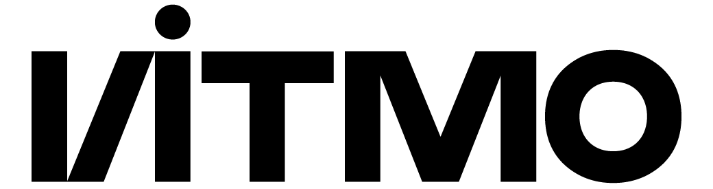
****

**Лабораторная работа №4   
«Анализ трафика компьютерных сетей   
с помощью утилиты Wireshark»**

*по дисциплине: Компьютерные сети*

Выполнил: Неграш Андрей, P33301

Преподаватель: Алиев Тауфик Измайлович

Санкт-Петербург, 2023

Оглавление

[1. Цель работы 3](#_Toc135629607)

[2. Постановка задачи 3](#_Toc135629608)

[3. Анализ трафика утилиты ping 3](#_Toc135629609)

[4. Анализ трафика утилиты tracert 5](#_Toc135629610)

[5. Анализ HTTP-трафика 7](#_Toc135629611)

[6. Анализ DNS-трафика 9](#_Toc135629612)

[7. Анализ ARP-трафика 10](#_Toc135629613)

[Вывод 11](#_Toc135629614)

# Цель работы

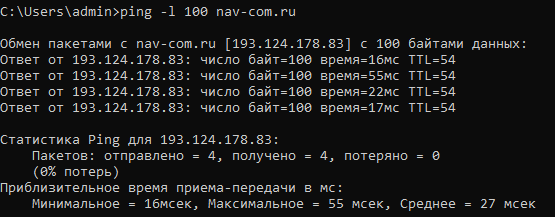
Изучить структуру протокольных блоков данных, анализируя реальный трафик на компьютере студента с помощью бесплатно распространяемой утилиты Wireshark.

# Постановка задачи

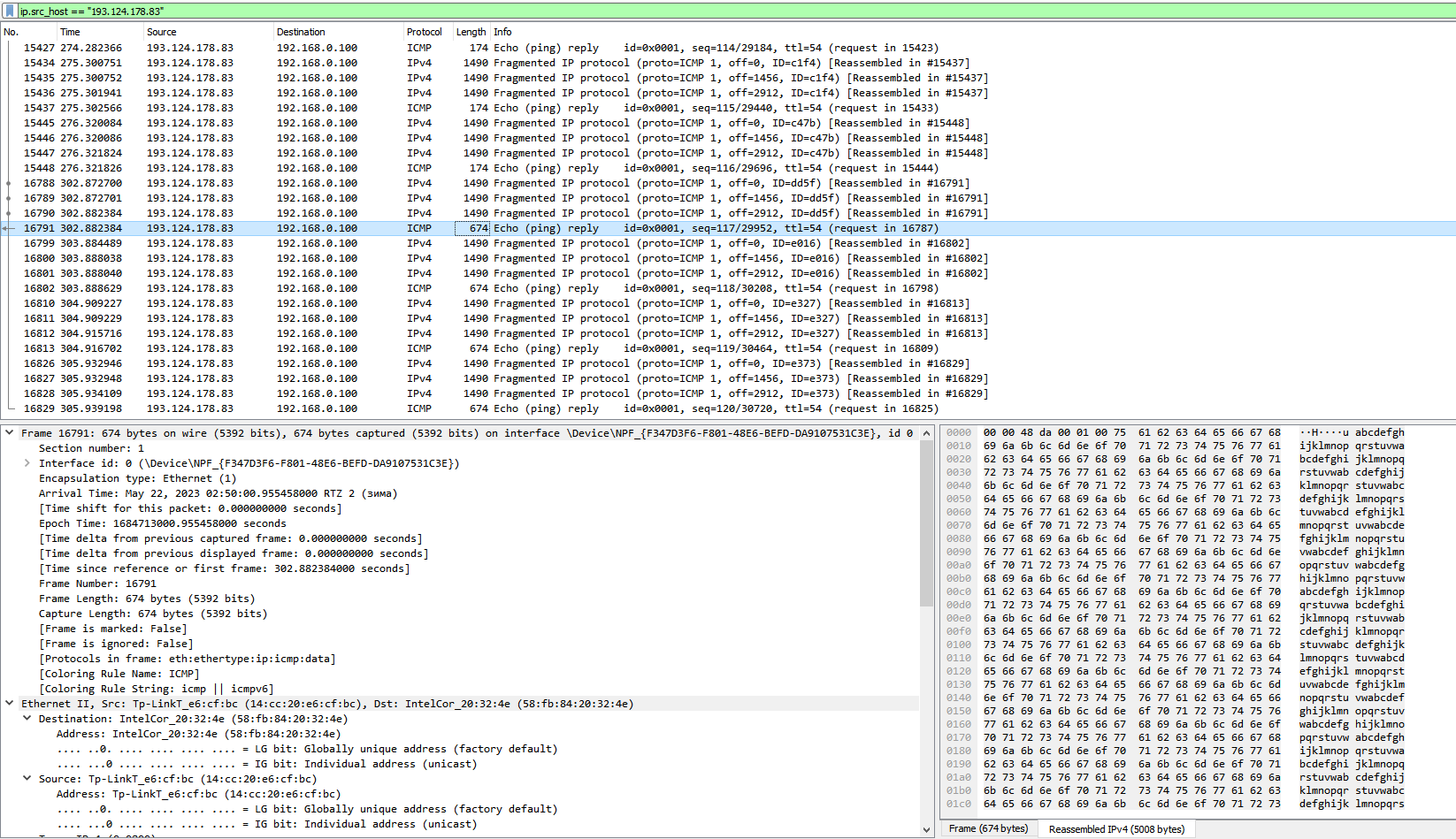
При выполнении данной работы был использован сайт [nav-com.ru](https://nav-com.ru), так как в нём по очереди встречаются мои инициалы в латинской транскрипции. Неграш Андрей Владимирович – Negrash Andrey Vladimirovich – NAV. Также этот сайт принадлежит мне и в какой-то степени является моей домашней страницей.

# Анализ трафика утилиты ping

Команда, запускаемая в командной строке, выглядит следующим образом:



В качестве размера пакета я поочерёдно подставлял значения от 100 до 10000 с шагом 500. Скриншот из утилиты Wireshark для размера пакета 5000 выглядят следующим образом:



Фрагментация исходного пакета происходит, если размер поля данных более 1456 байт. На то, является ли данный пакет конечным или промежуточным, указывает флаг MF (More Fragments) в IPv4-пакетах:

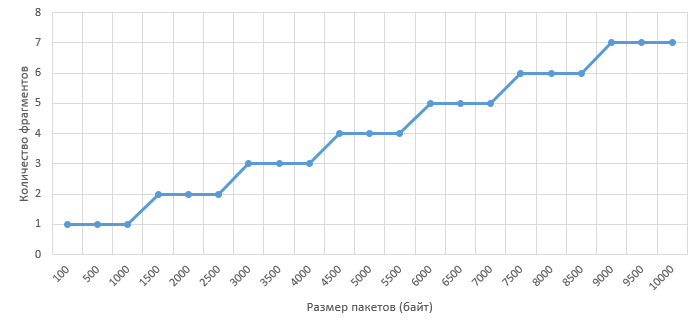
Изображение выглядит как текст, Шрифт, снимок экрана, линия

Автоматически созданное описание

При передаче ping-пакетов количество фрагментов напрямую зависит от размера передаваемого пакета. Без фрагментации можно передать 1456 байт данных, если же их больше, то они делятся на фрагменты данного объёма и передаются последовательно.

Таблица и график, на котором продемонстрирована зависимость размера пакета от количества фрагментов, на которое разделён каждый ping-пакет:

|  |  |  |  |
| --- | --- | --- | --- |
| **Размер пакета** | **Количество фрагментов** | **Размер пакета** | **Количество фрагментов** |
| 100 | 1 | 5500 | 4 |
| 500 | 1 | 6000 | 5 |
| 1000 | 1 | 6500 | 5 |
| 1500 | 2 | 7000 | 5 |
| 2000 | 2 | 7500 | 6 |
| 2500 | 2 | 8000 | 6 |
| 3000 | 3 | 8500 | 6 |
| 3500 | 3 | 9000 | 7 |
| 4000 | 3 | 9500 | 7 |
| 4500 | 4 | 10000 | 7 |
| 5000 | 4 |  |  |



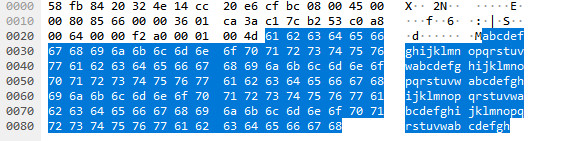
Для изменения поля TTL нужно добавить в запрос флаг «-i». Например:

ping -i 16 -l 5000 nav-com.ru

Примечание: в ОС Linux флаг для изменения TTL – это «-t». Запрос будет выглядеть так:

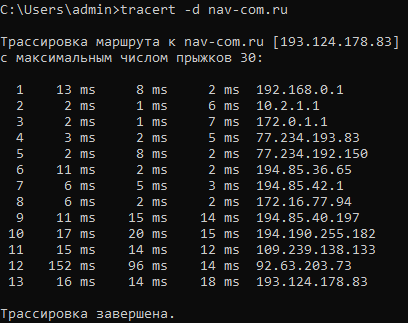
ping -t 16 -s 5000 nav-com.ru

В поле данных ping-пакета находится последовательно выстроенные буквы английского алфавита от a до w (без x, y и z):

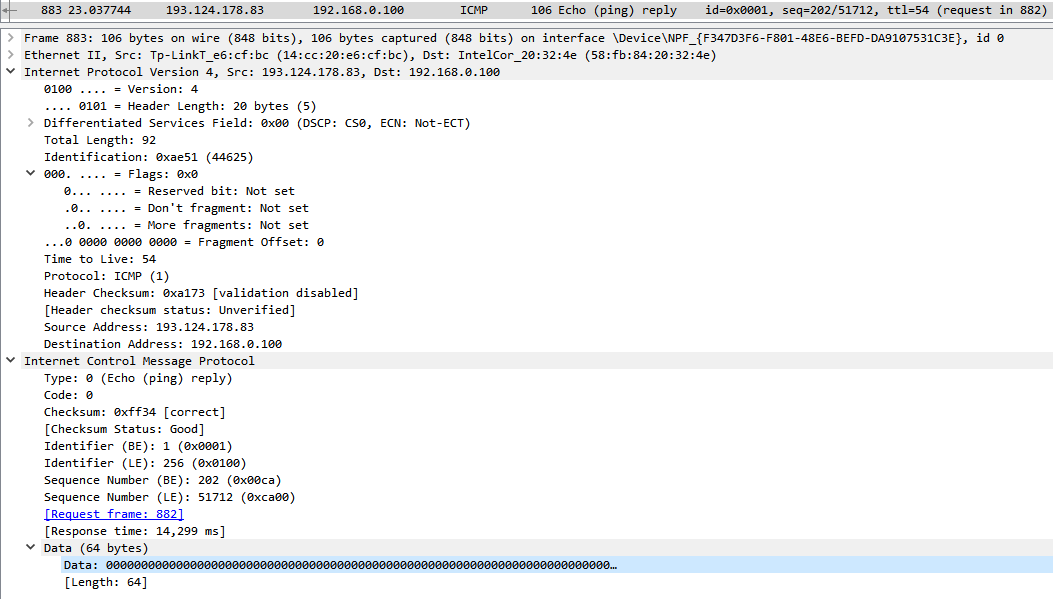


# Анализ трафика утилиты tracert

Команда, запускаемая в командной строке, выглядит следующим образом:



Скриншот из утилиты Wireshark выглядит следующим образом:

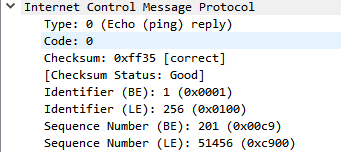


В заголовке IP содержится 20 байт, в поле данных – 64 байт. Это видно на скриншоте выше.

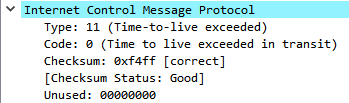
TTL в следующих друг за другом ICMP-пакетах tracert увеличивается на 1 каждые 3 пакета для того, чтобы выявить точное расстояние в хопах до искомого хоста. К тому же это позволяет установить IP-адреса всех встречающихся на маршруте устройств.

ICMP-пакеты, которые генерируются утилитой tracert, отличаются от генерируемых утилитой ping изменяющимся значением TTL и полем данных (здесь оно заполнено нулями).

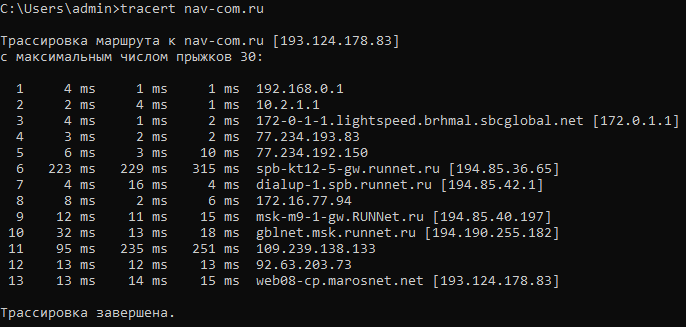
Отличие пакетов «ICMP reply» от «ICMP error» заключается в значении поля Type. ICMP reply нужно для того, чтобы сообщить об успешном достижении хоста, имея тип 0:

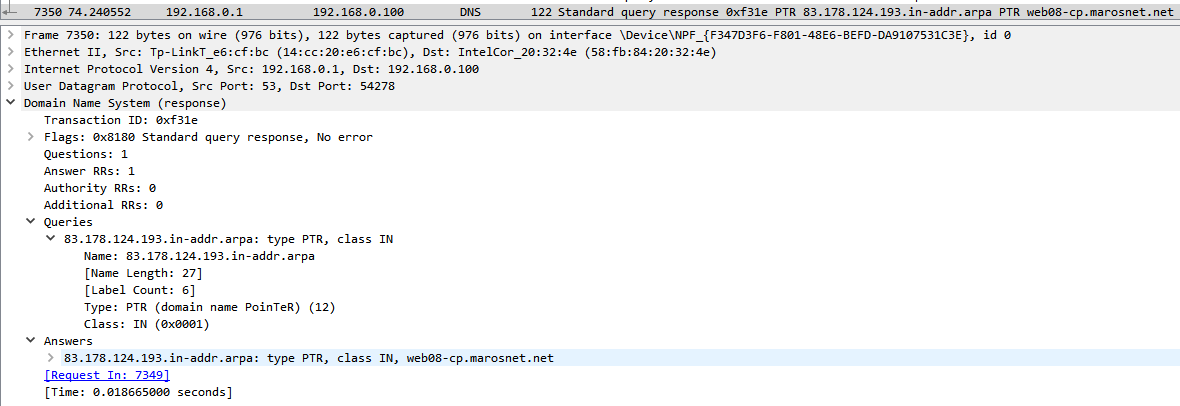


ICMP error – для понимания, что хост ещё не был достигнут и нужно инкрементировать TTL и послать следующий запрос. Такой ответ имеет код 11:



Если убрать ключ «-d», то утилита tracert начнёт посылать DNS-запросы, чтобы преобразовать IP-адреса в доменные имена. Скриншоты выполнения такого запроса:

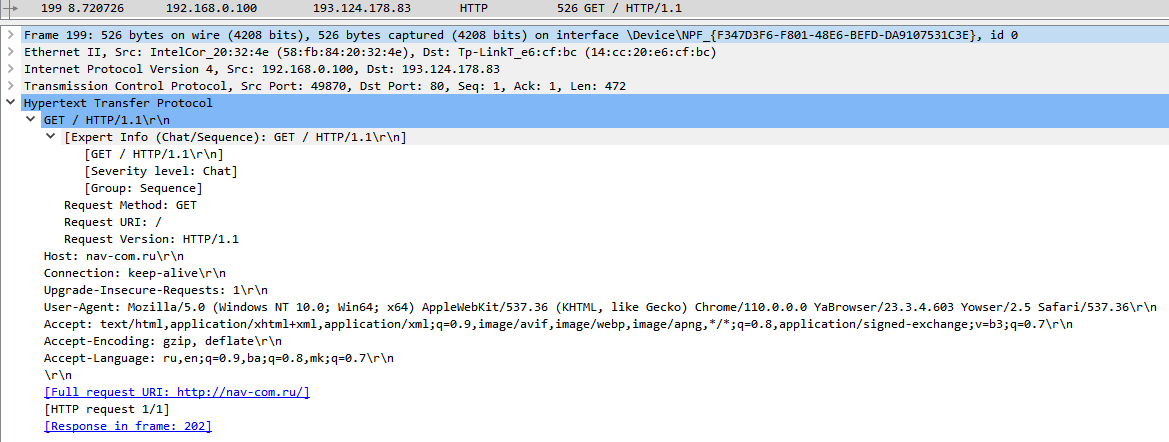


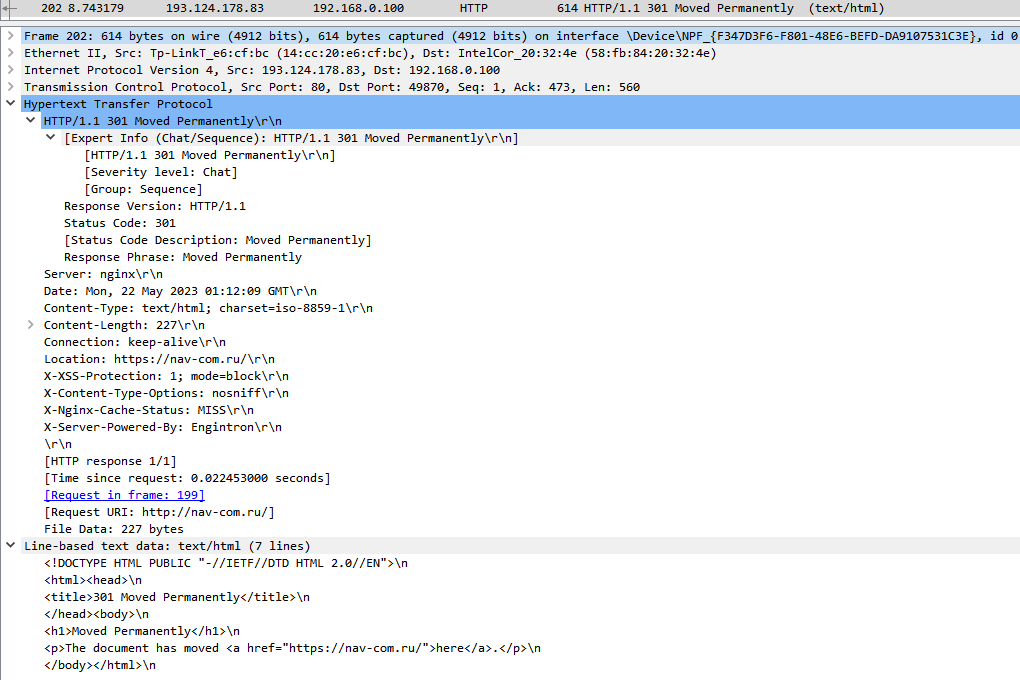


# Анализ HTTP-трафика

Воспользуемся браузером и зайдём на сайт по адресу nav-com.ru с включенной записью данных через Wireshark. Дождёмся загрузки всех данных и выключим запись.

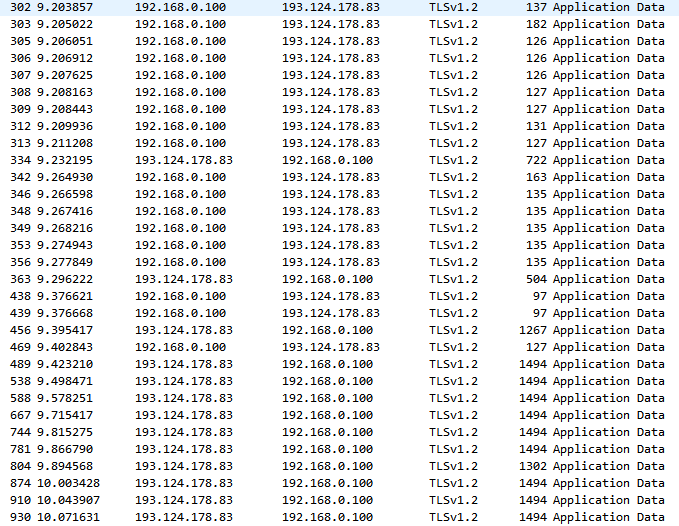
Скриншот GET-запроса в Wireshark выглядит так:

Ответом сервера является следующее сообщение:



Нам был отправлен код 301 с сообщением Moved Permanently. Это произошло по причине автоматического перенаправления сервером с небезопасного протокола HTTP на безопасный HTTPS. О том, как я это делал на сервере, я могу рассказать отдельно, но это не имеет отношения к лабораторной работе.

В теории мы должны были получить большое количество HTTP-ответов с загружаемыми данными, а получили большое количество тех же данных, но в зашифрованном виде и передаваемых по протоколу TLS:



Здесь сначала передаются html-разметка, а затем некоторые js-скрипты (прописанные вверху страницы), картинки и прочие медиаресурсы, а также js-скрипты, которые помещены в самый низ страницы и загружаются асинхронно.

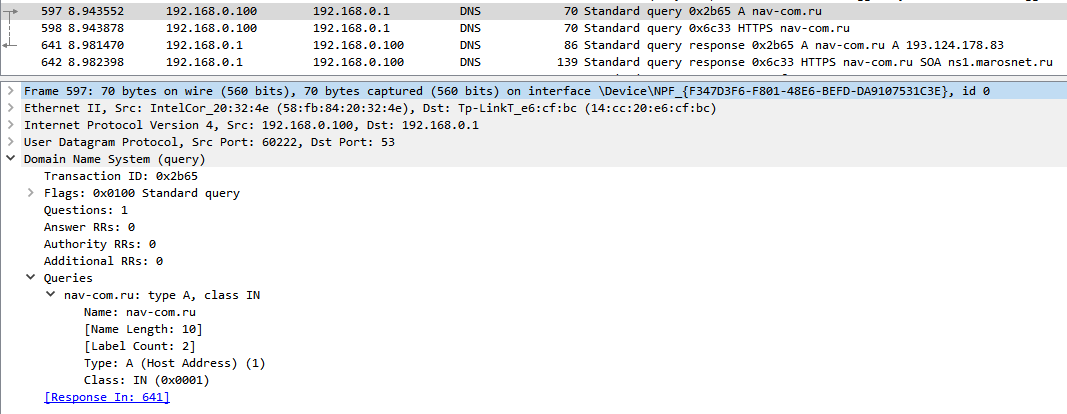
По той же причине при обновлении страницы мы не можем найти HTTP-запрос с нужным нам CONDITIONAL GET-запросом. Он производится с помощью протокола TLS, однако найти его с помощью Wireshark не представляется возможным.

Теоретически сервер бы вернул нам код 304 Not Modified, так как данный ресурс не изменяется и на этом обновление страницы бы завершилось.

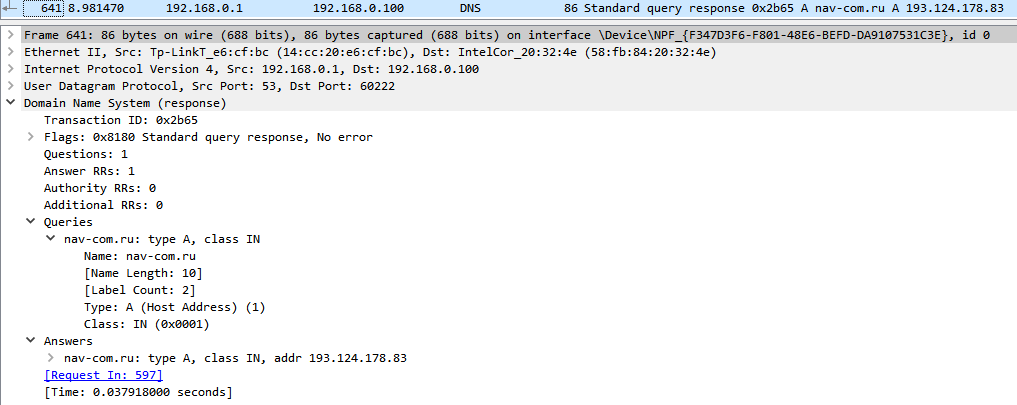
# Анализ DNS-трафика

Отследим и проанализируем трафик протокола DNS для заданного веб-ресурса. Для этого в Wireshark создадим фильтр, очистим кэш DNS и кэш браузера. Запустим сбор пакетов в Wireshark и зайдём на наш ресурс nav-com.ru.

Вид запроса:



Вид ответа:



На первом скриншоте видно два запроса и, соответственно, два ответа на них.

Адрес, на который отправлен DNS-запрос не совпадает с адресом посещаемого сайта, потому что адрес отправки соответствует установленному шлюзу по умолчанию моего компьютера.

Существуют следующие типы DNS-запросов:

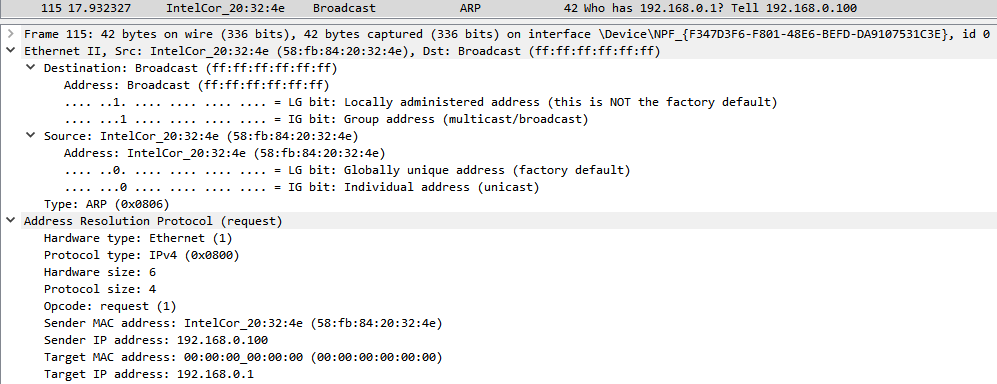
* Рекурсивный – ищет, к какому IP-адресу относится заданный домен
* Инверсивный – ищет, к какому домену относится заданный IP-адрес

Для получения содержащихся на сайте изображений понадобятся дополнительные независимые DNS-запросы, если эти изображения находятся не на том же хосте, что и сама страница, а взяты, например, по ссылке из банка открытых изображений. Тогда будет создан независимый DNS-запрос, который будет отправлен к этому самому банку.

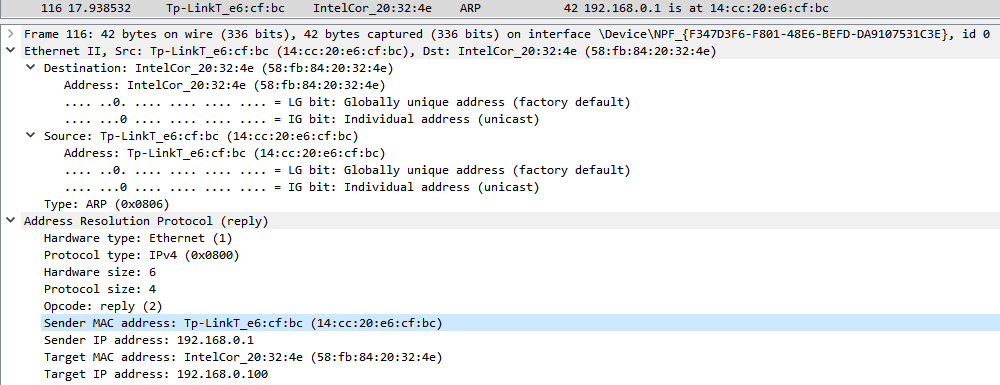
# Анализ ARP-трафика

Отследим и проанализируем трафик протокола ARP. Для этого очистим ARP-таблицу и повторно очистим кэш нашего браузера. Затем запустим сбор трафика в Wireshark и перейдём на наш ресурс nav-com.ru.

Запрос:

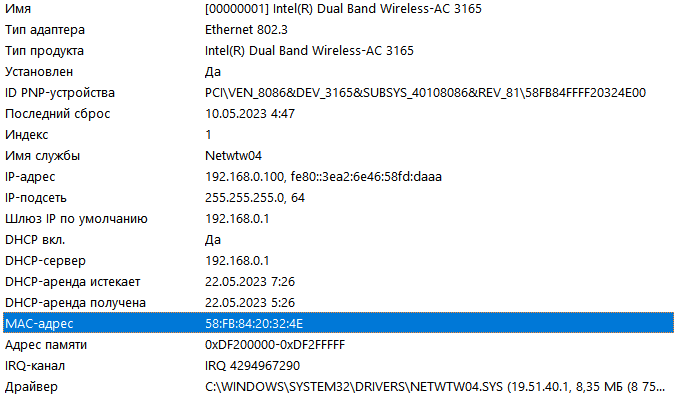


Ответ:

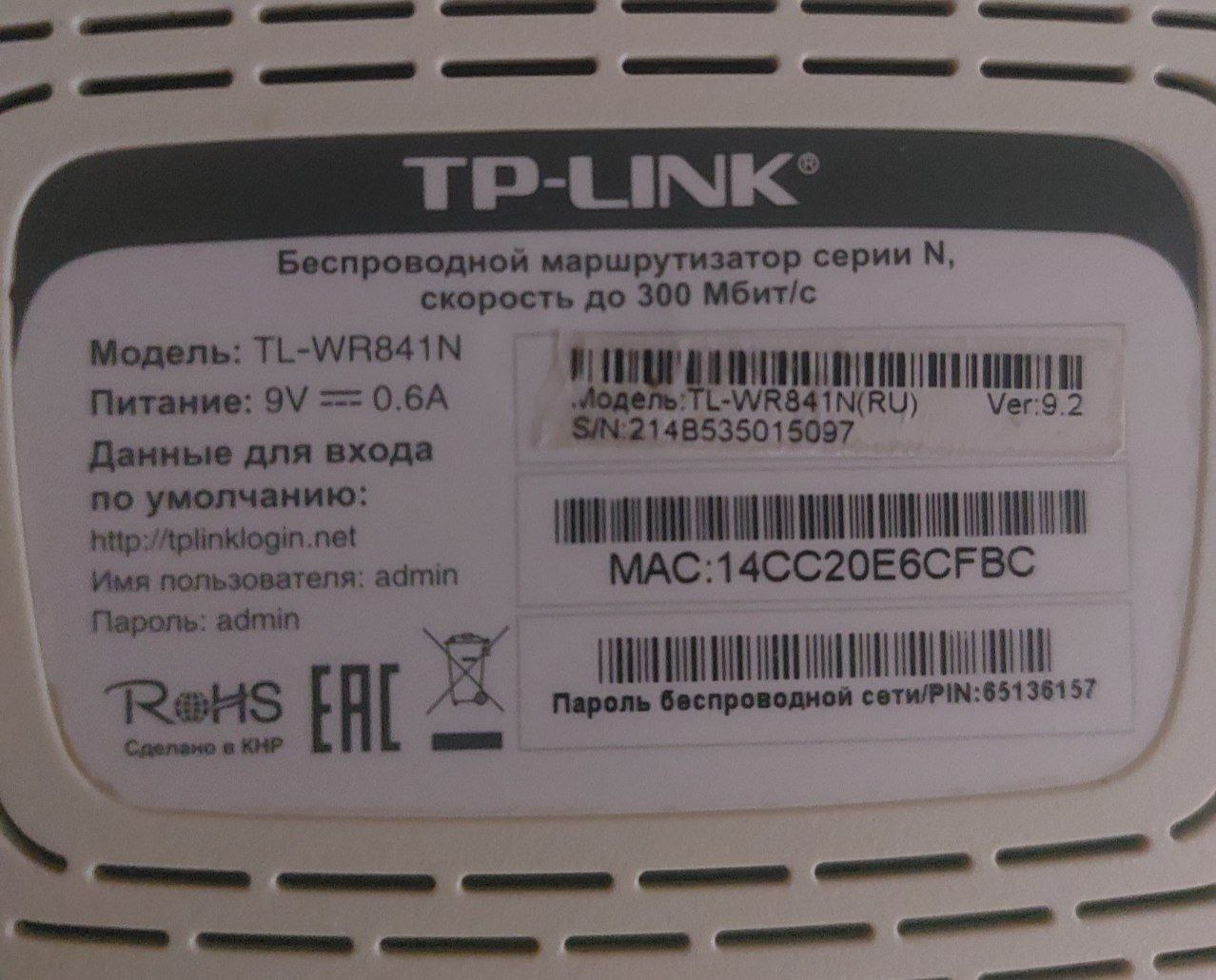


В захваченных пакетах ARP-протокола присутствуют следующие MAC-адреса:

* 58:fb:84:20:32:4e – это MAC-адрес нашего устройства



* 00:00:00:00:00:00 – это MAC-заполнитель, пока не будет получен реальный адрес
* 14:cc:20:e6:cf:bc – это MAC-адрес маршрутизатора (роутера)



В захваченных HTTP-пакетах присутствуют те же самые MAC-адреса.

ARP-запрос содержит IP-адрес источника для того, чтобы можно было сразу заполнить ARP-таблицу у получателя, записав MAC- и IP- адреса источника, а также для того, чтобы сразу понимать, куда будет отправлен ответ.

# Вывод

Итак, в процессе данной лабораторной работы я ознакомился с работой бесплатно распространяемой утилитой Wireshark, изучил структуру протокольных блоков данных, анализируя реальный трафик.