Необходимо помнить, что служебная информация будет продублирована в каждом из фрагментов.

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

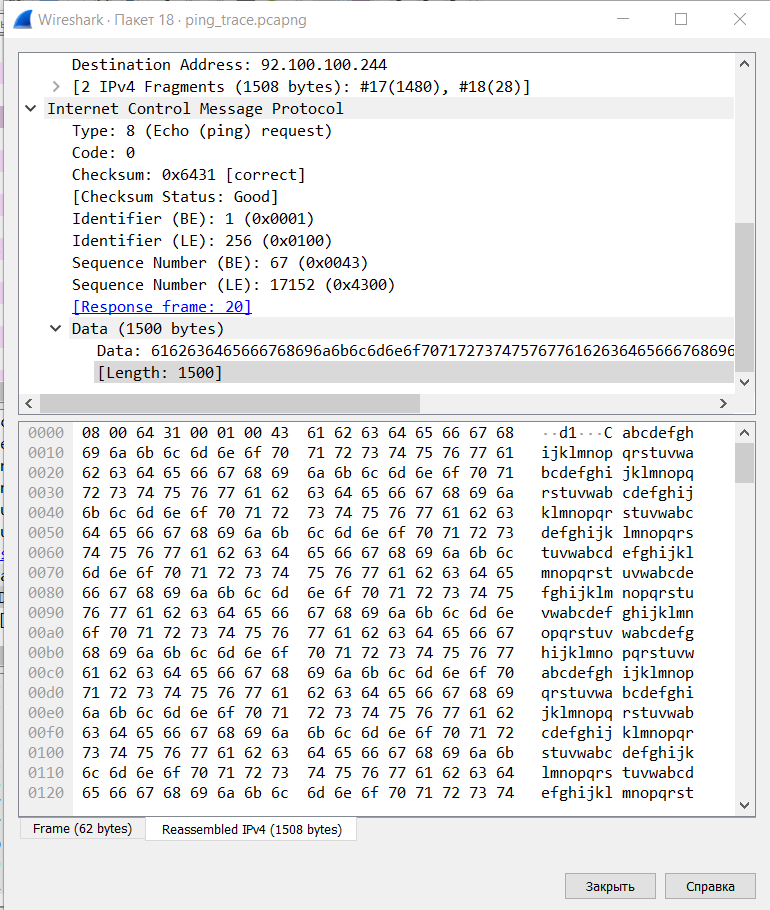


1. Как изменить поле TTL с помощью утилиты ping?

Добавить ключ –i, аргументом которого будет срок жизни пакета в миллисекундах.

1. Что содержится в поле данных ping-пакета?

Повторяющийся до достижения, заданного размере пакета английский алфавит.

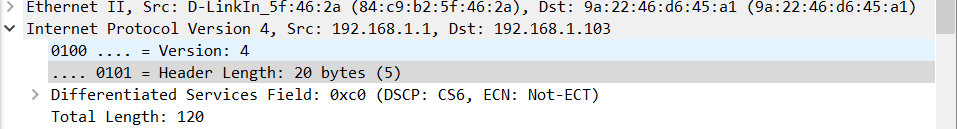


## Анализ трафика утилиты tracert (traceroute)

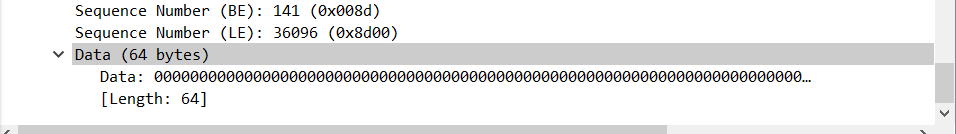
tracert -d kulbakoay.ddns.net

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

20 в заголовке

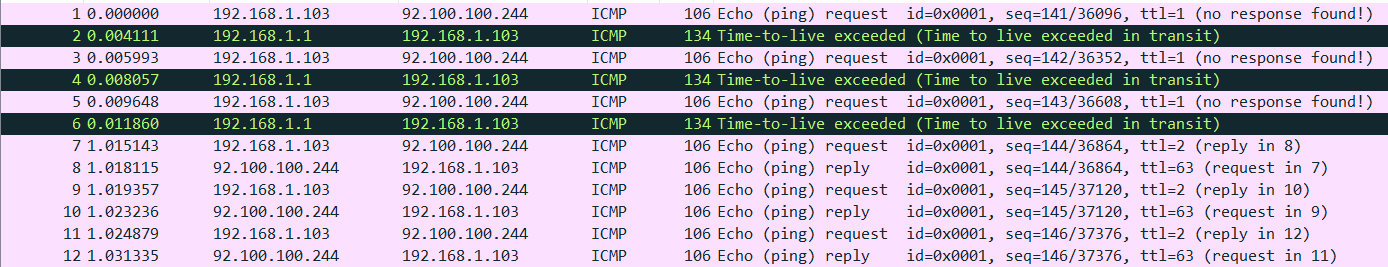


64 в данных



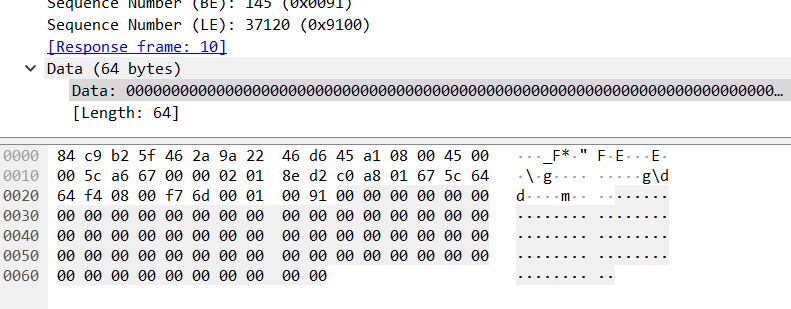
1. Как и почему изменяется поле TTL в следующих друг за другом ICMP-пакетах tracert?

Утилита tracert последовательно отправляет пакеты с увеличивающимся на 1 TTL, чтобы получать от каждого промежуточного узла IP-адрес, тем самым построив карту маршрута.



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

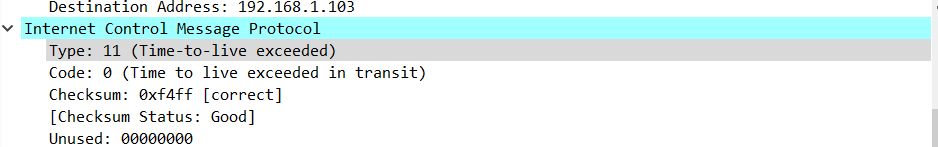
Содержимым поля data - здесь это нули, а в ping был алфавит.



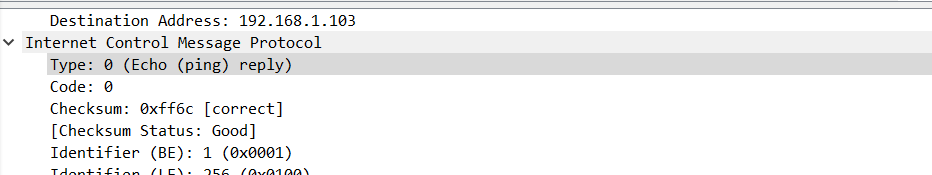
1. Чем отличаются полученные пакеты «ICMP reply» от «ICMP error» и зачем нужны оба этих типа ответов?

Оба типа нужны чтобы различать причину истечению TTL – в случае успешного достижения хоста приходит reply, а в случае ошибки error.

Пример неудачи:



Пример удачи:



1. Что изменится в работе tracert, если убрать ключ “-d”? Какой дополнительный трафик при этом будет генерироваться?

Ключ -d скрывает отображение доменных имён промежуточных хостов.



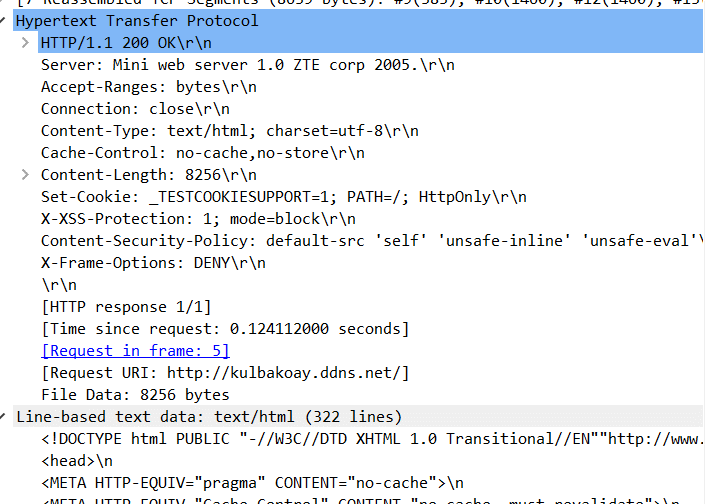
## Анализ HTTP трафика

curl -I -X GET http://kulbakoay.ddns.net/

curl -I -X GET --header "If-Modified-Since: Sun, 02 May 2021 19:14:39 GMT" http://kulbakoay.ddns.net/

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).

Было выполнено 2 запроса к серверу в несколько попыток через браузеры Firefox, Chrome и Edge, а также с помощью скрипта. Во всех случаях, сервер возвращал ответ 200 и текст-документа.



Как можно видеть в тегах ответа, запрашиваемая страница не кэшируема. Мной также были протестированы [www.google.com](http://www.google.com), [www.vk.com](http://www.vk.com), [www.facebook.com](http://www.facebook.com), но во всех случая возвращался ответ 200. Скорее всего связано это с тем, что современные страницы, основанные на реактивных технологиях изменения документов через js, CDN и получения данных через ajax, опираются на кэширование со стороны клиента (<https://stackoverflow.com/a/2222735>).

## Анализ DNS трафика

ipconfig /flushdns

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

После очистки кэша, адрес необходимого нам сайта отсутствует в локальной памяти и нужно сначала получить с DNS-сервера, где из таблиц сопоставления будет получен IP по домену.

1. Какие бывают типы DNS-запросов?

* Прямой - преобразование домена в IP-адрес.
* Обратный – преобразование IP-адреса в домен.
* Рекурсивный – DNS-сервер опрашивает другие сервера, пока не найдёт ответ или не обнаружит, что домен не существует.
* Итеративный – тоже самое, что рекурсивный, но также допускается выполнение поиска клиентом.

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

Если встраиваемое в документ изображение находится на другом сервере (тоже самое произойдёт при использовании html-тега iframe или при подключении ресурсов страницы с помощью CDN).

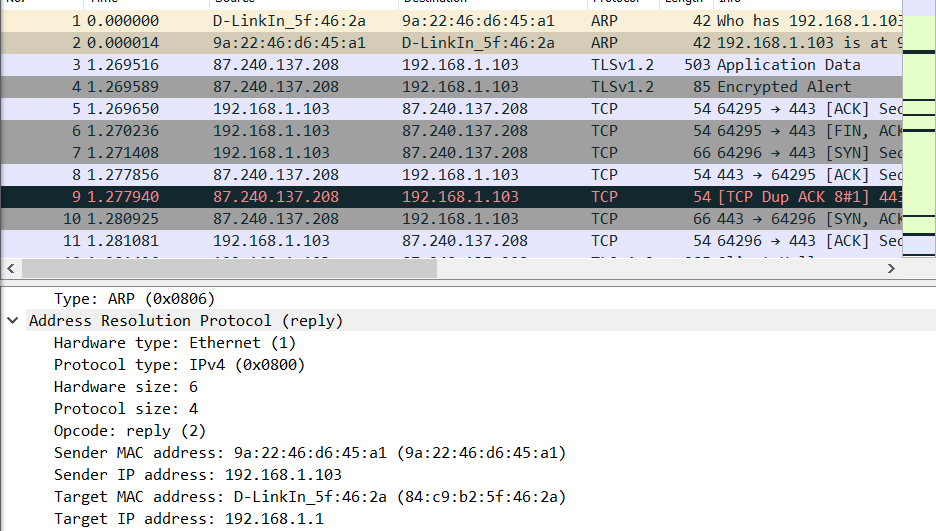
## Анализ ARP трафика

netsh interface ip delete arpcache

arp -a

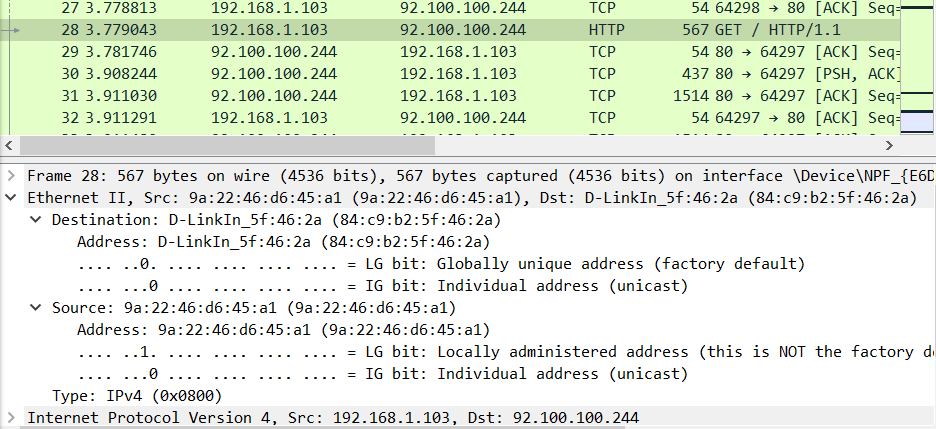
1. Какие МАС-адреса присутствуют в захваченных пакетах ARPпротокола? Что означают эти адреса? Какие устройства они идентифицируют?

Присутствую MAC-адреса устройств отправителя и получателя.



1. Какие МАС-адреса присутствуют в захваченных HTTP-пакетах и что означают эти адреса? Какие устройства они идентифицируют?

Те же самые, что и в ARP запросе, только теперь они служат для идентификации отправителя и получателя HTTP-пакета. На каждой стороне мы можем видеть маршрутизатор локальной сети и непосредственно компьютер.



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

IP-адрес источника необходимом получателя для отправки ответа.

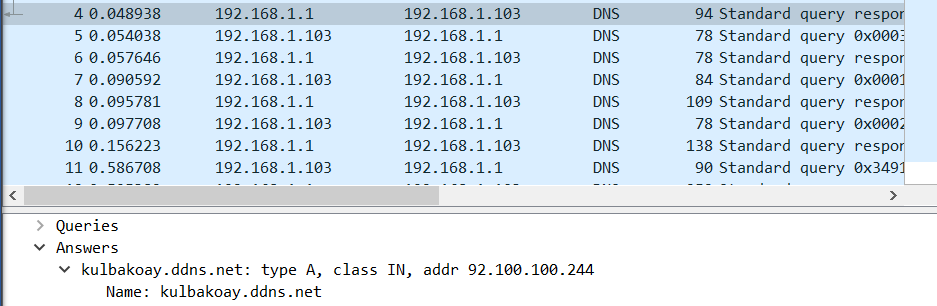
## Анализ трафика утилиты nslookup

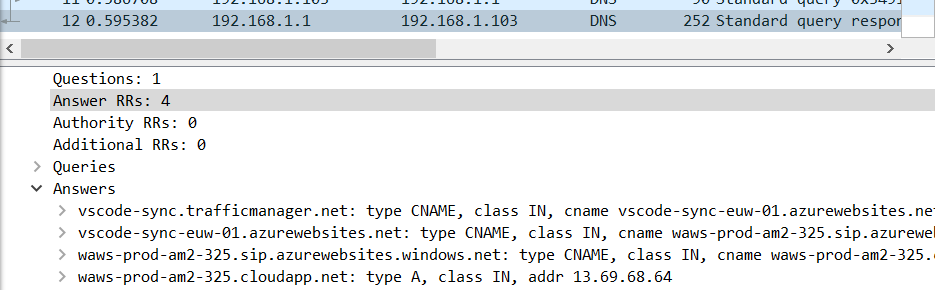
nslookup kulbakoay.ddns.net

nslookup -type=NS kulbakoay.ddns.net

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

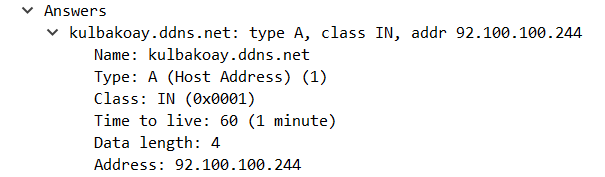
В п.2 DNS-ответе содержится IP сайта требуемого по имени сайта, в п.4 имена авторитативных серверов.





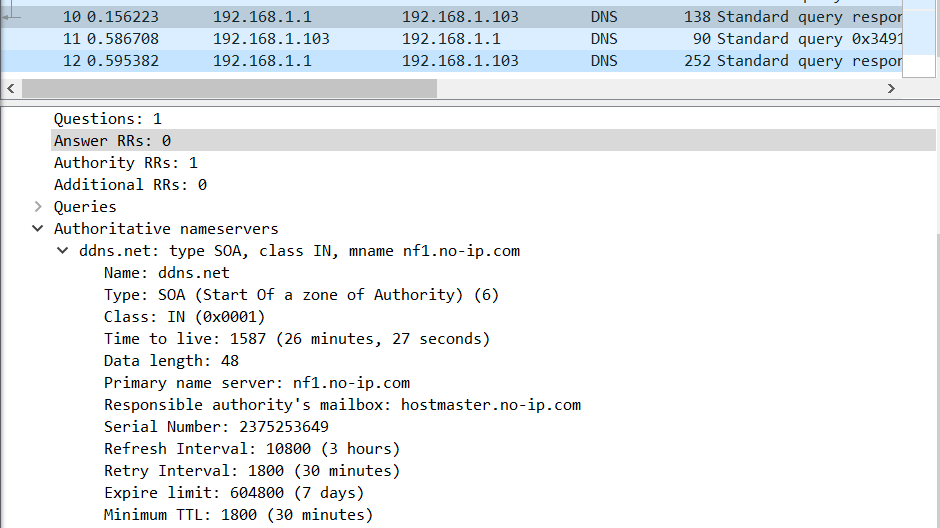
1. Что содержится в поле «Answers» DNS-ответа?

Имя хоста, тип и класс записи, TTL, длина поля данных, IP-адрес запрашиваемого хоста.



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

Авторитарной сервер лишь один – поставщик услуг ddns.



## Анализ FTP трафика

Для поиска подходящего ftp-сервера был взят список из 3329 серверов с <https://scientifically.info/publ/7-1-0-52>. С помощью скрипта:

if (args.size() > 2) throw Exception("You should set search pattern as an argument")

args[0].toList().with { *pattern* ->

    def sites = new File("ftp\_db.txt").readLines()

        .collect {

            def links = it.trim().replaceAll("\\s+", " ").split(" ")

            [links[1], links[3], links[5]]

        }

        .flatten()

        .findAll { it.toList().containsAll(pattern) }

    sites.each {  println it }

    println "Total = ${sites.size()}"

}

были отобраны подходящие согласно инициалам KAY сервера, всего = 57. И многих их них оказались уже нерабочими. Тем не менее, попался и рабочий – <ftp://koyanet.lv>, где хранятся дистрибутивы Linux и различные сопутствующие утилиты.

set f=ftp\_commands.txt

echo "

open koyanet.lv 20

anonymous

anon

get debian/README.html ~\

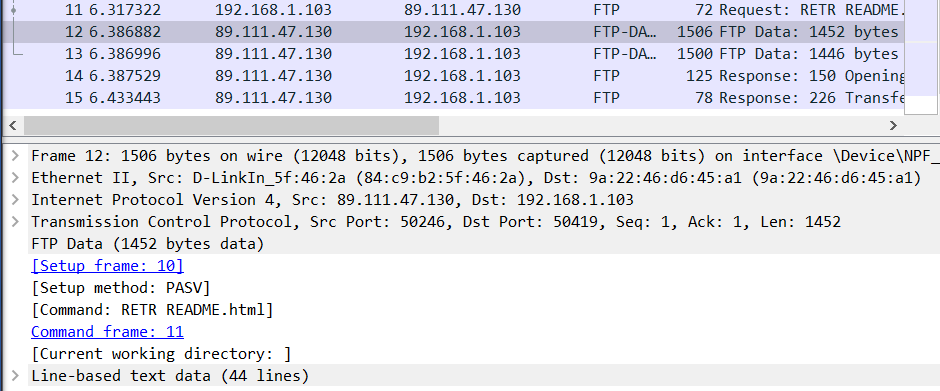
" > %f%

ftp -s:%f%

del %f%

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

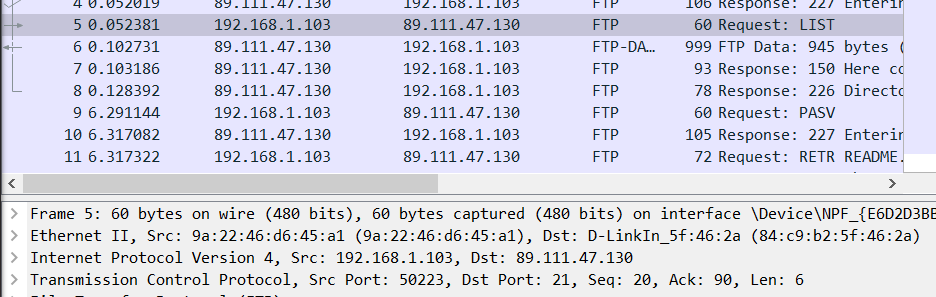
Размер может быть разным, в моём случае он был ограничен MTU Ethernet и равнялся 1452 байтам.



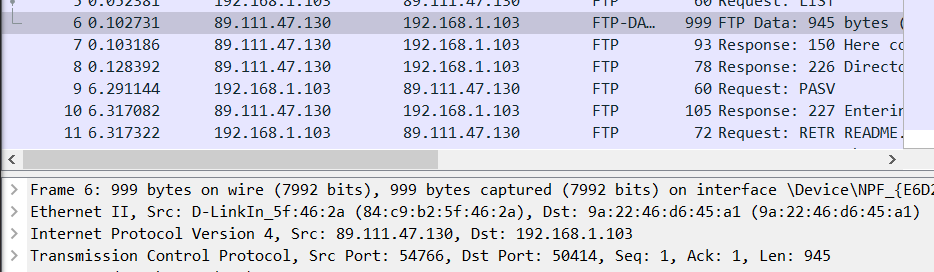
1. Как выбирается порт транспортного уровня, который используется для передачи FTP-пакетов?

Для FTP используется порт 21, передающий управляющие сообщения. Для FTP-DATA может быть выбран любой порт, но по умолчанию используется 20.

Порт 21 на транспортном уровне:



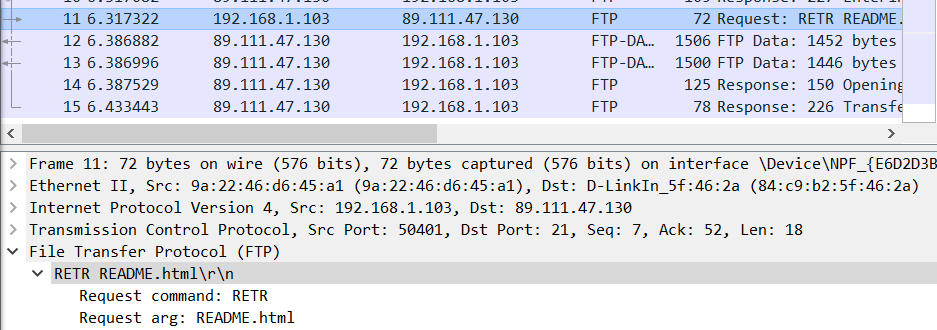
Свободный порт для передачи файлов:



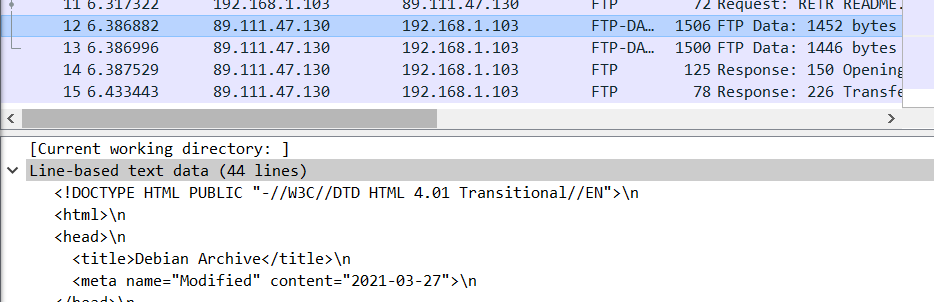
1. Чем отличаются пакеты FTP от FTP-DATA?

FTP передаёт команды серверу, а FTP-DATA работает с файлами.

Запрос html-файла для загрузки:



Непосредственно загрузка:



## Анализ DHCP трафика

for /l %%i in (1, 1, 2) do (

    ipconfig /release

    ipconfig /renew

)

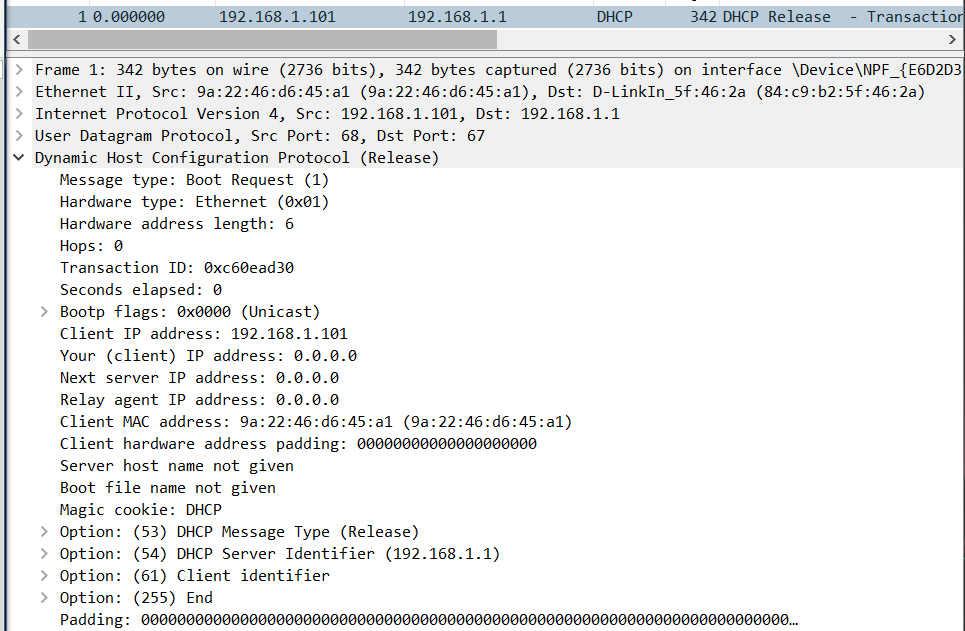
1. Чем различаются пакеты «DHCP Discover» и «DHCP Request»?

Пакеты различаются назначением. Discover -запрос поиска, рассылается на все устройства локальной сети, для поиска DHCP-сервера. Если клиент удачно выберет предложенный «DHCP Offer», то отправит «DHCP Request» - этим сообщением он принимает предлагаемый адрес и уведомляет DHCP-сервер об этом.

1. Как и почему менялись MAC- и IP-адреса источника и назначения в переданных DHCP-пакетах. Каков IP-адрес DHCP-сервера?

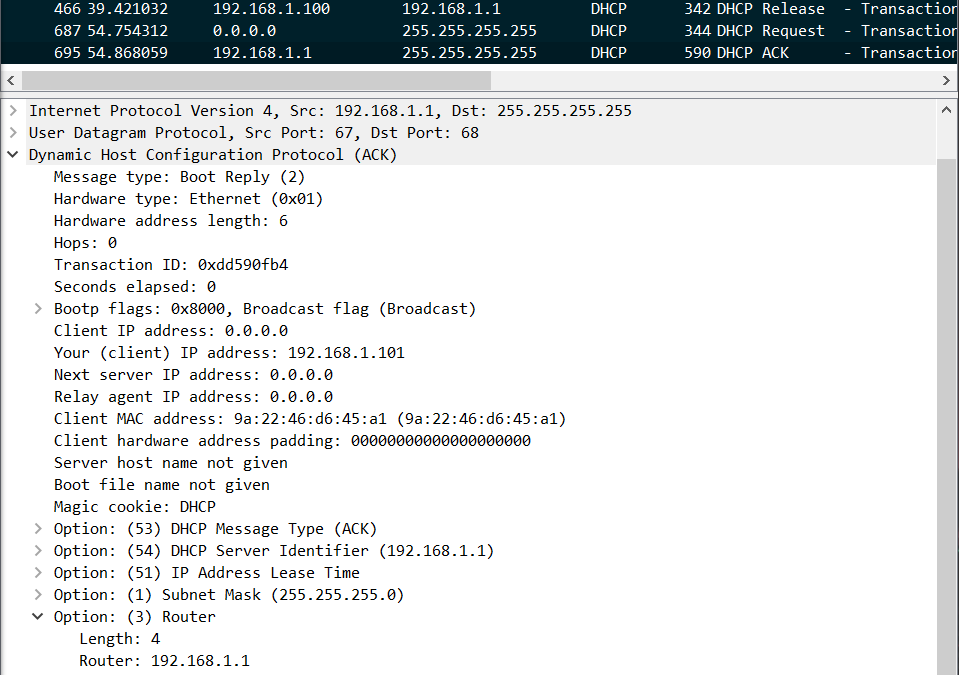
Изначально у отправителя отсутствует IP (0.0.0.0), есть только MAC-адрес, по которому DHCP-сервер отправит ответ Адреса назначения являются широковещательными, чтобы уведомить все устройства, но ответить должен только DHCP-сервер. При отправке Offer или ACK пакетов, адреса источника соответствуют адресам DHCP-сервера, адреса назначения широковещательные.

Пример Release пакета:



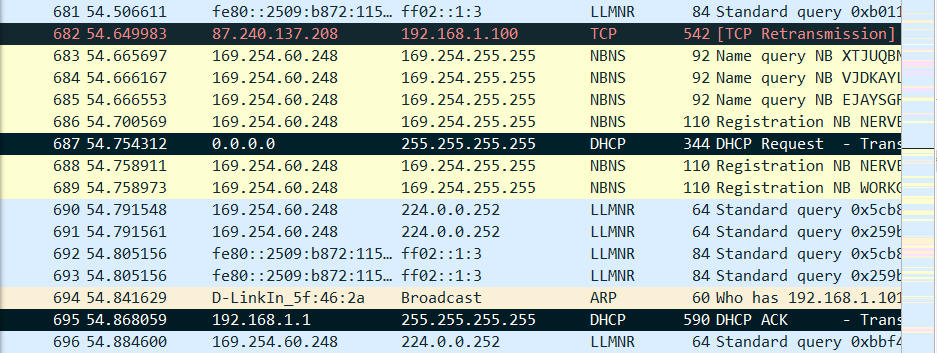
1. Каков IP-адрес DHCP-сервера?

В качестве DHCP-сервера выступает роутер, чей адрес == 192.168.1.1



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

Будут отображаться все захваченные пакеты. Современная версия Wireshark также имеет фильтр dhcp.



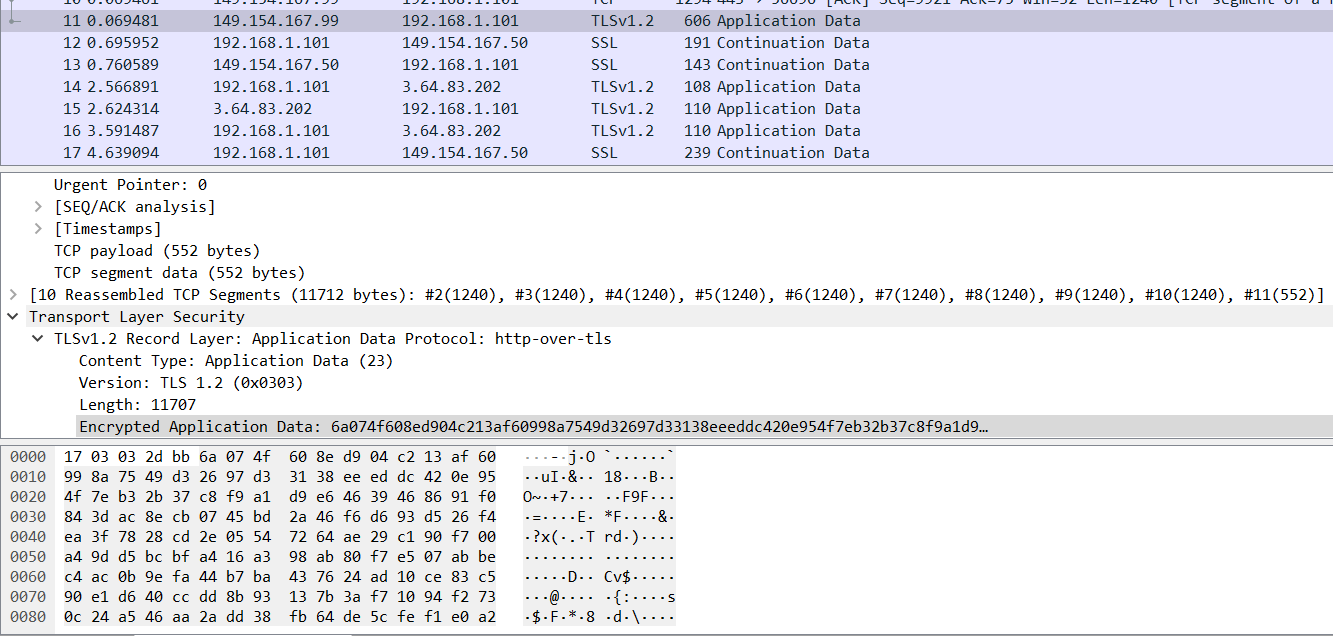
Временная диаграмма последовательности обмена DHCP:



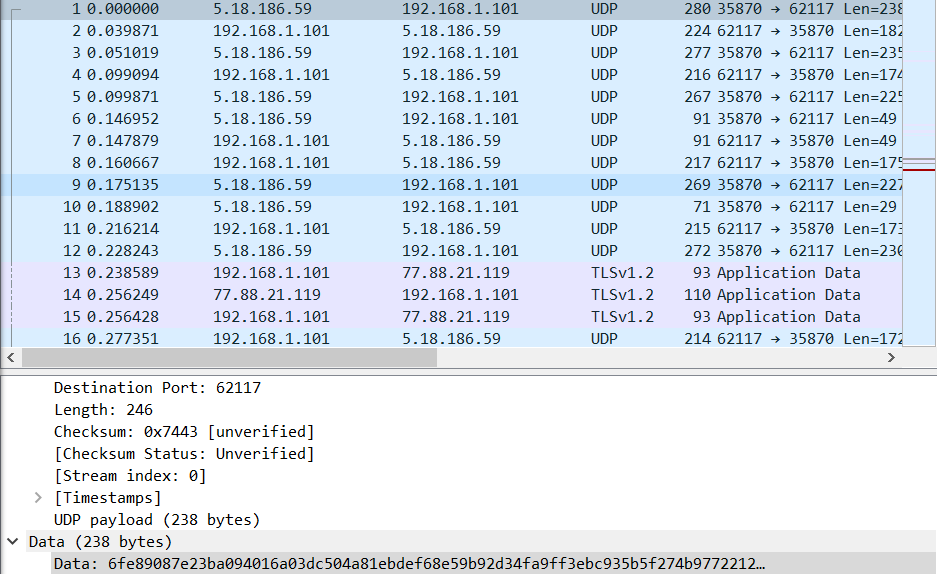
## Анализ Telegram трафика

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

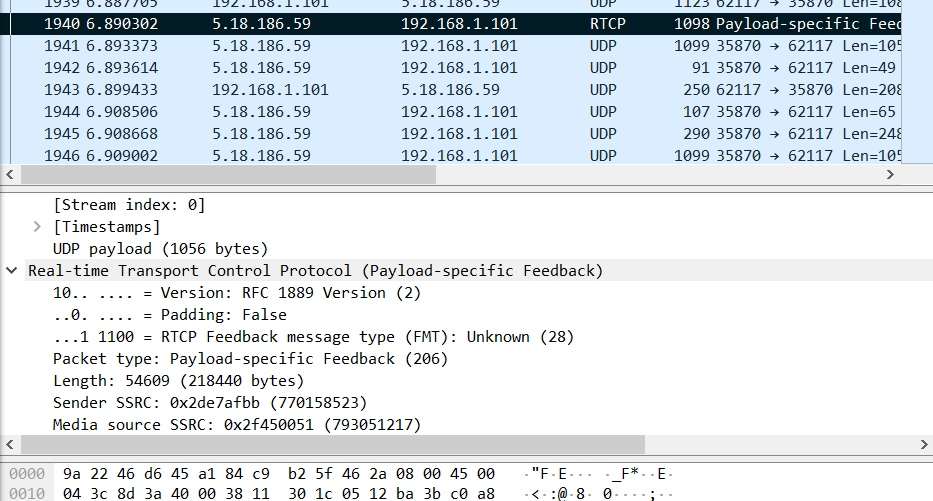
Для передачи текстовых сообщений Telegram использует TLS (в настройках можно изменить это поведение, обернув пакеты в MTProto – собственный протокол, для препятствования DPI). Что интересно, версия 1.2, хотя уже давно существует и активно используется другими приложениями 1.3 (пример, Discord).



Аудио-звонки отправляют, как и ожидалось, UDP-пакеты.



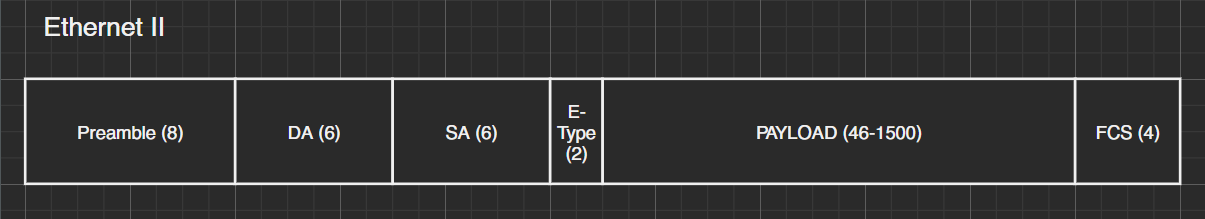
Для передачи видео используется RTCP протокол.



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

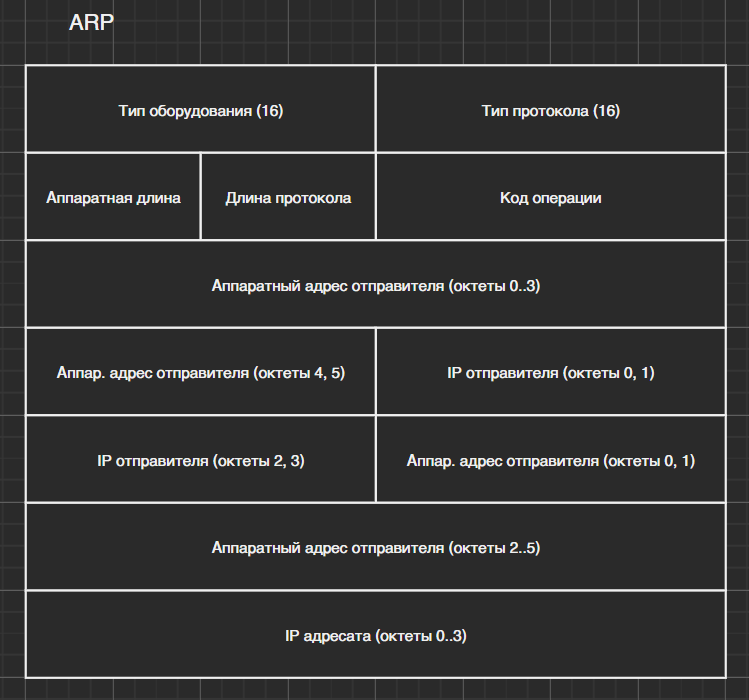
Wireshark поддерживает фильтрацию по протоколу, следовательно в случае Telegram нужно использовать: TLS – для текста, UDP – для звука, RTCP – для видео

# Структуры



* **Preamble** - последовательность бит, по сути, определяющая начало Ethernet фрейма.
* **Destination Address** - MAC назначения
* **Source Address** - MAC отправителя
* **E-TYPE** – Идентификатор типа L3 протокола
* **Payload** – данные
* **Frame Check Sequences** - значение используемое для выявления ошибок передачи.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_



Функционально, ARP делится на две части. Одна - определяет физический адрес при посылке пакета, другая отвечает на запросы других машин. ARP-таблицы имеют динамический характер, каждая запись в ней "живет" определенное время после чего удаляется. Менеджер сети может осуществить запись в ARP-таблицу, которая там будет храниться "вечно". ARP-пакеты вкладываются непосредственно в Ethernet-кадры.

**HA-Len** - длина аппаратного адреса; **PA-Len** - длина протокольного адреса. **Тип оборудования** — это тип интерфейса, для которого отправитель ищет адрес; код содержит 1 для Ethernet.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_



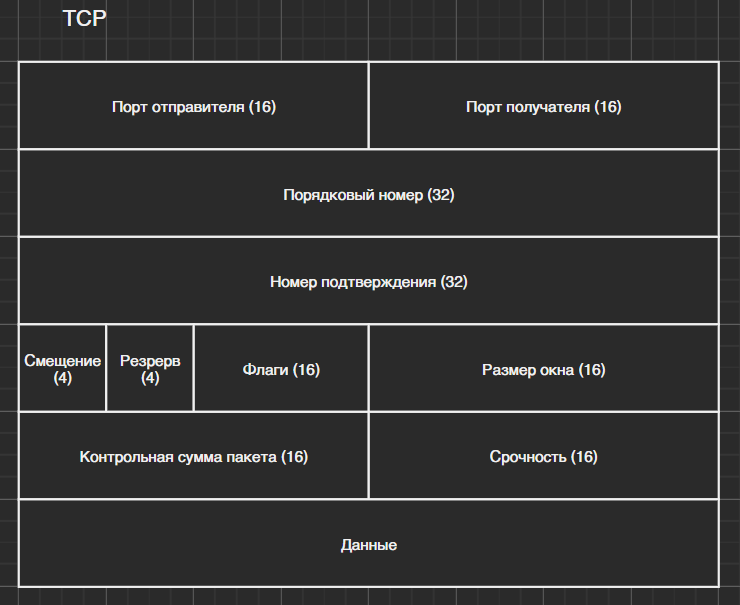
* **Версия** - первым полем заголовка пакета является версия протокола.
* **Internet Header Length** - следующие четыре бита содержат размер заголовка пакета. Поскольку число опций не постоянно, указание размера важно для отделения заголовка от данных.
* **Differentiated Services Code Point** - используется для разделения трафика на классы обслуживания, например, для установки чувствительному к задержкам трафику.
* **Размер пакета** - полный размер пакета в байтах, включая заголовок и данные.
* **Идентификатор** - используется для идентификации фрагментов пакета, если он был фрагментирован.
* **Флаги** - поле, содержащее флаги контроля над фрагментацией.
* **Смещение фрагмента** - указывает смещение поля данных текущего фрагмента относительно начала поля данных первого фрагментированного пакета в блоках по 8 байт.
* **Time to Live** - определяет максимальное количество маршрутизаторов на пути следования пакета. Наличие этого параметра не позволяет пакету бесконечно ходить по сети.
* **Протокол** - указывает, данные какого протокола содержит пакет (например, TCP).
* **Контрольная сумма** - используется для проверки целостности заголовка. Каждый хост или маршрутизатор сравнивает контрольную сумму заголовка со значением этого поля и отбрасывает пакет, если они не совпадают.
* **Адрес источника** - адрес отправителя пакета.
* **Адрес назначения** - адрес получателя пакета.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_



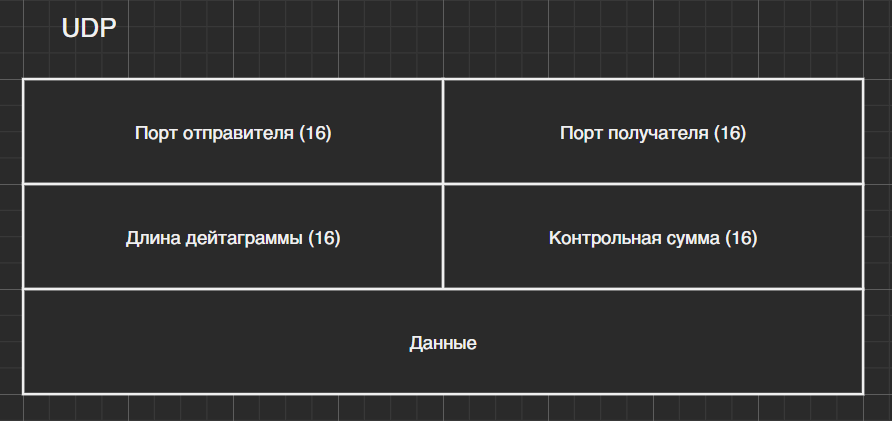
* **Тип** – определяет действие по диагностики.
* **Код** – код ошибки.
* **Контрольная сумма** – тот же принцип, что и у IP.
* **Данные** – фрагмент пакета, при передаче которого произошла ошибка.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_



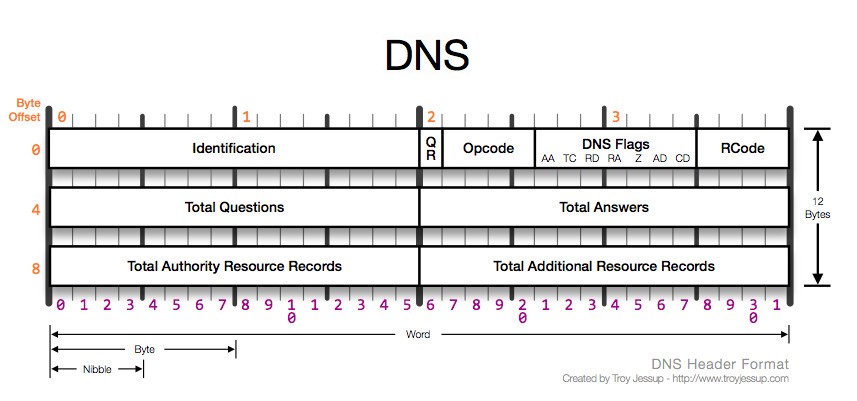
* **Порт источника и порт приемника** – точки, в которых процессы верхнего уровня принимают услуги TCP.
* **Порядковый номер** – обычно номер первого байта в сообщении. Может также использоваться для обозначения исходного порядкового номера в передаче.
* **Номер подтверждения** – порядковый номер следующего байта данных, который ожидает получить приемник.
* **Сдвиг данных** – Число 32-разрядных слов в заголовке TCP.
* **Резервные** – область, зарезервированная для использования в будущем.
* **Флаги** – различная управляющая информация, в том числе биты SYN, ACK и FIN.
* **Окно** – размер приемного окна (буфера памяти) приемника.
* **Контрольная сумма** – показывает, не был ли заголовок поврежден при передаче.
* **Указатель срочности** – указывает на первый байт срочных данных в пакете.
* **Параметры** – различные дополнительные параметры TCP.
* **Данные** – информация верхнего уровня.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_



* **Порт источника и порт приемника** – точки, в которых процессы верхнего уровня принимают услуги UDP.
* **Длина сообщения** равна числу байт в UDP-дейтограмме, включая заголовок.
* Поле **контрольная сумма** содержит код, полученный в результате контрольного суммирования UDP-заголовка и поля данные.
* **Данные** – полезная нагрузка пакета.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_



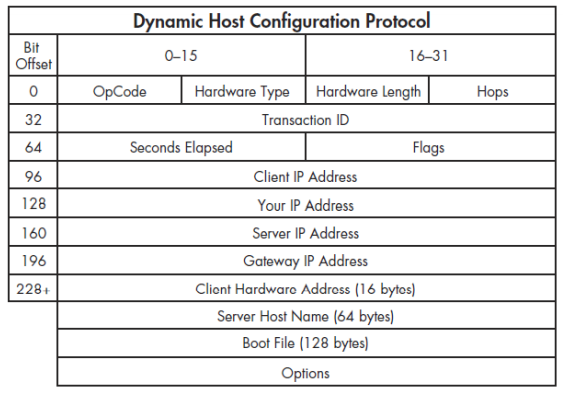
**Identification** устанавливается клиентом и возвращается сервером. Это поле позволяет клиенту определить, на какой запрос пришел отклик.

16-битовое поле **flags** поделено на несколько частей, как показано на рисунке 3.



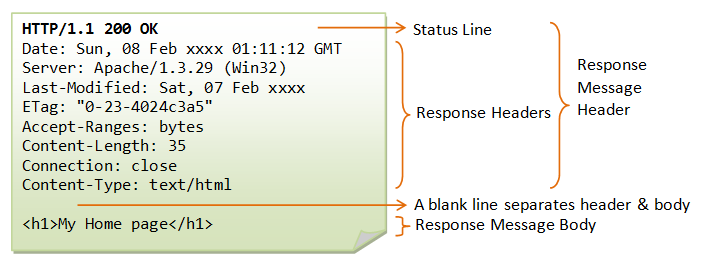
Следующие четыре 16-битных поля указывают на количество пунктов в четырех полях переменной длины, которые завершают запись. В запросе **number of questions** обычно равно 1, а остальные три счетчика равны 0. В отклике **number of answers** по меньшей мере равно 1, а оставшиеся два счетчика могут быть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

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_



* **OpCode** - тип DHCP-сообщения. Если значение 0×01 – запрос к серверу, иначе - оно являет ответом DHCP-сервера.
* **Hardware Type** - тип адреса на канальном уровне. DHCP может работать поверх различных протоколов на канальном уровне, поэтому нужно указывать на каком именно.
* **Hardware Length** - длина аппаратного адреса в байтах.
* **Hops** - количество промежуточных маршрутизаторов, которые находятся на пути между клиентом и сервером.
* **Transaction ID** – идентификатор процесса получения IP-адреса.
* **Seconds Elapsed** - время в секундах с момента начала процесса получения IP.
* **Flags** - поле для флагов и параметров протокола.
* **Client IP Address** - IP-адрес клиента. Не пусто, если у клиента уже есть IP и он хочет продлить время аренды IP-адреса.
* **Your ID Address** – предложенный DHCP-серверов IP клиенту.
* **Server IP Address** - IP-адрес сервера.
* **Client Hardware Address** – MAC клиента.
* **Server Hostname** – доменное имя сервера (если присуттвует).
* **Boot File** - служит указателем для бездисковых рабочих станций о имени файла инициализации на сервере.
* **Options** – информация для динамической конфигурации хоста.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_



* **Статус** – код ответа и сообщения описывающее, что код значит.
* **Заголовки** – служебная информация.
* **Данные** – основное содержание сообщения.

# Выводы

В ходе выполнения лабораторной работы мною были получены навыки работы с анализатором трафика Wireshark, где были захвачены и изучены пакеты разных протоколов, их расположение по уровням TCP/IP модели, назначение и структура. Сложность работы заключается в объёме информации, в которой очень легко запутаться, а также устаревшим требованиям: к примеру, найти рабочий FTP-сервер, а к тому же с своими инициалами крайне затруднительно (стоит развернуть легковесные сервера для каждого учащегося на Helios, как это сделано для лабораторных других предметов). Проблемы могут возникнуть и с аппаратурой: в моём случае маршрутизатор подключен к PON-розетке оператору, что создаёт дополнительную локальную сеть, которая по каким-то причинам мешала корректной работе маршрутизатора как DHCP-сервера: помогло подключение ноутбука по ethernet-кабелю напрямую к PON.

# Список использованной литературы

1. Т. И. Алиев, В. В. Соснин, Д. Н. Шинкарук – Компьютерные сети и телекоммуникации: задания и тесты – СПБ: СПБГУ ИТМО, 2018. – 112 с.
2. Т. И. Алиев – Сети ЭВМ и телекоммуникации – СПБ: СПБГУ ИТМО, 2011 – 400 с.