МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

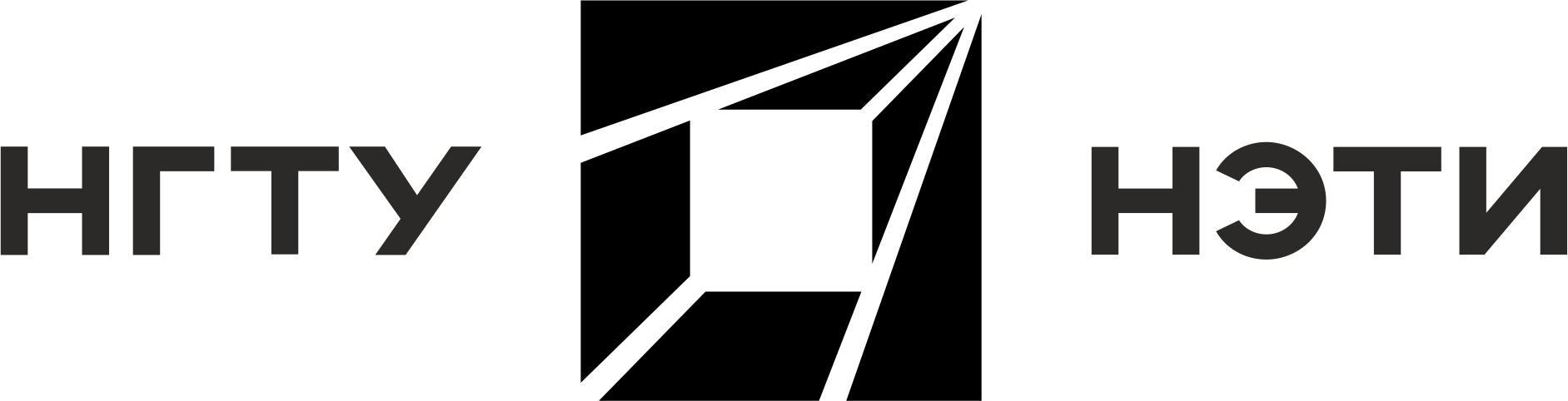
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ

«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра систем сбора и обработки данных



**ЛАБОРАТОРНАЯ РАБОТА №1**

по дисциплине: Сетевые информационные технологии

на тему: Моделирование компьютерных сетей

Вариант №3

Факультет: ФПМИ

Группа: ПММ-21

Выполнили: Сухих А.С., Черненко Д.А.

Проверил: к.т.н., доцент Кобылянский В.Г.

Дата выполнения: 25.10.22

Отметка о защите:

Новосибирск 2022

**Цель работы:** Изучение структуры передаваемых по сети кадров и пакетов, работающих на канальном и сетевом уровне.

**Ход работы:**

**1. Запустить перехват пакетов в WireShark.**

Через «Захват» и «Опции» были захвачены различные пакеты данных: DNS и UDP, а также пакеты TCP с ошибками получения:

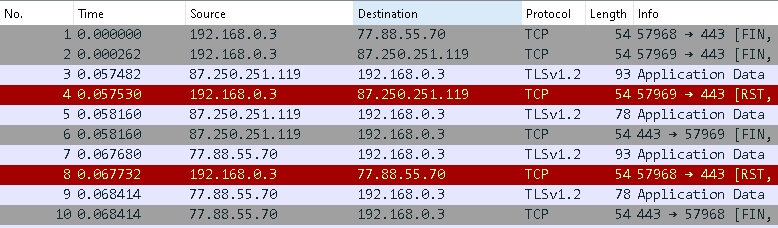


Рисунок 1.1 — захват пакетов данных

**2. Определить с помощью утилиты ping доступность заданных узлов sklad-service.ru, eye.moof.ru, gmail.com, wiw.ru, luminator.ru, hotlog.ru.**

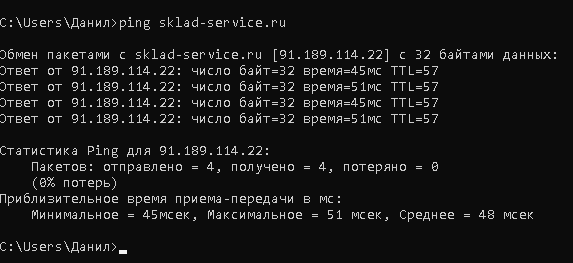


Рисунок 2.1 — обмен с sklad-service.ru данными

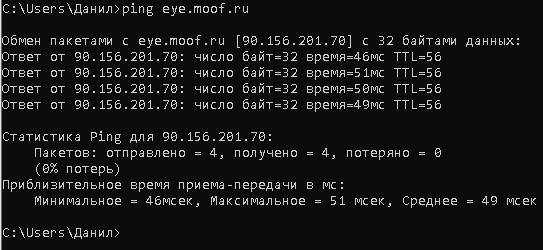


Рисунок 2.2 — обмен с eye.moof.ru данными

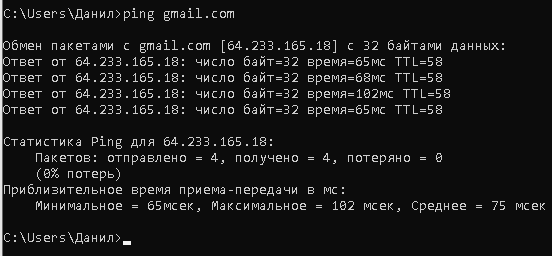


Рисунок 2.3 — обмен с gmail.com данными

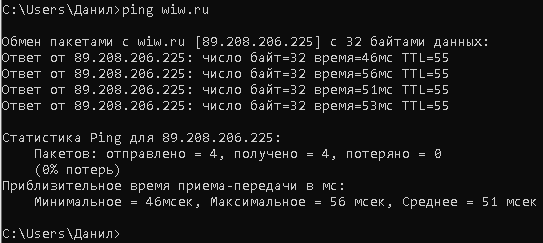


Рисунок 2.4 — обмен с wiw.ru данными

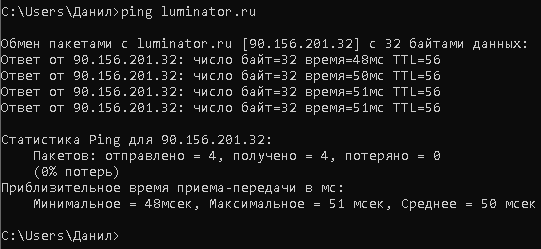


Рисунок 2.5 — обмен с luminator.ru данными

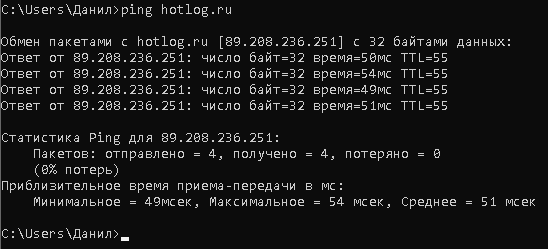


Рисунок 2.6 — обмен с hotlog.ru данными

Трассировка узла:

[andrew@manpc ~]$ tracepath gmail.com

1?: [LOCALHOST] pmtu 1500  
1: \_gateway 0.345ms  
1: \_gateway 0.308ms  
2: l37-192-51-254.novotelecom.ru 3.069ms  
3: 10.245.138.241 2.246ms  
4: 10.245.138.242 1.915ms asymm 5  
5: l49-128-50.novotelecom.ru 2.529ms asymm 6  
6: bbr03.spb.ertelecom.ru 43.639ms asymm 12  
7: net131.234.188-159.ertelecom.ru 43.714ms asymm 11  
8: no reply

После седьмого хопа трассировка не смогла получить ответ. Наиболее вероятно это связано с тем, что магистральные провайдеры блокируют ICMP-ответы по запросам с целью избежания DoS-атак.

**4. С помощью клиента WinSCP подключиться по протоколу FTP к серверу fpm2.ami.nstu.ru и выполнить копирование в Ваш домашний каталог текстового файла test2.txt.**

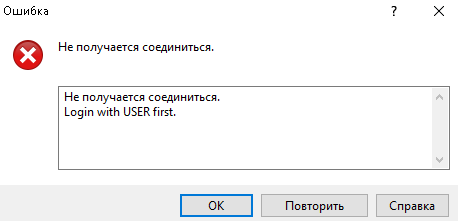


Рисунок 4.1 — невозможно подключиться к серверу

**5. Остановить перехват пакетов и сохранить результаты в файл с расширением .pcapng.**

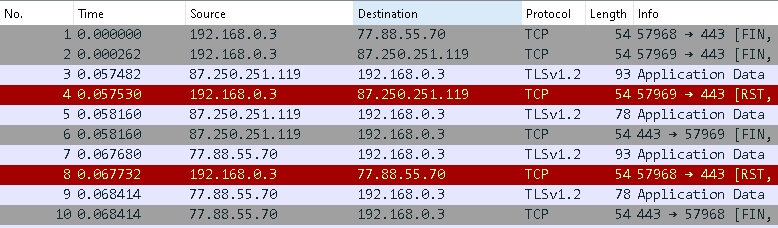


Рисунок 5.1 — результаты работы программы

Данный набор пакетов я сохранил в файл с раширением .pcang.

**6. С помощью WireShark определить внутреннюю структуру кадров и пакетов, передаваемых по сети; сравнить ее со структурами, описанными в протоколах Ethernet, IP и TCР.**

**TCP:**

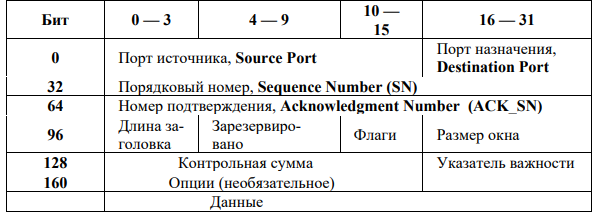
****

Рисунок 6.1 — структура TCP сегмента

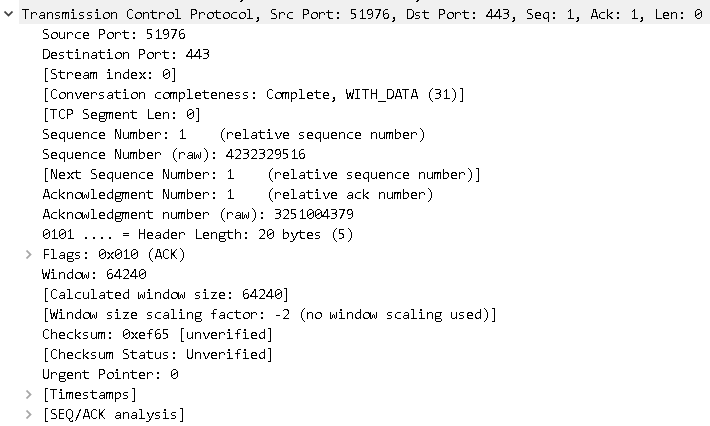


Рисунок 6.2 — структура TCP сегмента в программе Wireshark

**IP:**

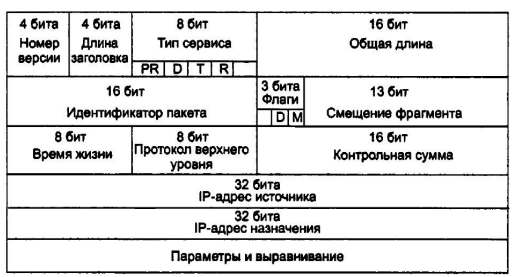


Рисунок 6.3 — структура IP пакета

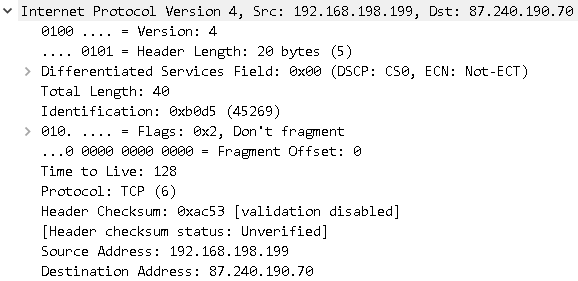


Рисунок 6.4 — структура IP пакета в программе Wireshark

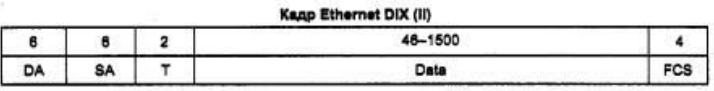


Рисунок 6.5 — структура Ethernet кадра

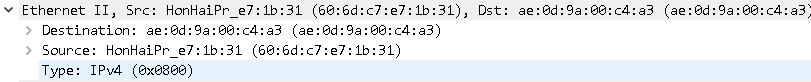


Рисунок 6.6 — структура Ethernet кадра в программе Wireshark

**7. Определить последовательность прохождения запросов, реализующих алгоритм трассировки одного из заданных узлов.**

[andrew@manpc AMI\_1st\_sem]$ tracepath sklad-service.ru

1?: [LOCALHOST] pmtu 1500

1: \_gateway 0.339ms

1: \_gateway 0.298ms

2: l37-192-51-254.novotelecom.ru 1.727ms

3: 10.245.138.241 1.525ms

4: 10.245.138.242 2.094ms asymm 5

5: 10.245.141.14 2.144ms asymm 4

6: no reply

7: ctv-r1.nic.ru 55.098ms asymm 15

8: wcarp.hosting.nic.ru 55.477ms reached

Resume: pmtu 1500 hops 8 back 8

Трассировка была выполнена с помощью команды tracepath. Она использует UDP-порты для отправки пакетов, чтобы отслеживать путь к указанному пункту назначения.

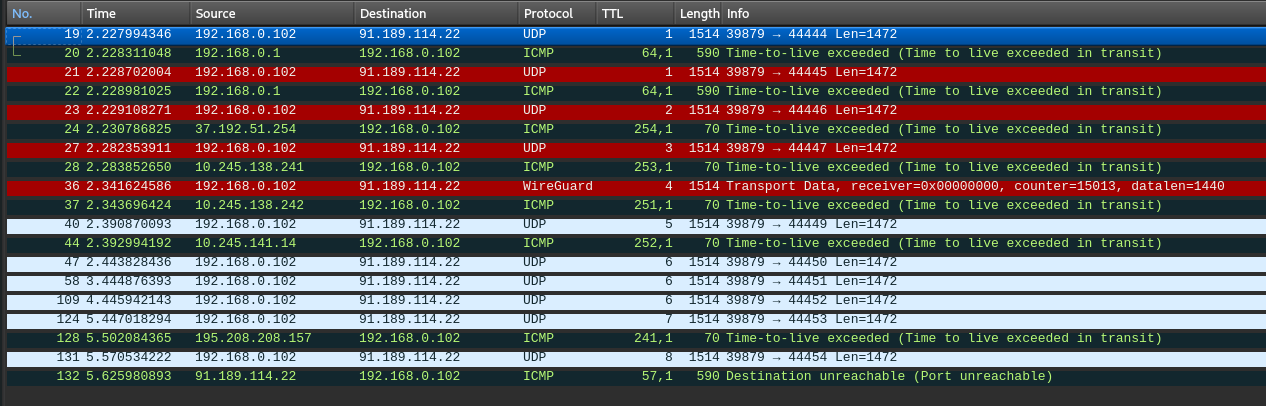


Рисунок 7.1 - трафик ICMP-пакетов в результате работы команды tracepath

В процессе своей работы программа отправляет дейтаграммы на указанный адрес с параметром Time-to-live (время жизни), увеличивающимся на 1 с каждым хопом. Таким образом, каждый маршрутизатор в сети получает пакет с истекшим временем жизни и отправляет ICMP-ответ с ошибкой Time-to-live exceeded.

В процессе работы программа отправляет до 3 пакетов. На 6 хопе программа не получила ответ, поэтому видим все 3 отправленных пакета.

**8. Восстановить сеанс обмена данными по протоколу HTTP между браузером и сервером при выполнении п.3.**

Сеанс обмена был осуществлён исключительно с сайтом nstu.ru, так как на момент изучения данного пункта лабораторной работы сайт moodle был недоступен.

Первым делом, перехватим трафик программой WireShark:

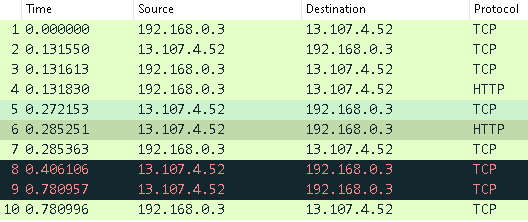


Рисунок 8.1 — перехват данных с сайта nstu.ru

Рассмотрим более детально пакеты протокола HTTP:

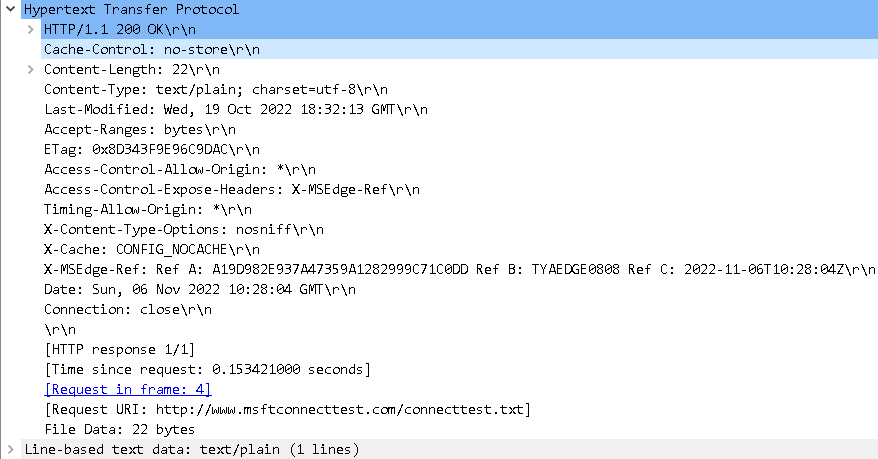


Рисунок 8.2 — результат обращения к сайту nstu.ru по протоколу http

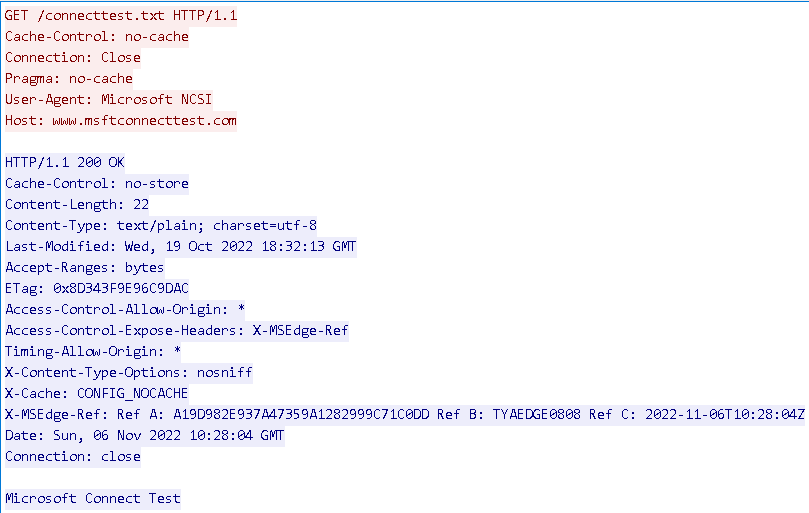


Рисунок 8.3 — результат обращения к сайту nstu.ru по протоколу http

Как мы можем увидеть, у нас вышло получить доступ к сайту, а также мы смогли зафиксировать тип и размер передаваемых данных.

**9. Восстановить сеанс обмена данными по протоколу FTP при выполнении п.4, найти перехваченные логин и пароль, а также восстановить содержимое переданного файла.**

Так как п.4 выполнить не удалось из-за отсутствия доступа к серверу fpm2.ami.nstu.ru., данное задание будет выполнено с подключением к общедоступному серверу ftp.dlptest.com, позволяющему загружать файлы, которые будут храниться на сайте не более 10 минут.

Восстановленный сеанс обмена данными:

220 Welcome to the DLP Test FTP Server

USER dlpuser

331 Please specify the password.

PASS rNrKYTX9g7z3RgJRmxWuGHbeu

230 Login successful.

SYST

215 UNIX Type: L8

TYPE I

200 Switching to Binary mode.

PORT 192,168,0,102,169,111

200 PORT command successful. Consider using PASV.

RETR ftp\_uploat\_test.txt

150 Opening BINARY mode data connection for ftp\_uploat\_test.txt (9 bytes).

226 Transfer complete.

QUIT

221 Goodbye.

С сервера был скачан файл ftp\_uploat\_test.txt с содержимым “Some text”, который был ранее на сервер. Восстановить содержимое переданного файла можно по пакету протокола “FTP-DATA”:

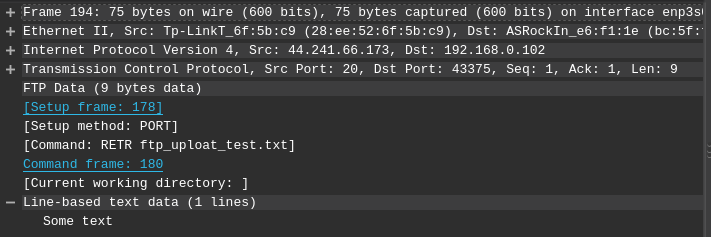


Рисунок 9.1 - восстановление с помощью Wireshark содержимого файла, переданного по протоколу FTP

**10. Определить последовательность прохождения запросов, реализующих протокол ARP. Построить схему работы протокола и формат пакетов.**

Схема работы ARP-протокола следующая:

1. Отправляем ARP-запрос с нашего ПК по нужному нам IP на широковещательный MAC-адрес: ff:ff:ff:ff:ff:ff.
2. Если в сети есть устройство с нужным нам IP, оно отправляет ARP-ответ, в котором находится MAC-адрес компьютера.
3. Отправитель ARP-запроса получает ответ и извлекает из него MAC-адрес, после чего записывает полученные IP и MAC адреса в таблицу соответствия.

Формат ARP-запроса можно увидеть на рисунке 10.1, а также он описан ниже:

1. Тип сети
2. Тип протокола сетевого уровня
3. Длина локального адреса
4. Длина глобального адреса
5. Тип операции (1 - запрос, 2 - ответ)
6. Локальный адрес отправителя
7. Глобальный адрес отправителя
8. Локальный адрес получателя
9. Глобальный адрес получателя

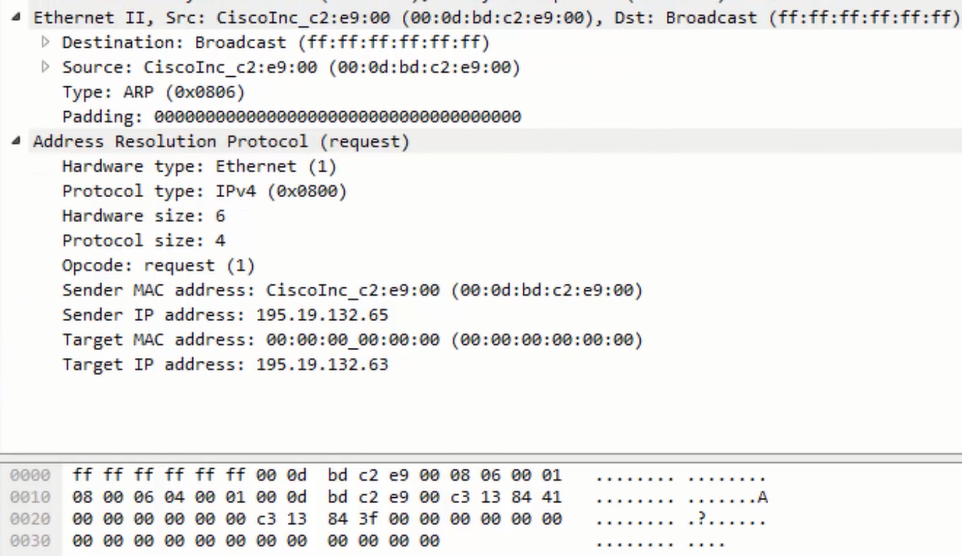


Рисунок 10.1 — ARP-запрос

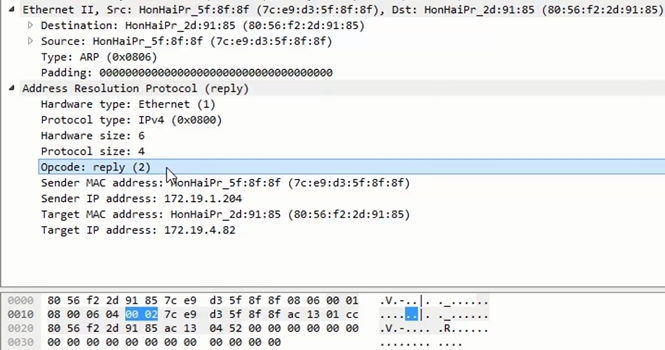


Рисунок 10.2 — ARP-ответ

**11. Найти в перехваченном трафике пакеты, передаваемые по протоколу STP, определить назначение данного протокола.**

STP (Spanning Tree Protocol) - протокол связующего дерева. Работает на канальном уровне модели OSI и обеспечивает отсутствие петлей между коммутаторами (или маршрутизаторами), связанных избыточными физическими соединениями. Примером таких соединений является резервирование канала, когда коммутаторы объединяются двумя или более каналами связи, либо при соединении коммутаторов в кольцо. При отправке broadcast-сообщения сетевым устройством возникает широковещательный шторм.

STP позволяет устанавливать такие избыточные соединения в заблокированное состояние, на которые при падении основного соединения переключается сетевое устройство.

При фильтрации пакетов на интерфейсе сетевой карты с помощью Wireshark не было обнаружено пакетов с протоколом STP, так как в домашней локальной сети нет резервных соединений и иных петлей.

**12. Найти в перехваченном трафике широковещательные запросы по протоколам DHCP, ARP и ответы на них. Определить структуру передаваемых по этим протоколам кадров.**

Начнём с протокола DHCP, так как ARP мы уже рассматривали выше. Для начала зайдём в командную строку и возьмём себе новый IP адрес при помощи команды inconfig /release и inconfig /renew:

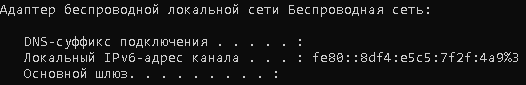


Рисунок 12.1 — освобождение IP-адреса

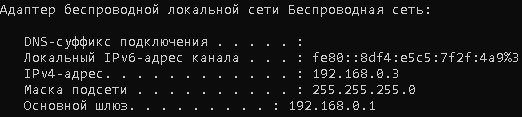


Рисунок 12.2 — получение IP-адреса

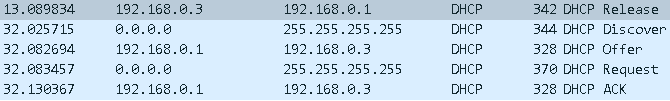


Рисунок 12.3 — DORA: получение от сервера IP-адреса клиентом

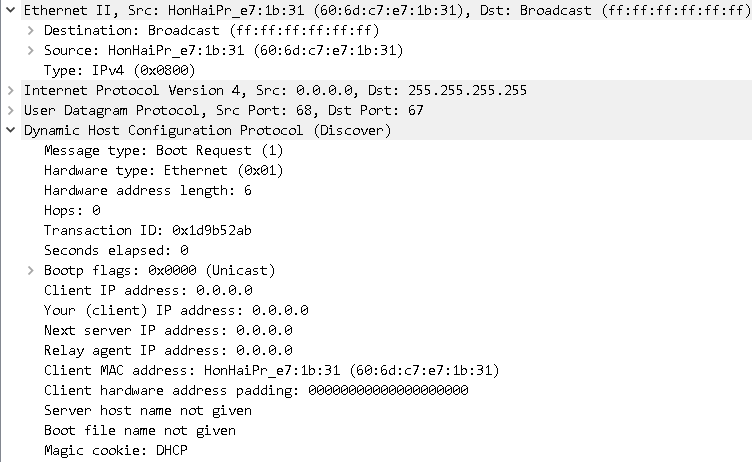


Рисунок 12.4 — DISCOVER: отправка серверу запроса от клиента с целью получения IP

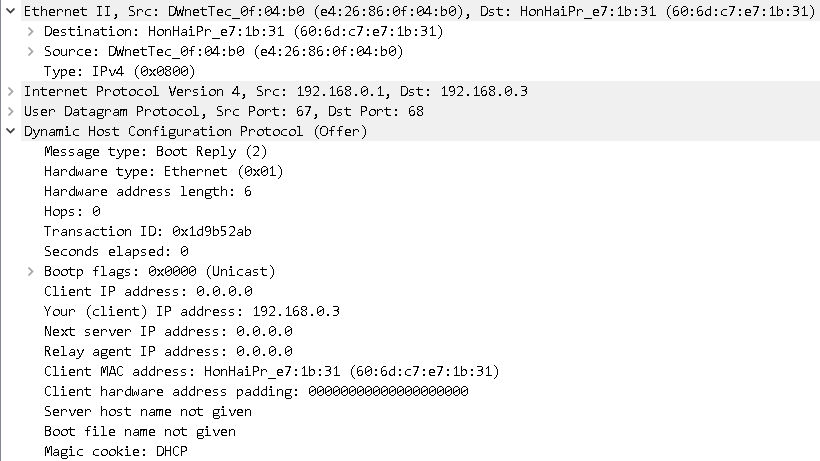


Рисунок 12.5 — OFFER: отправка пакета с назначенным IP-адресом клиенту

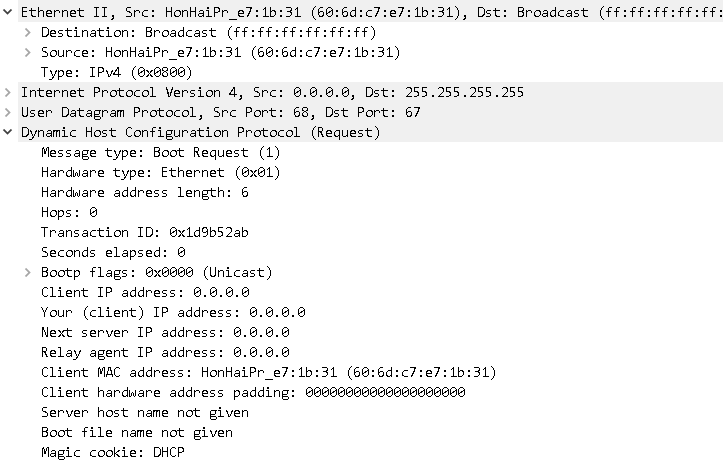


Рисунок 12.6 — REQUEST: отправка пакета клиентом серверу с IP клиента для подтверждения присвоения ему этого адреса

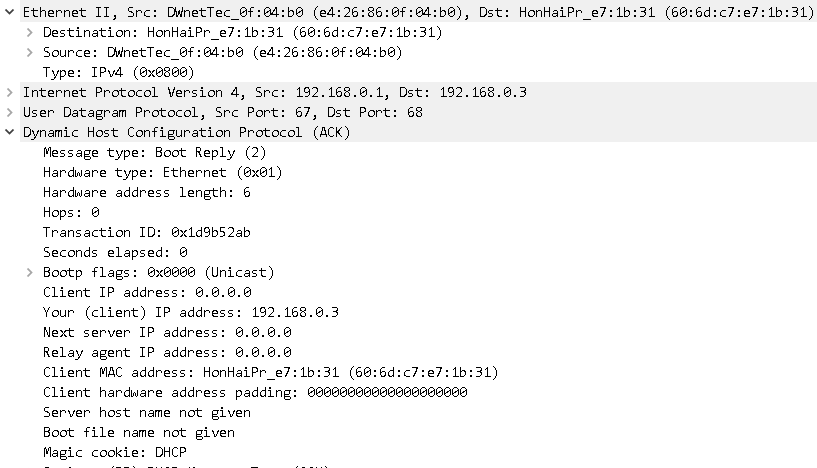


Рисунок 12.6 — ACK: получение пакета от сервера с информацией о присвоении клиенту IP-адреса

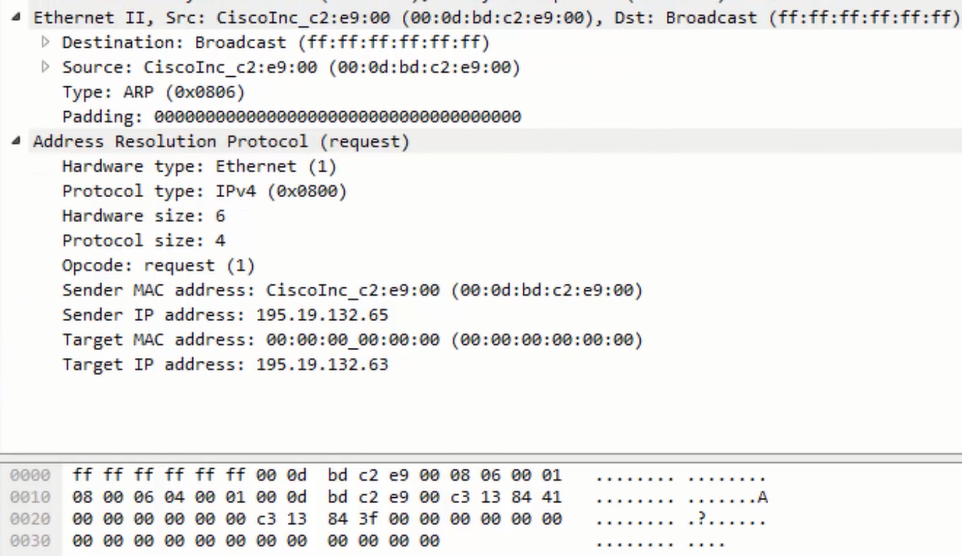


Рисунок 12.7 — ARP-запрос

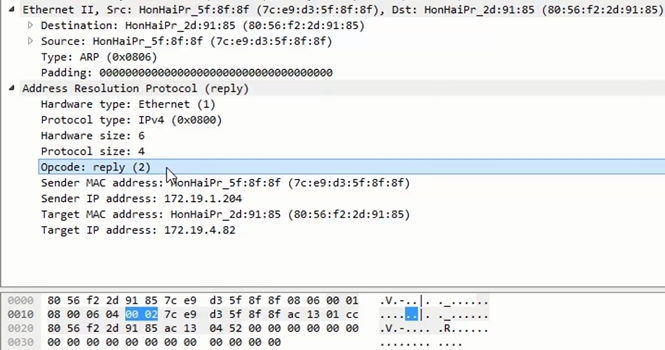


Рисунок 12.8 — ARP-ответ

**13. Определить значение поля «Тип данных» для кадра Ethernet при передаче пакетов IP, ARP, ICMP, DNS, DHCP.**

У всех протоколов кроме ARP тип данных в кадре Ethernet указан как IPv4(6), так как это поле указывает какой протокол используется в протоколе третьего уровня. Протокол ARP можно отнести к протоколам третьего уровня, так как он взаимодействует с IP-адресами. Остальные протоколы используют протокол IP для маршрутизации, поэтому он указан в типе данных.

| Протокол | “Тип данных” в кадре Ethernet |
| --- | --- |
| IP | IPv4 (0x0800) |
| ARP | ARP (0x0806) |
| ICMP | IPv4 (0x0800) |
| DNS | IPv4 (0x0800) |
| DHCP | IPv4 (0x0800) |

**14. Построить статистику по используемым за время сеанса протоколам.**

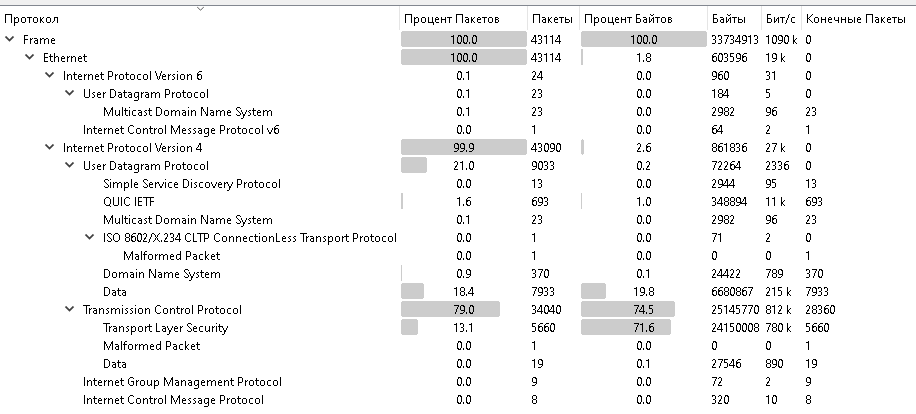


Рисунок 14.1 — статистика по протоколам

**15. Изучить процесс установления соединения по протоколу TCP.**

TCP (Transfer Control Protocol) - протокол контроля передачи данных. Протокол транспортного уровня, обеспечивающий гарантию доставки данных, а также гарантию сохранения порядка следования байтов. При установке соединения использует алгоритм “тройного рукопожатия”. Перед непосредственной передачей данных одной из сторон (как правило клиентом) инициируется процесс обмена отправкой сегмента SYN с некоторым номером передаваемого байта Sequence Number (SeqN), в ответ другая сторона (как правило сервер) отправляет сегмент готовности к установке соединения SYN-ACK с номером передаваемого байта SeqN из своей последовательности, а также следующим номером байта Acknowledgement Number (AckN) ожидаемого от инициатора. Инициатор отправляет сегмент подтверждения ACK с номером передаваемого байта SeqN, соответствующим AckN из полученного ответа и ожидаемым номером следующего байта AckN из последовательности байтов сервера. Соединение считается установленным.

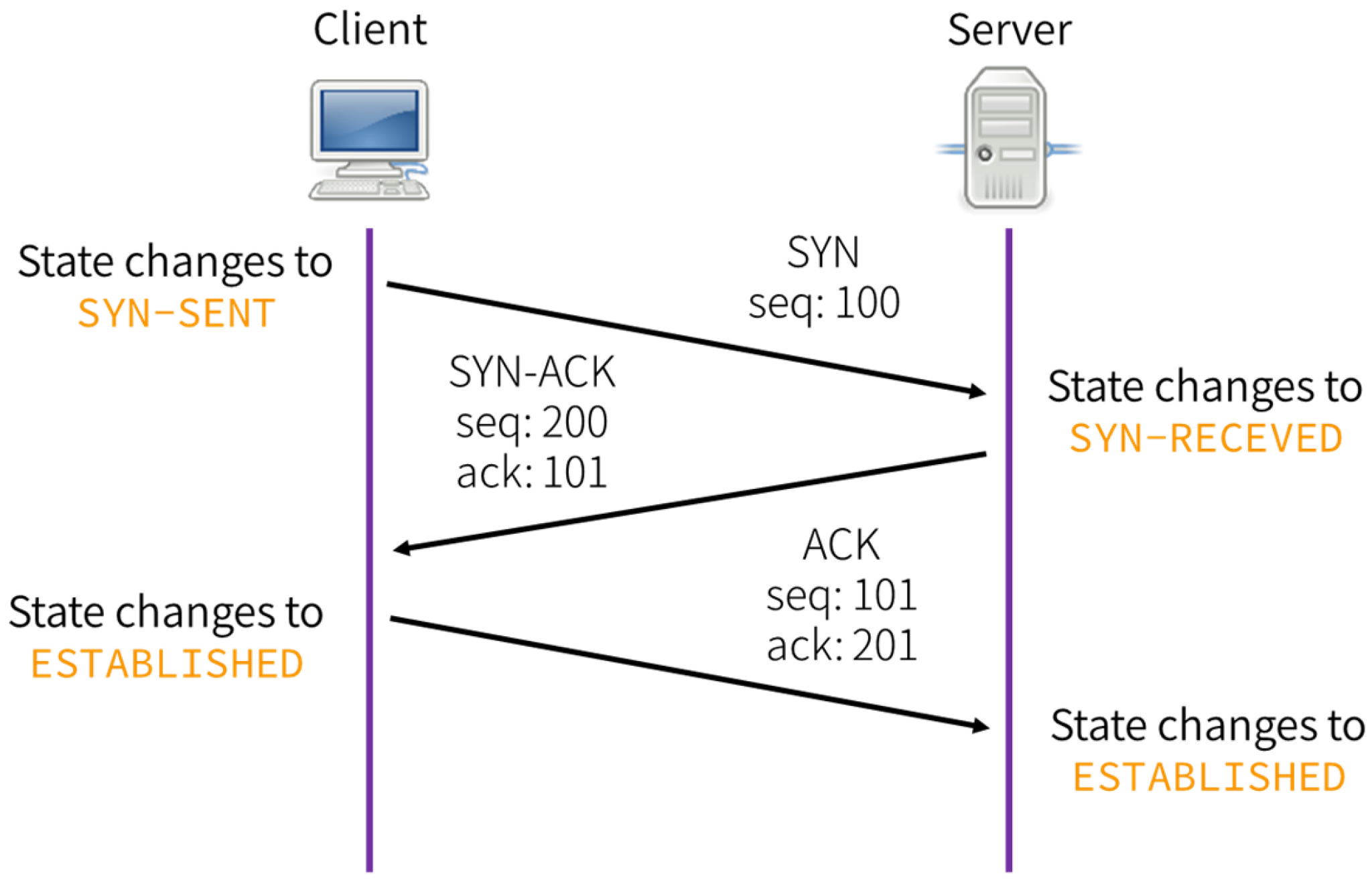


Рисунок 15.1 — схема “тройного рукопожатия”

Пример работы этого протокола рассмотрим с помощью Wireshark:

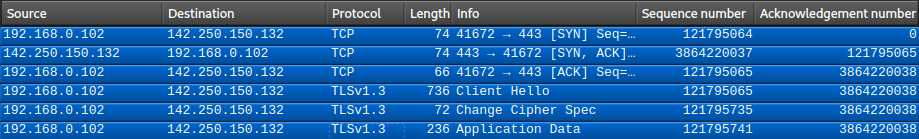


Рисунок 15.2 — анализ трафика установки TCP-соединения

С адреса 192.168.0.102 инициализируется соединение с сервером 142.250.150.132 отправкой сегмента с флагом SYN и SeqN = 121795064. Сервер принимает соединение, отправляя сегмент с флагами SYN, ACK и значениями SeqN = 3864220037 и AckN = 121795065, что на 1 больше, чем полученный ранее от инициатора SeqN.

Хост подтверждает соединение отправляя сегмент с флагом ACK и SeqN, равным полученному AckN от сервера, а также AckN больше полученного SeqN от сервера на 1.

При дальнейшей передаче данных от хоста в различных сегментах AckN остается прежним, а номера передаваемых байтов SeqN изменяются.

**Вывод:** В ходе лабораторной работы нашей бригадой были детально изучены протоколы UDP, TCP, ARP, HTTP, DNS, DHCP, и т.д.. Благодаря этому удалось выявить различия между этими протоколами.