Практическая работа

Дисциплина: компьютерные сети

Автор: Игошев Ростислав Вадимович

Группа: П50-4-21

Тема: Использование Wireshark для просмотра трафика

Цель работы: научиться использовать Wireshark для просмотра трафика, рассмотреть его функции и возможности, выполнить практическое задание.

Wireshark - программа-анализатор трафика для компьютерных сетей Ethernet и некоторых других. Программа позволяет пользователю просматривать весь проходящий по сети трафик в режиме реального времени, переводя сетевую карту в неразборчивый режим (сетевая плата позволяет принимать все пакеты независимо от того, кому они адресованы).

Начнём с стартового меню приложения, появившееся после запуска.

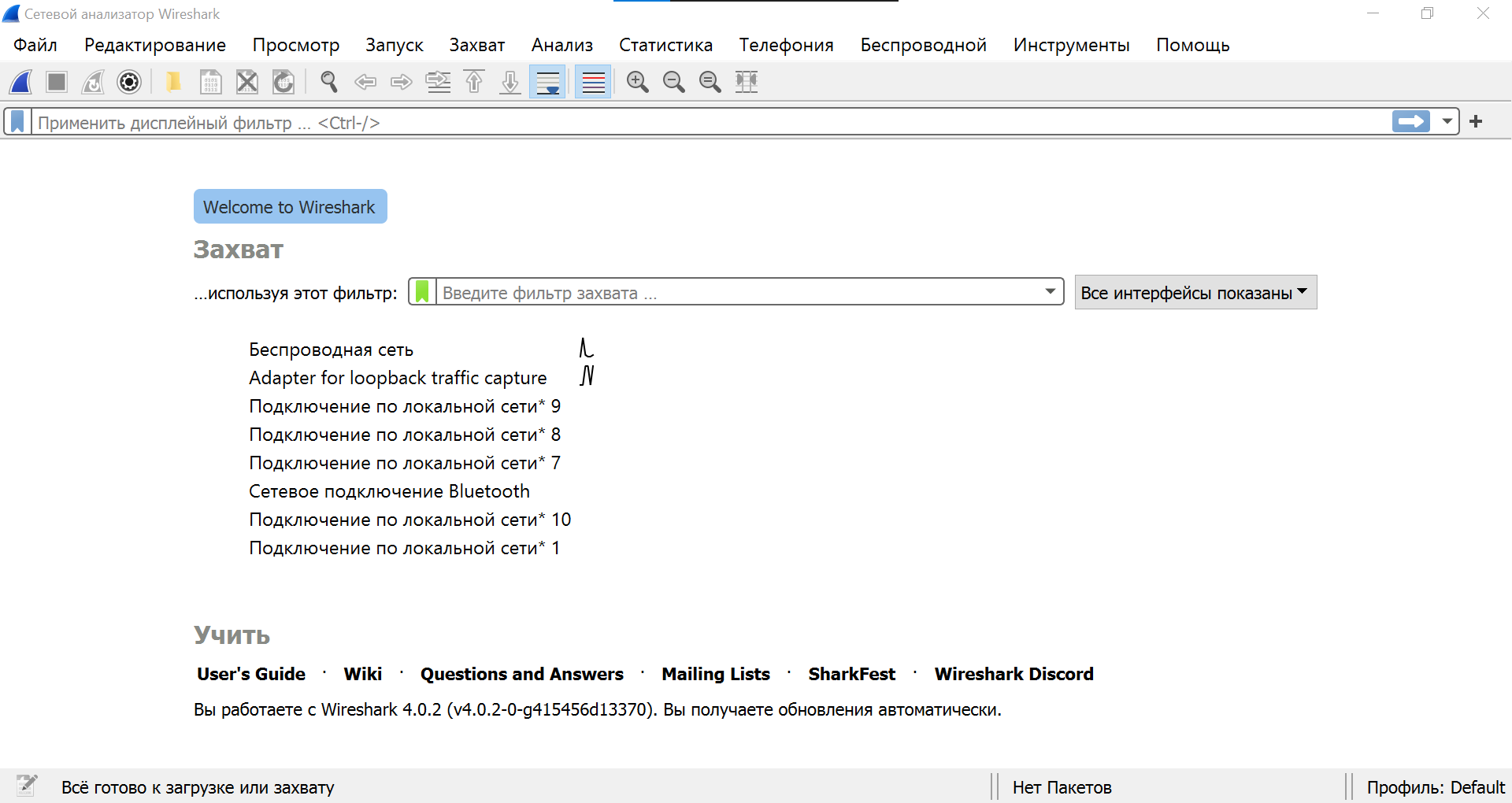


Рисунок 1 – Стартовое меню

В этом меню есть стандартная для профильных приложений верхняя шторка с полезными инструментами, некоторые кнопки, а также колонка для ввода фильтра захвата, но, для начала, следует выбрать, какой тип сети просматривать для фиксирования трафика.

После выбора сети, начнётся перехват пакетов, и перед пользователем появится список пакетов, изменяющийся в реальном времени, которых удалось перехватить.

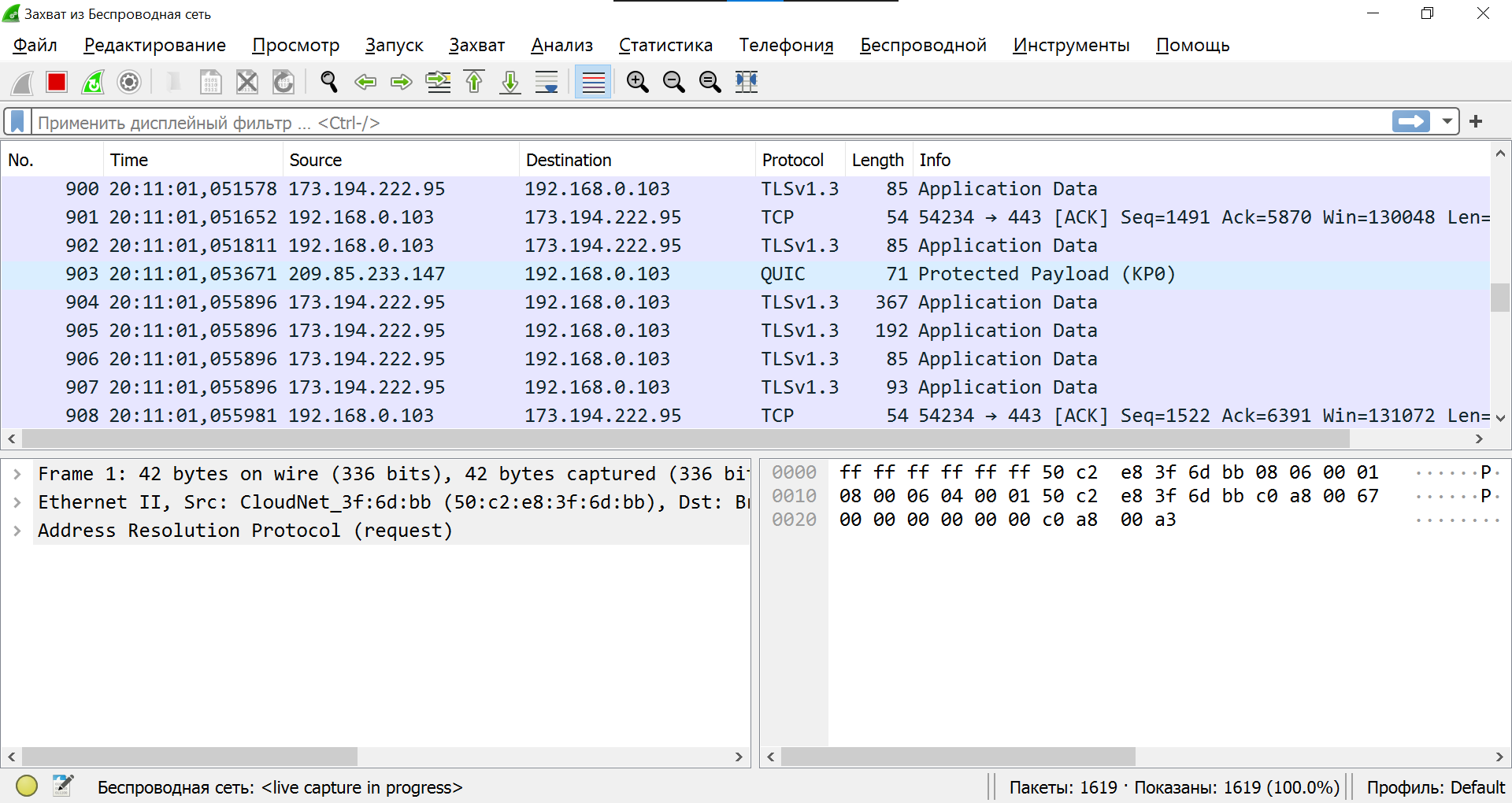


Рисунок 2 – Меню перехвата пакетов

Кнопки, находящиеся ниже наименования программы, позволяют начать, остановить и перезапустить перехват пакетов соответственно.

Выделив пакет, в нижней половине программы появится поля и байты выделенного пакета.

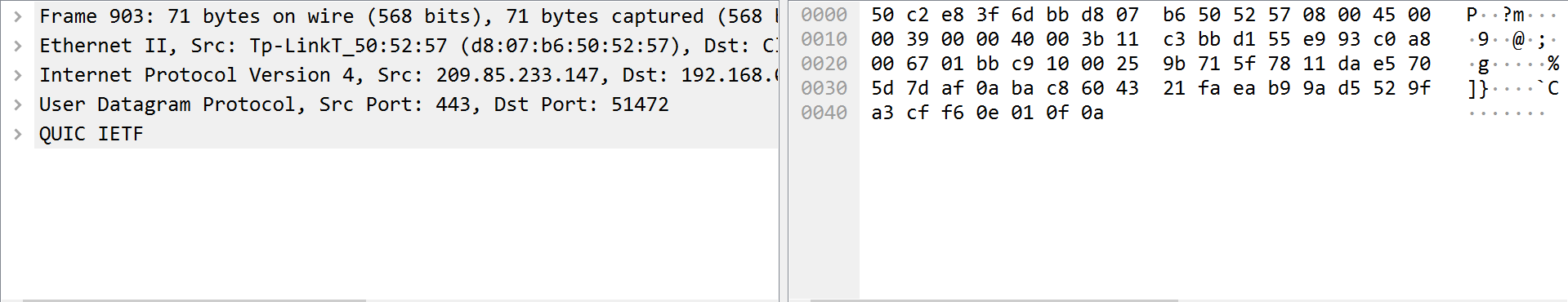


Рисунок 3 – Данные о пакете

Нажав на один из диапазонов данных, в соседнем окне выделится соответствующий первому диапазон.

Итак, в первой области приложения показывается следующая информация, состоящая из семи колонок:

1. Номер пакета;
2. Время (по умолчанию в формате количества секунд, прошедшего после времени захвата первого пакета);
3. IP-адрес источника;
4. IP-адрес получателя;
5. Протокол пакета;
6. Длина пакета в байтах;
7. Краткая информация о содержимом пакета (порты приема-передачи, порядковый номер пакета, номер подтверждения и др.).

При нажатии на колонку отображаемые пакеты сортируются по ней.

Некоторым пользователям интерфейс может показаться неудобным. Для этого в программе существует настройка отображения «Просмотр».

Здесь, к примеру, можно изменить настройку отображения времени на удобное «Время Дня».

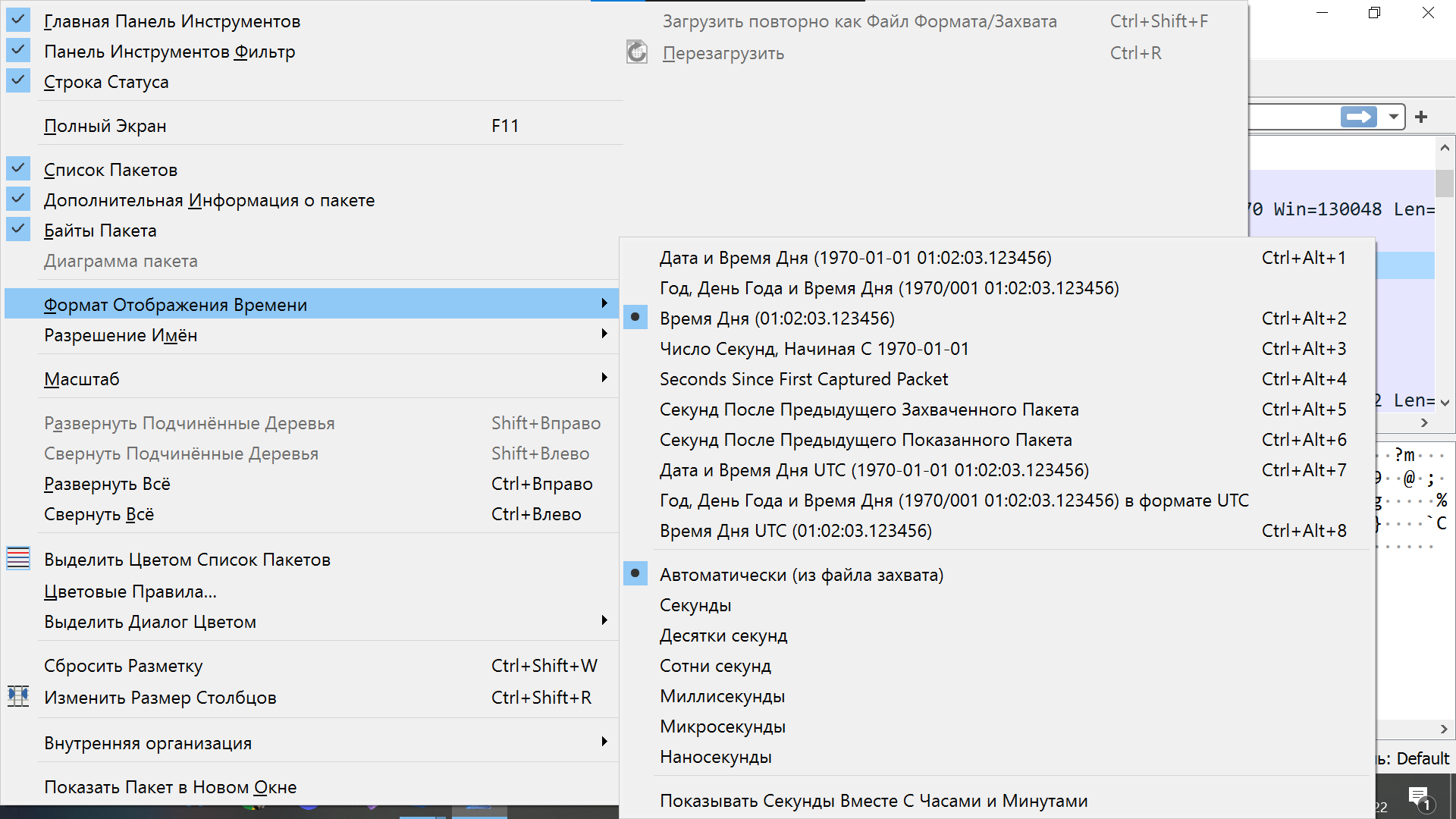


Рисунок 4 – Настройки «Просмотр»

Понаблюдав за перехватываемыми пакетами, можно заметить, что пакеты могут иметь разный цвет.

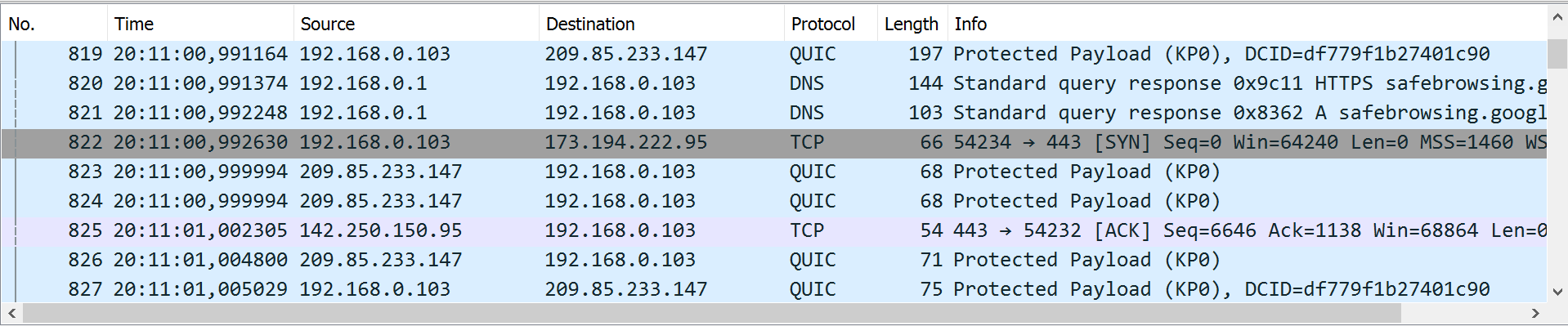


Рисунок 5 – Пакеты

Цвет пакета зависит от его протокола и состояния. Некоторые из них:

* Фиолетовый – TCP (Надежность)
* Зеленый – HTTP (Веб)
* Оранжевый - Маршрутизация
* Голубой – UDP (Простота)
* Розовый – ICMP (Ping)
* Серый – Начало и конец TCP передачи
* Красный – Сброс или прерывание
* Желтый - ARP (MAC по IP)
* Черный – Ошибки

Цвета тоже можно настроить во вкладке «Просмотр» нажав на кнопку «Цветовые правила».

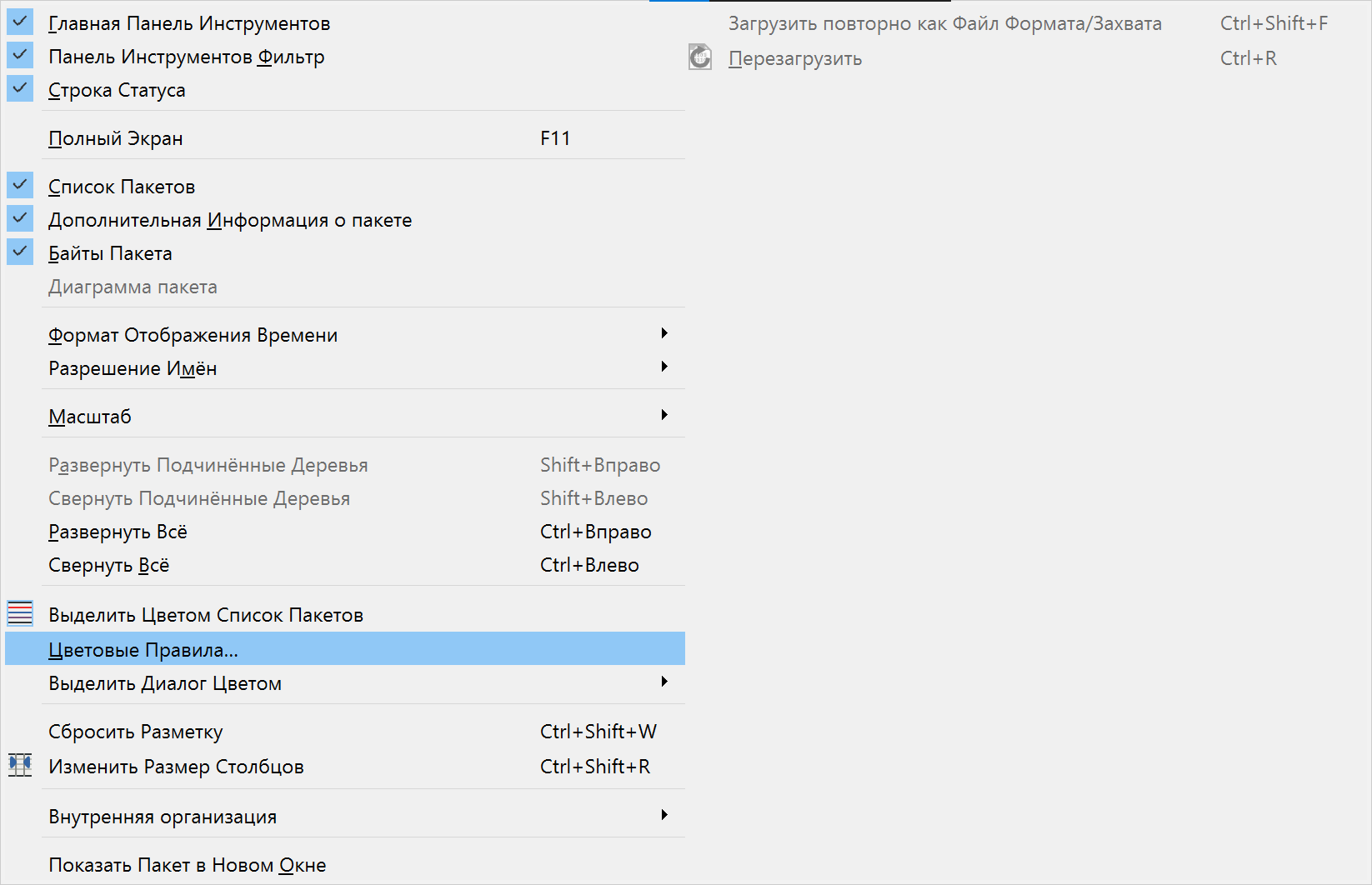


Рисунок 6 – Цветовые Правила

В Wireshark есть функция фильтрации, позволяющая находить определенные необходимые пакеты. Для этого существует специальная строка, куда можно вводить фильтр.



Рисунок 7 – Строка ввода фильтров

Если фильтрация прошла успешно, то цвет строки станет зелёным, иначе – красным. Все фильтры приведены на данной странице: https://wireshark.org/docs/dfref, и их огромное количество.



Рисунок 8 – Успешная фильтрация



Рисунок 9 – Ошибка фильтрации

Введя фильтр, программа покажет нам пакеты, подходящие под условие фильтра. Введем: «dns».

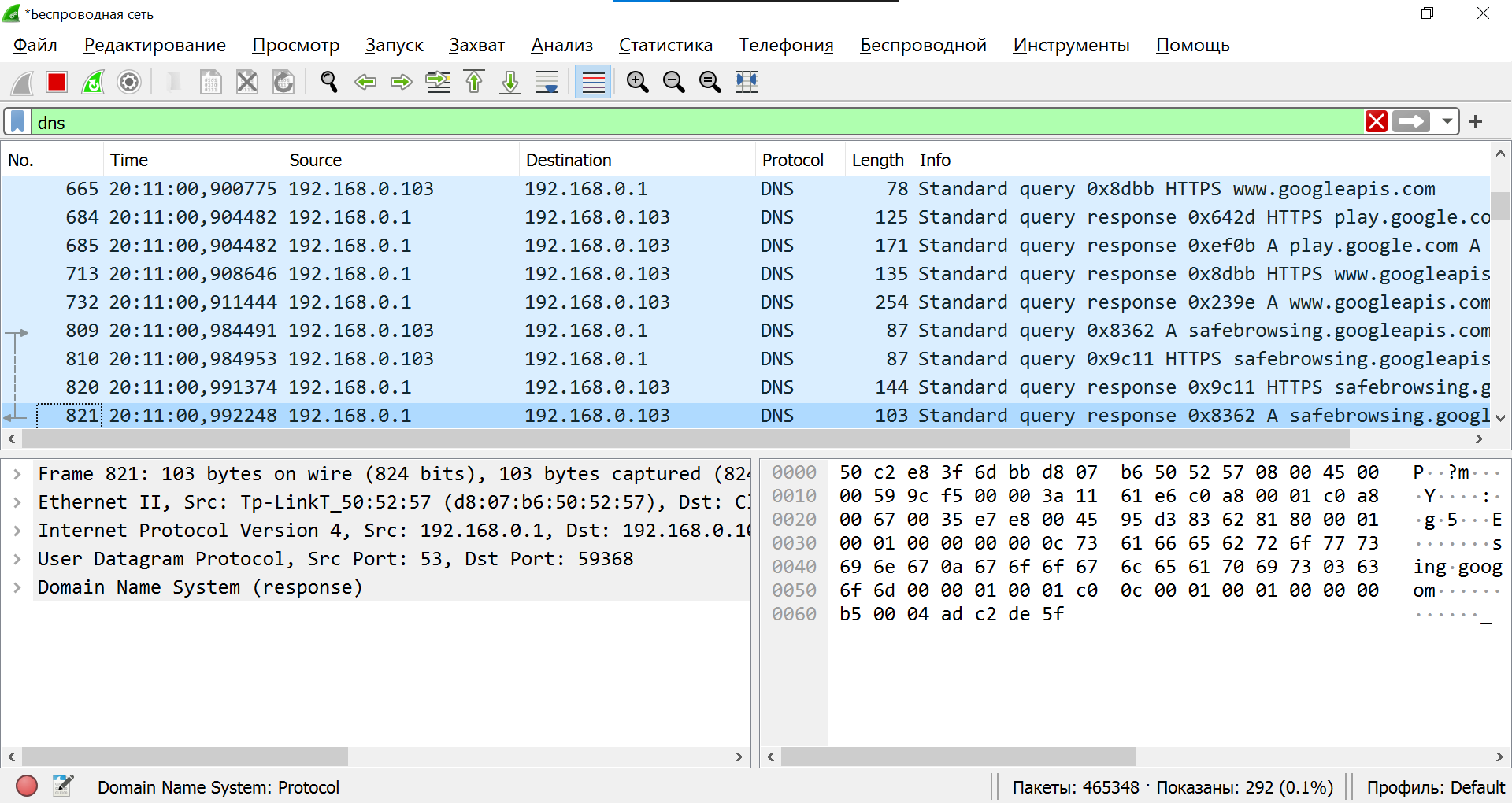


Рисунок 10 – Фильтрация по dns

Теперь, попробуем отфильтровать по выражению «arp», которые содержат физические и интернет-адреса.

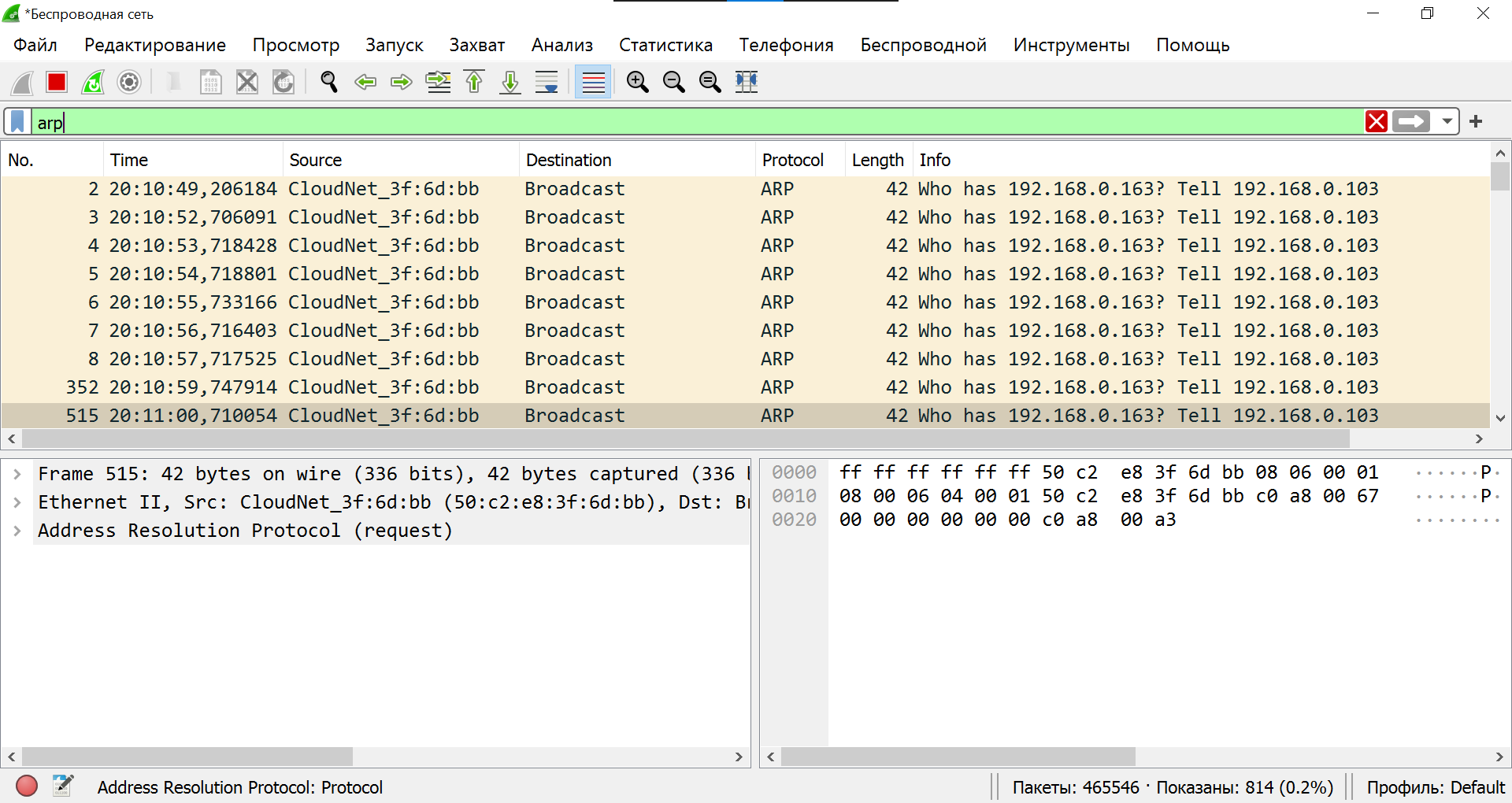


Рисунок 11 – Фильтрация по arp

Также, пакеты можно отфильтровать по порту. В качестве примера введем в строке фильтрации «tcp.port == 80».

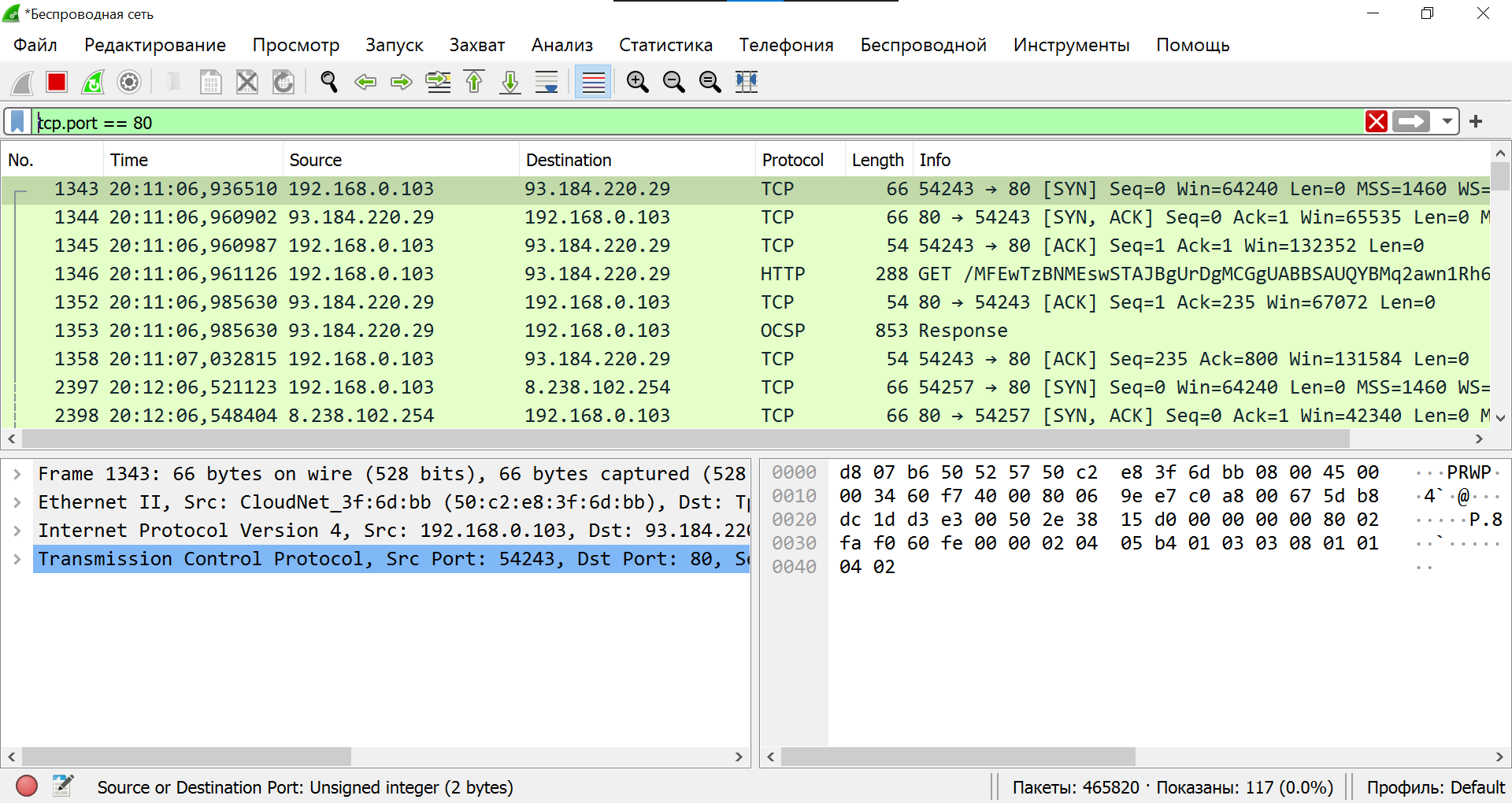


Рисунок 12 – Фильтрация по порту

Пакеты можно сортировать и по адресу. Для этого в строку нужно ввести ip.addr == (свой локальный адрес). Свой локальный адрес можно узнать, выполнив эти действия: Выберите Параметры > Сеть & Интернет > Wi-Fi, а затем выберите сеть Wi-Fi, к которой подключены. В разделе Свойства найдите IP-адрес, указанный рядом с пунктом IPv4-адрес.

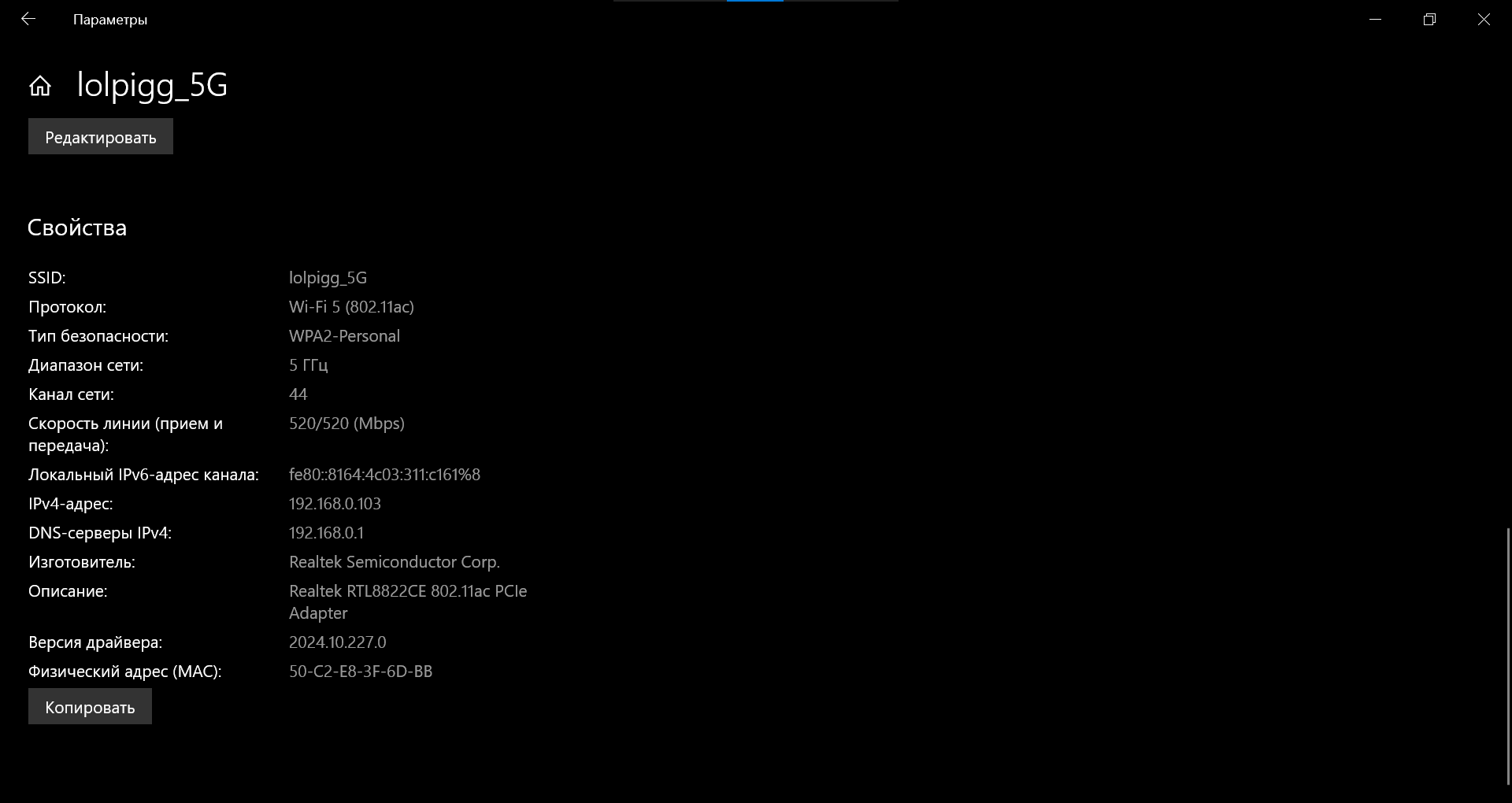


Рисунок 13 – Настройки сети

Кроме того, помимо этого способа, узнать свой локальный IP можно через командную строку, введя в ней команду ipconfig, где IPv4 – адрес и будет нужным.

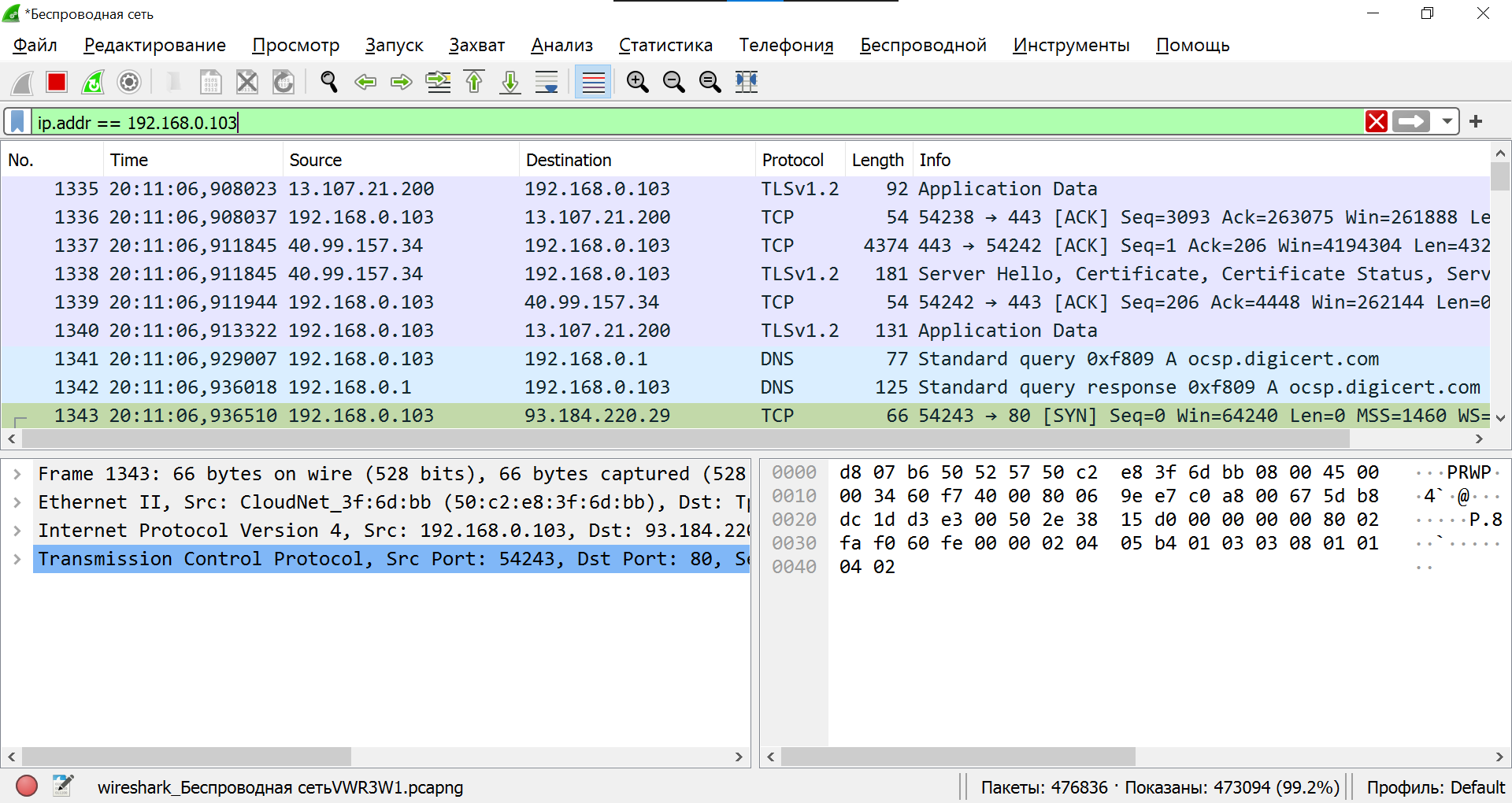


Рисунок 14 – Фильтрация по адресу

Если нужно отфильтровать пакеты по адресу только отправителя или только получателя, то надо использовать поля ip.src и ip.dst. Для портов – tcp.srcport и tcp.dstport для TCP и udp.srcport и udp.dstport для UDP.

Сбросить фильтр можно, нажав на крестик в строке. Кроме того, нужные фильтры можно сохранять в закладках, дабы использовать их в последующем.



Рисунок 15 – Закладка

Теперь, разберём байты пакета. В правой нижней области отображены данные выбранного пакета в сыром виде. Эта область состоит из трех колонок:

1. Смещение с начала пакета;

2. Байты в шестнадцатеричном виде;

3. Текстовое представление содержимого пакета.

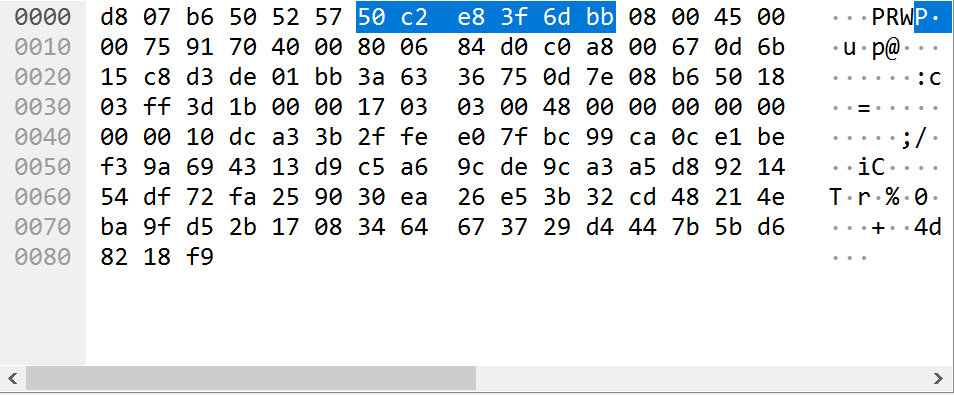


Рисунок 16 – Байты

Также, отображение байтов можно изменить, к примеру, на двоичный способ записи.

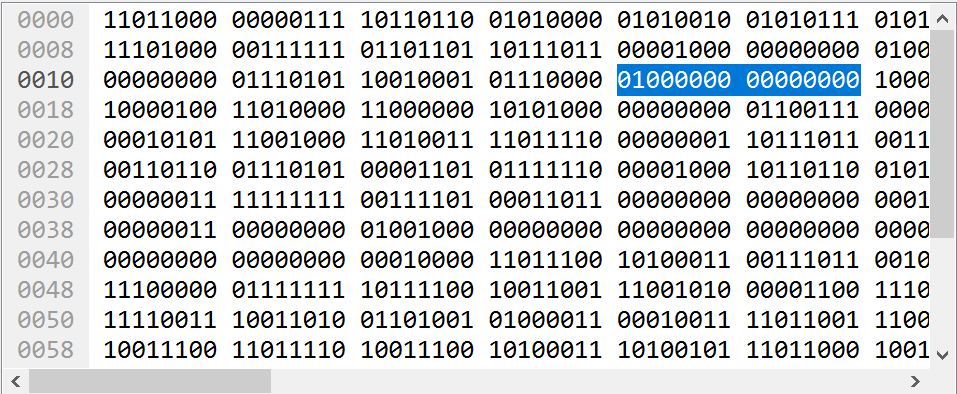


Рисунок 17 – Двоичный способ записи

При наведении на поле оно подсвечивается целиком, и слева снизу пишется байтовый диапазон и назначение выделенного поля.

Области можно настраивать. Изменим назначение левой нижней области. Для этого открываем меню «Редактирование», «Параметры». Переходим к странице «Появление», «Разметка».

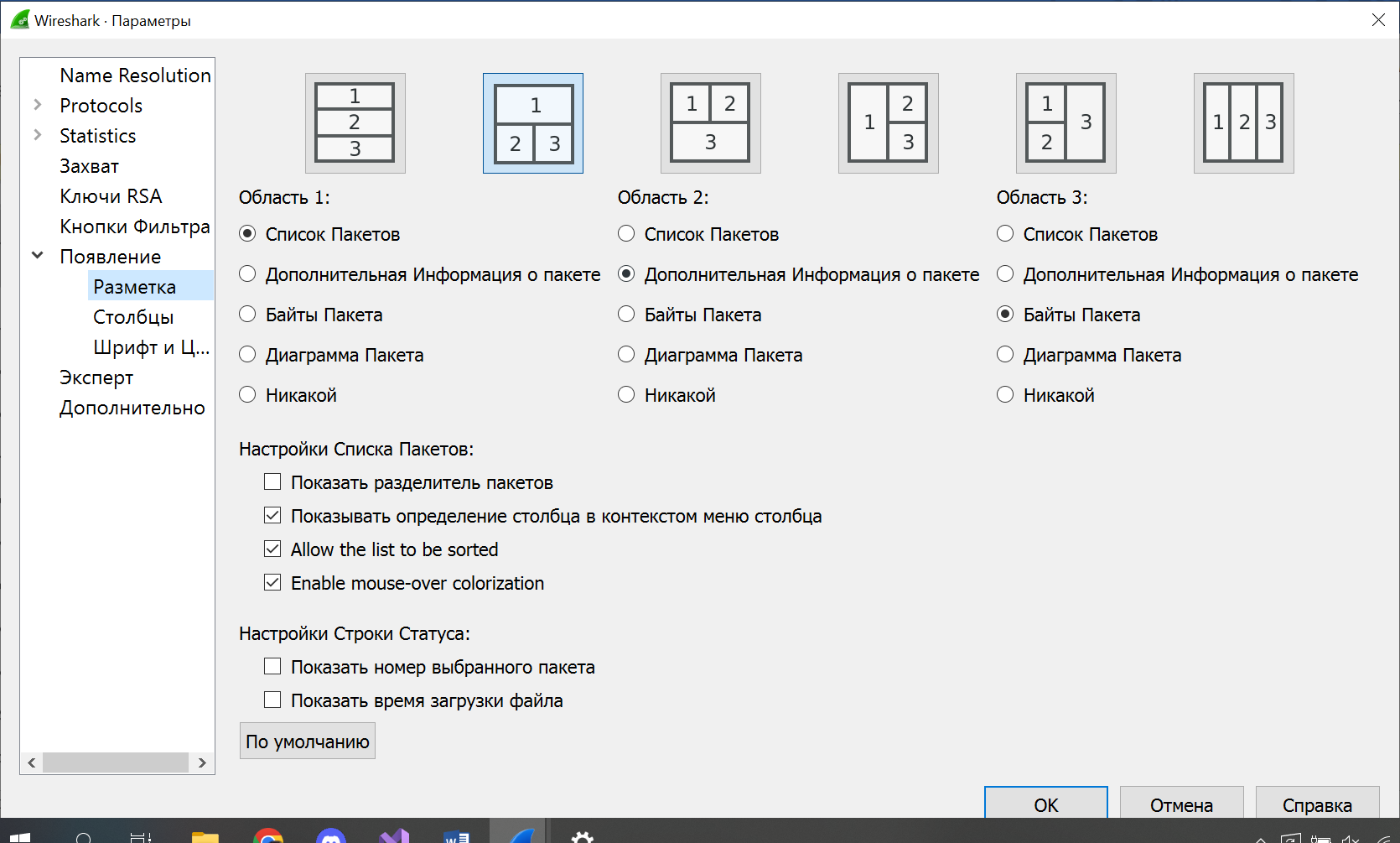


Рисунок 18 – Параметры Разметки

Для примера, изменим область под номером 2 на Диаграмму Пакета.

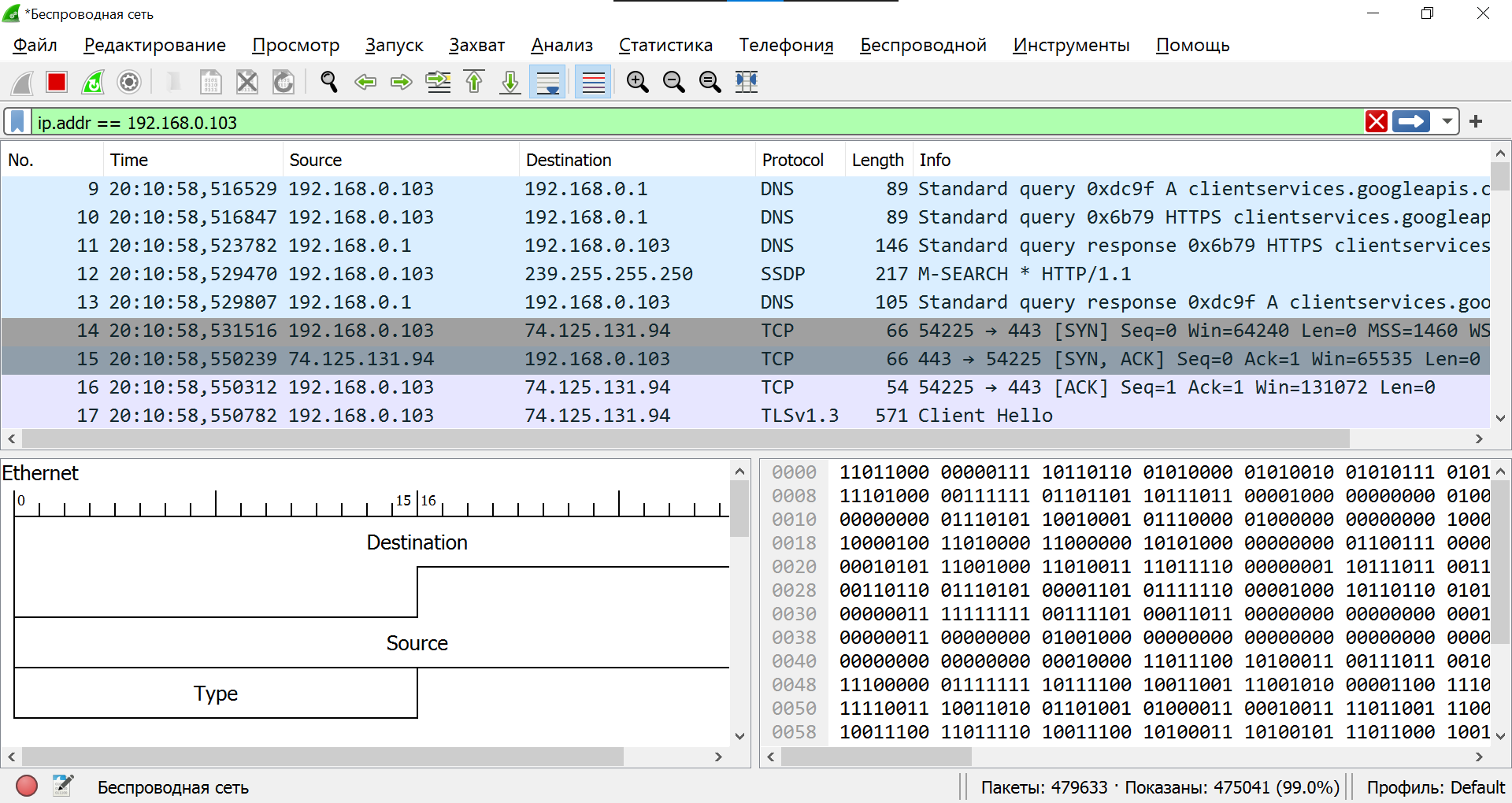


Рисунок 19 – Диаграмма пакета во 2 области

Как видите, во второй области началась изображаться диаграмма. Это графическое представление полей пакета. На этой диаграмме каждый блок представляет собой заголовок, добавляемый уровнем модели OSI.

К примеру, в пакете ICMP самый верхний блок – канальный, а два нижних – сетевые. В каждом блоке есть соответствующие поля.

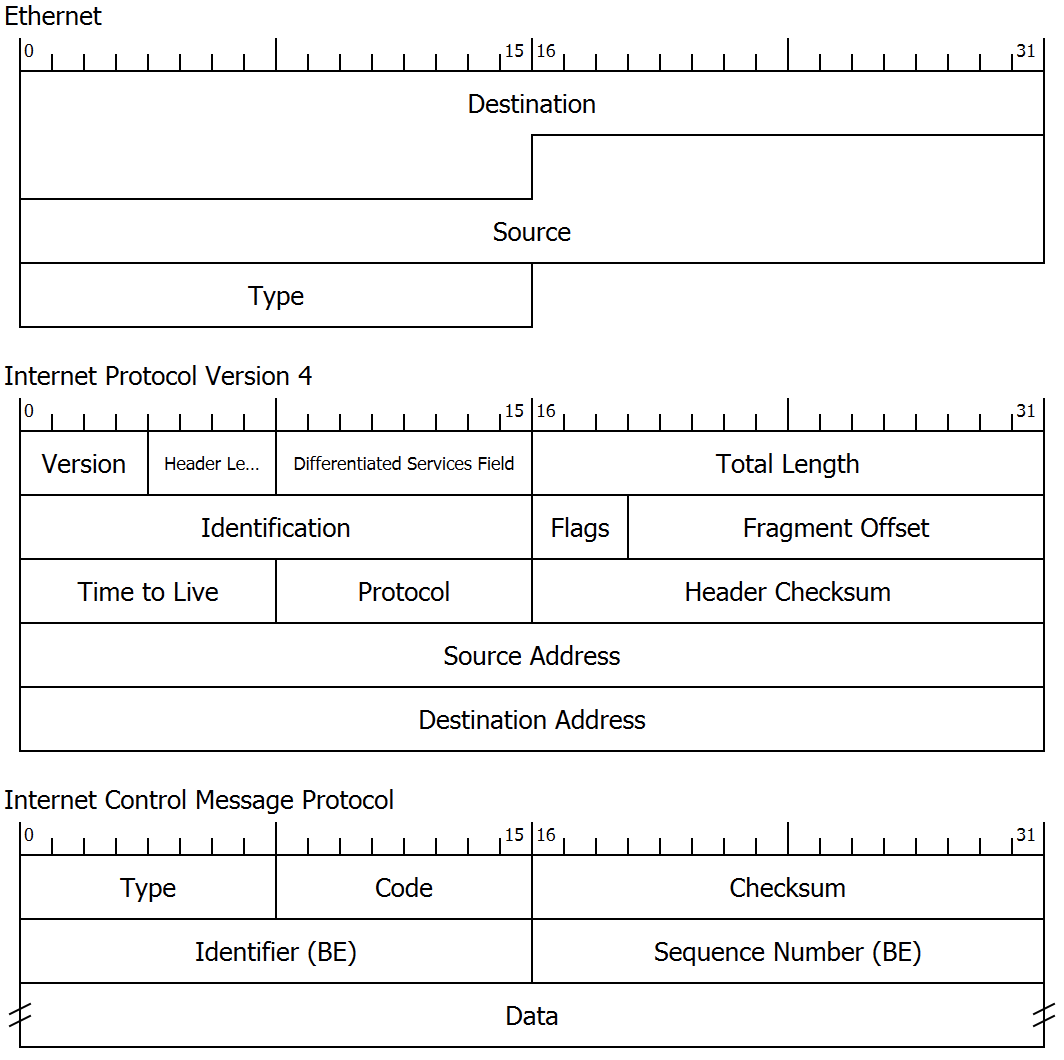


Рисунок 20 – Диаграмма TCMP – пакета

Диаграмма TCP – пакета:

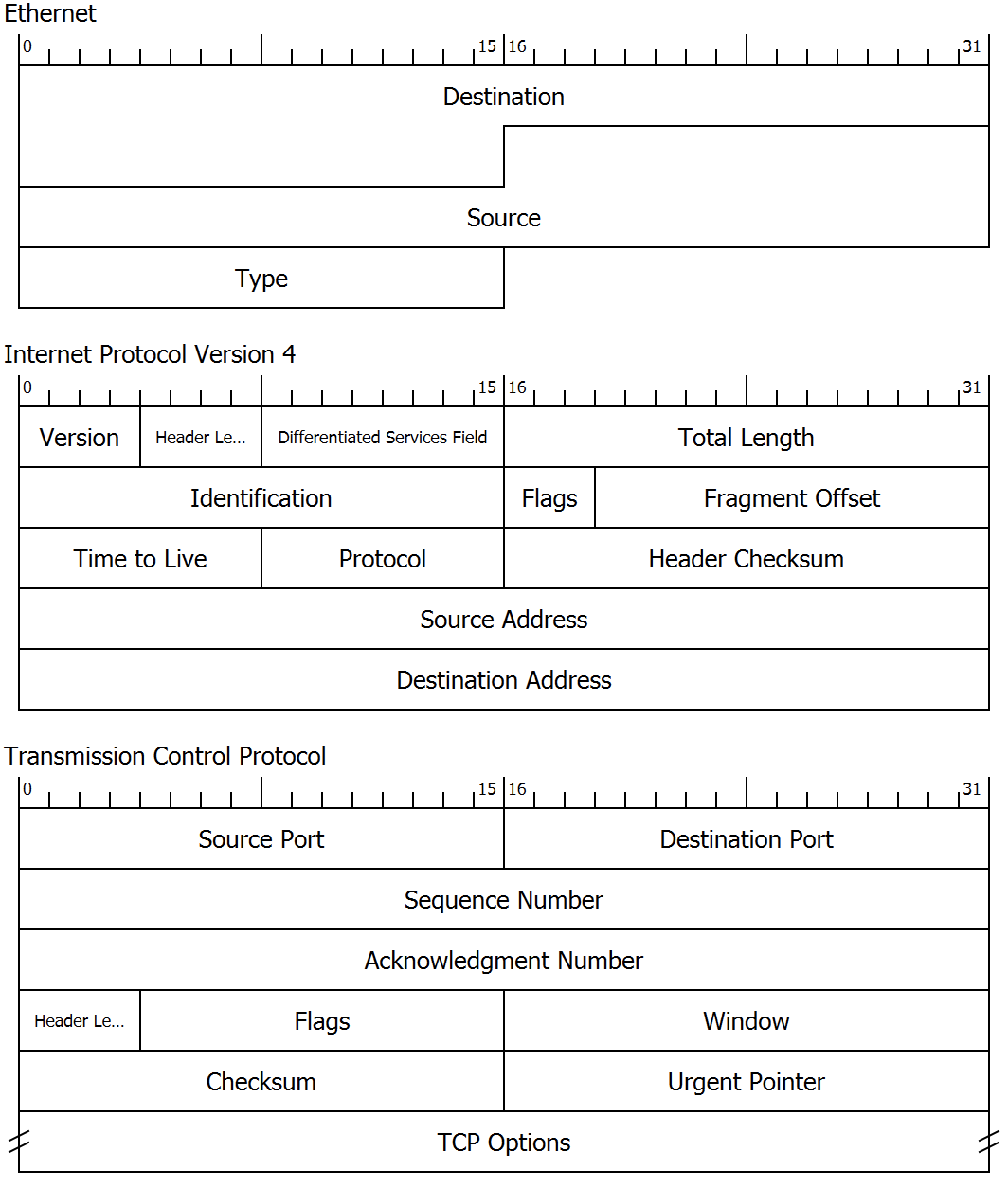


Рисунок 21 – Диаграмма TCP – пакета

Диаграмма HTTP – пакета:

Таблица 1 – Диаграмма HTTP - пакета

|  |  |
| --- | --- |
| **Протокол** | **Уровень** |
| Ethernet | Канальный |
| Internet Protocol Version 4 | Сетевой |
| Transmission Control Protocol | Транспортный |
| Transport Layer Security | Представления |
| Hypertext Transfer Protocol | Прикладной |

Нажав на диаграмму правой кнопкой мыши, можно увидеть значения полей:

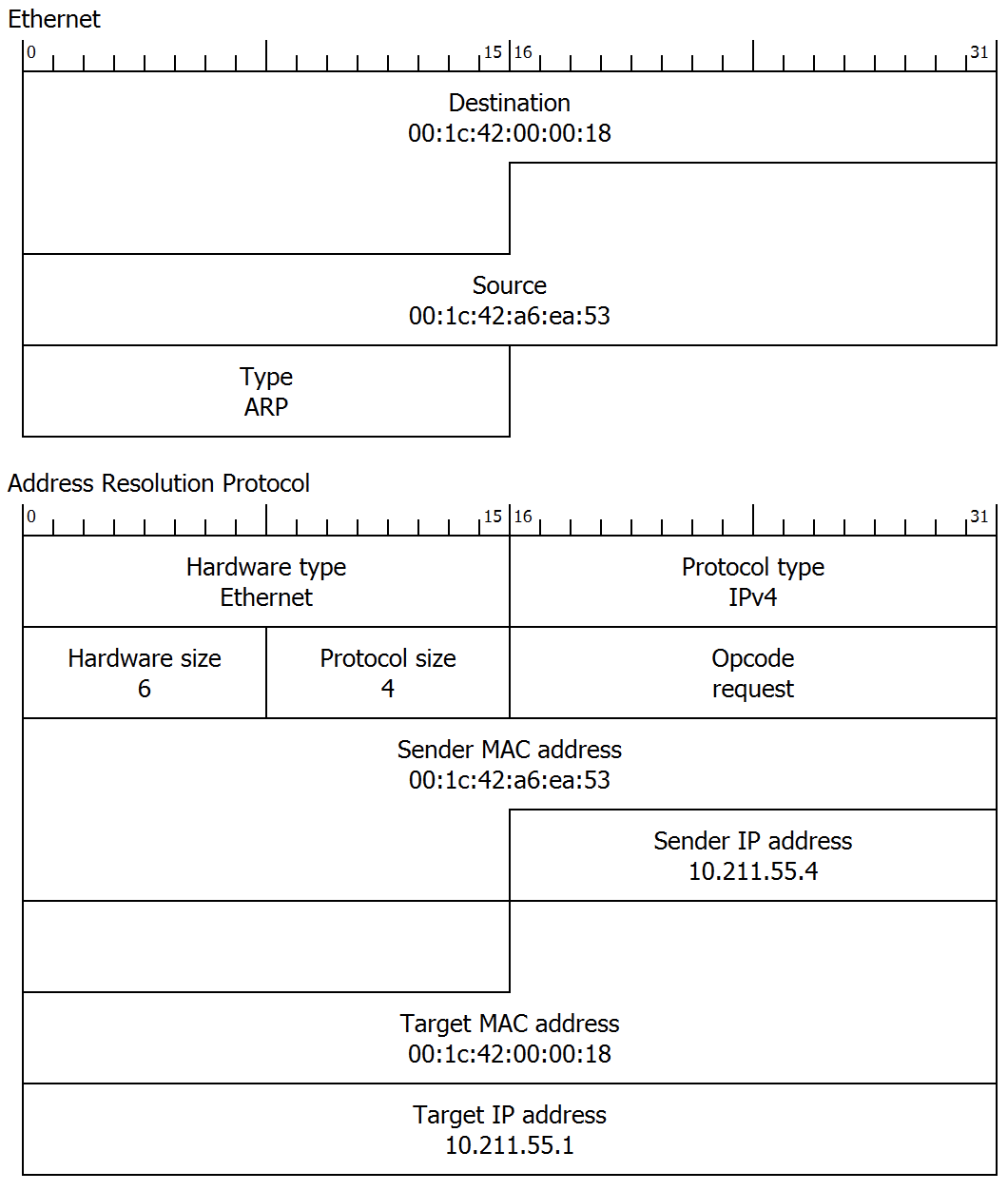


Рисунок 22 – Поля пакета

В левой нижней области расположены поля пакета, сгруппированные по протоколам. Первая группа всегда содержит метаданные, предоставленную самой программой, а дальше группы расположены в порядке возрастания: сначала идут протоколы канального уровня, затем сетевого, и так далее.

Каждую группу можно развернуть, чтобы получить подробные сведения. У заголовка каждого протокола есть свои собственные поля.

У пакета TCP, к примеру, в заголовке порта – адреса отправителя и получателя, а в TCP – порты.

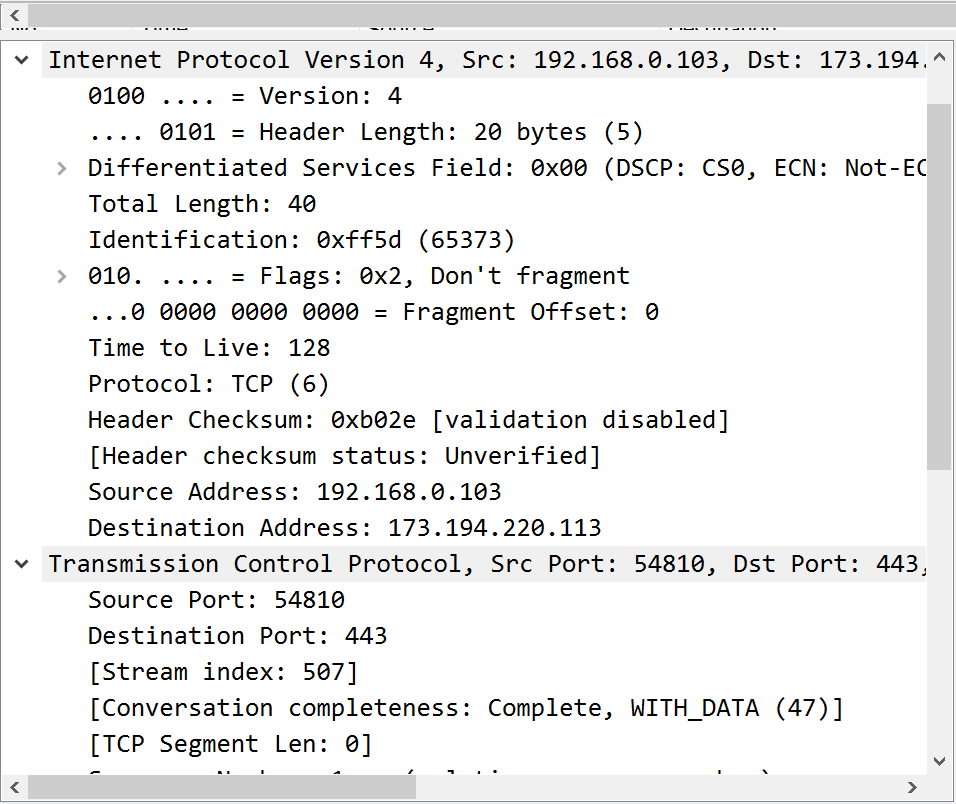


Рисунок 23 – Поля пакета протокола TCP

В полях протокола ICMP пакета можно найти коды успеха или ошибки.

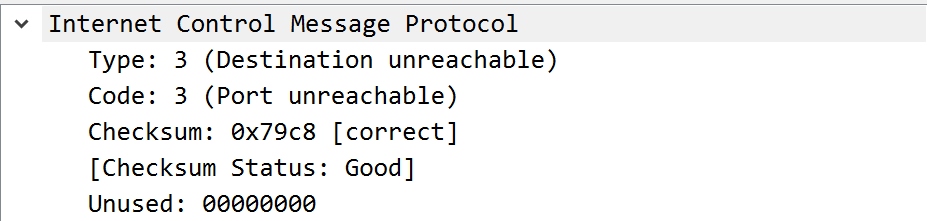


Рисунок 24 – Надпись о состоянии пакета

А в полях ARP – результирующие физические адреса.

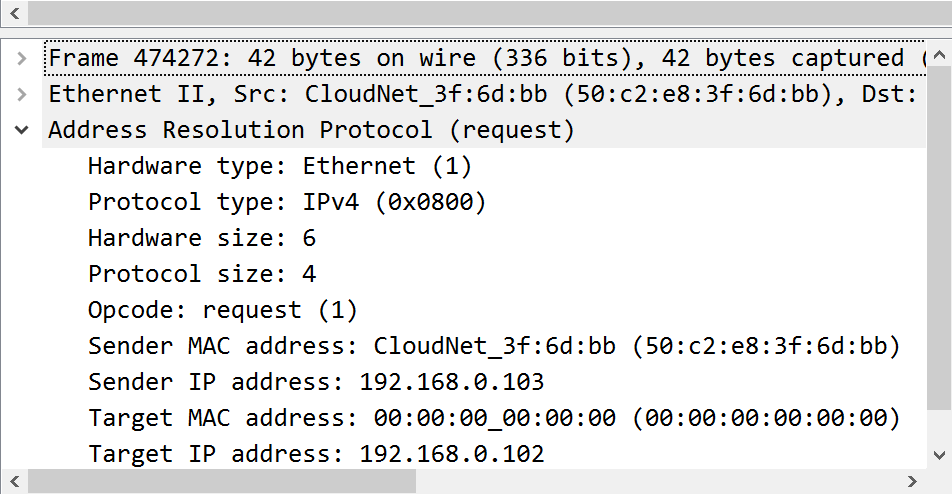


Рисунок 25 – Результирующие физические адреса

Теперь, попробуем захватить пакеты, которые мы будем отправлять, в реальном времени. Для начала, выполним команду ping в командной строке и словим ICMP сигнал с помощью фильтра.

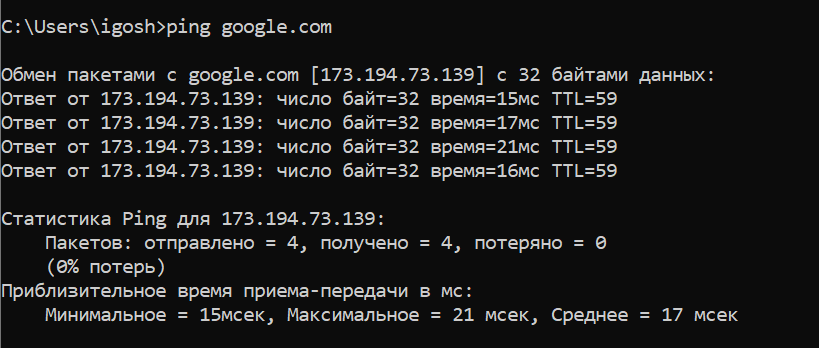


Рисунок 26 – Команда Ping

Всего получается 8 пакетов: 4 отправлено и 4 получено. Выделив пакет, в его полях написан адрес отправителя и получателя.

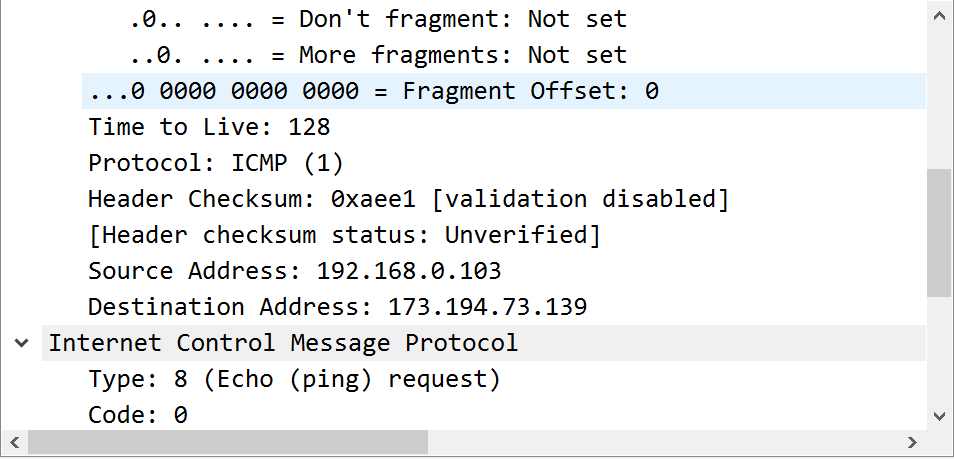


Рисунок 27 – Отправитель/получатель

Найдем в поисковике данный адрес.

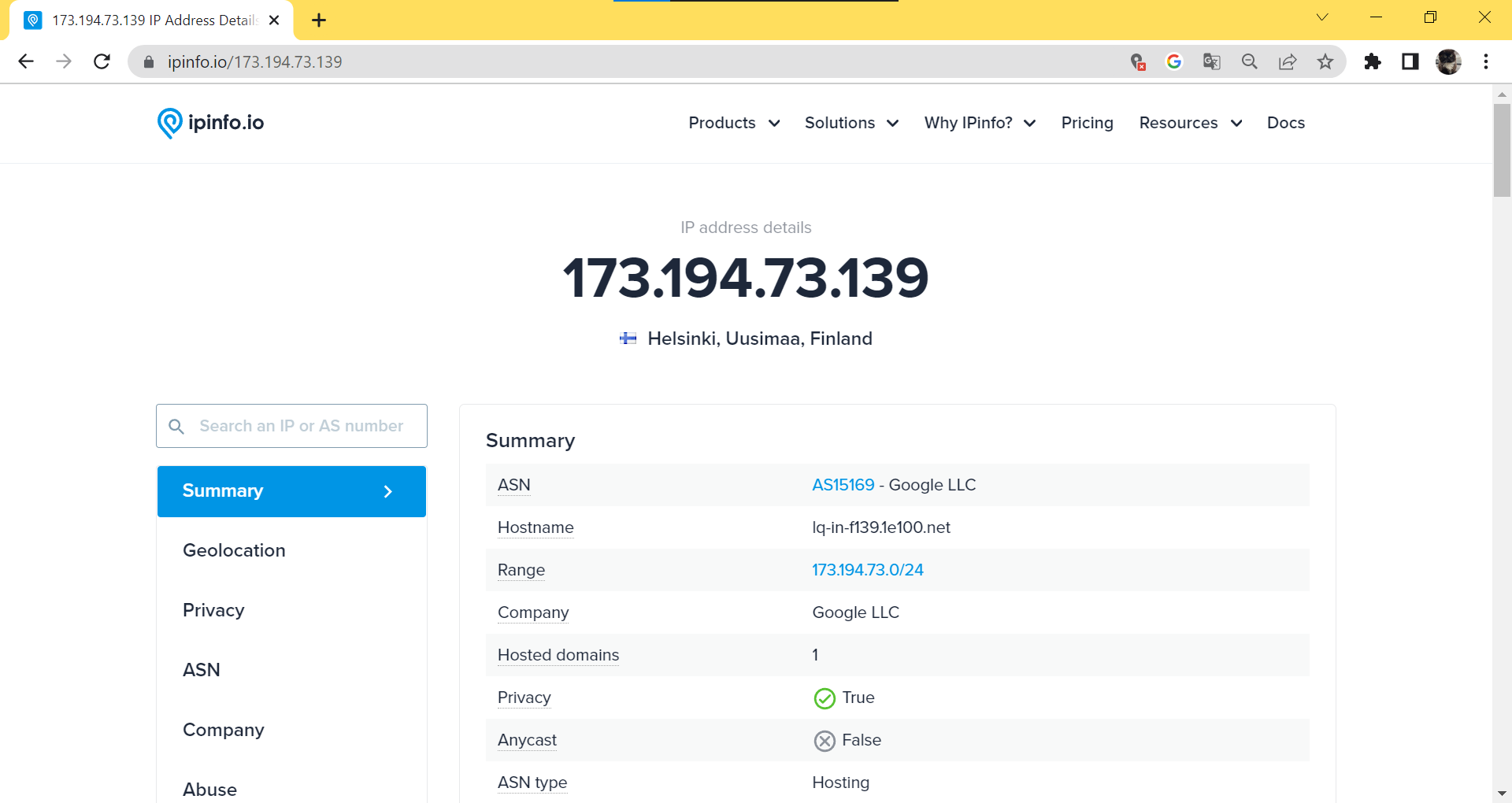


Рисунок 28 – Адрес получателя

Как видим, это действительно адрес Google!

Также протокол ICMP может нести данные, и мы видим их в области байтов пакета: это 32 английские буквы по алфавиту. Таким образом мы убедились, что программа может захватывать отправляемые пакеты и отображать информацию, невидимую для конечного пользователя.

Теперь, попробуем добавить фильтр HTTP и открыть браузер, найдя сайт без защиты.

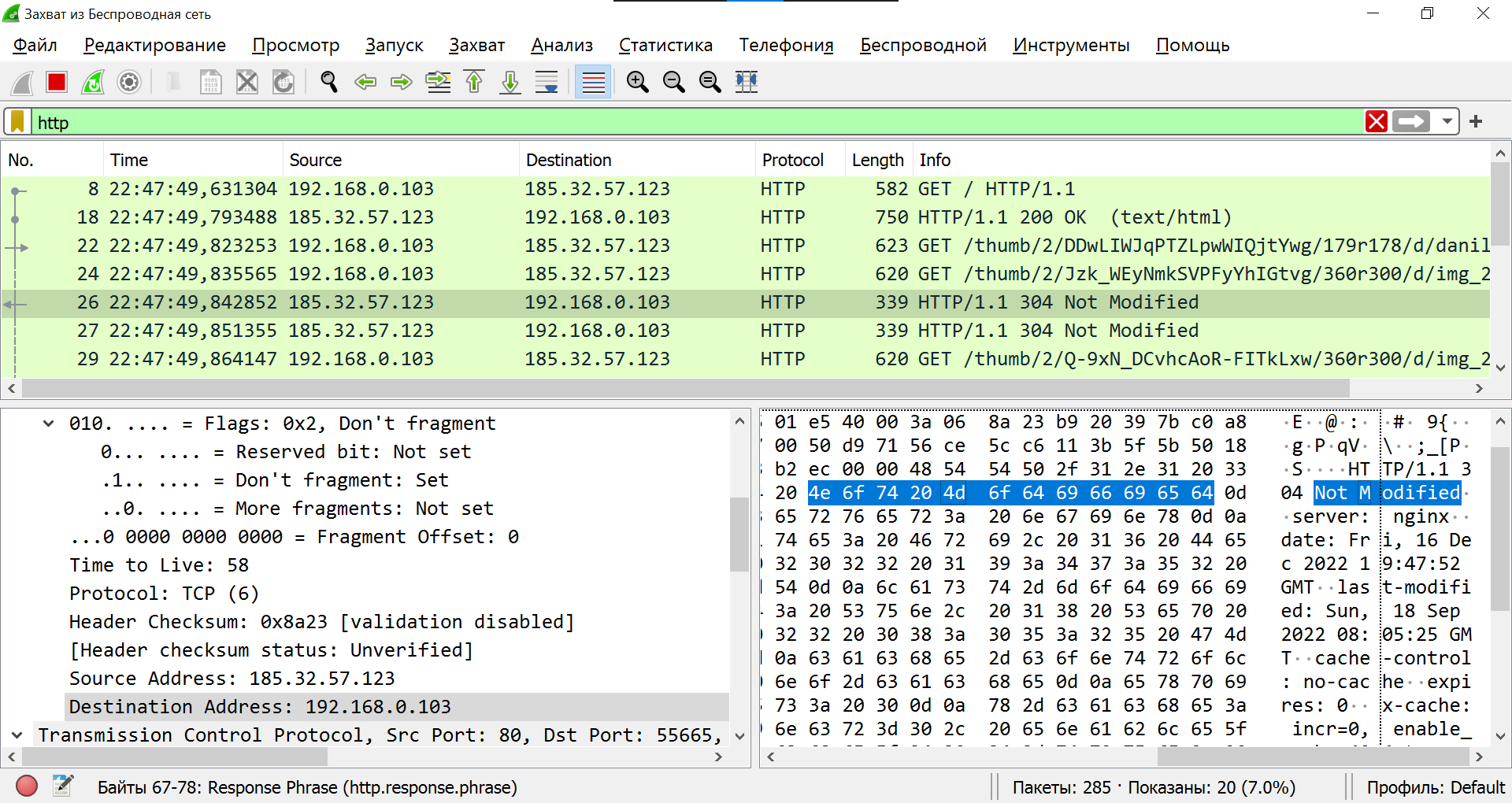


Рисунок 29 – HTTP – пакет

В колонке информации мы видим методы, адреса и типы данных.

В области байтов пакета мы видим хорошо различимый текст, а в полях пакета мы видим адрес, на который делаем запрос, агент-пользователя, и много других полезных и интересных данных.

Кроме того, в Wireshark можно определить входящий/исходящий пакет, который обозначен стрелочками. Направо – запрос, налево – ответ.

Точкой обозначен пакет, каким-либо другим образом относящийся к выбранному.



Рисунок 30 – Стрелочки

Нажав на пакет и выбрав Следовать -> Поток HTTP мы сможем увидеть полный диалог клиента и сервера:

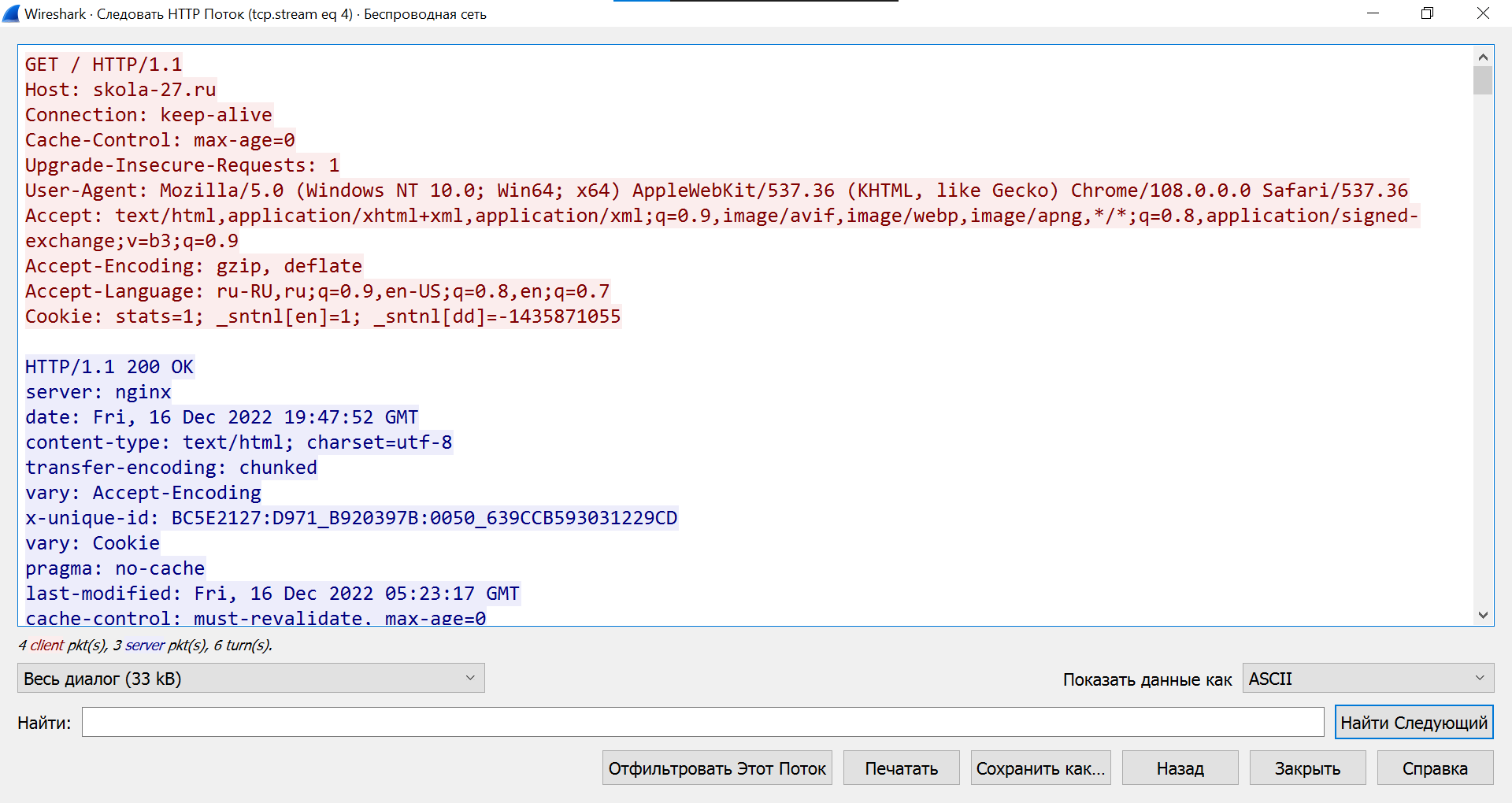


Рисунок 31 – Диалог клиента и сервера

Красный цвет – запрос, синий цвет – ответ, если диалог продолжается, то ниже есть еще блоки текста. Таким образом, этот пример показывает захват пакетов в реальном времени.

Выполнив незашифрованный POST – запрос, таким образом могут украсть наши данные:

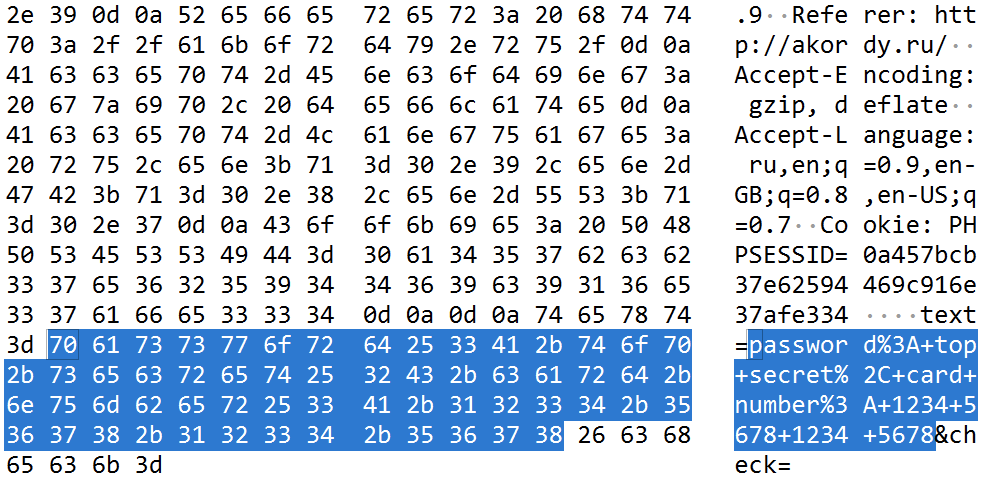


Рисунок 32 – Пароль в пакете

Что же делать с HTTPS? Данные будут зашифрованы, и разобрать их не получится, ибо похоже это будет на текстовую кашу.

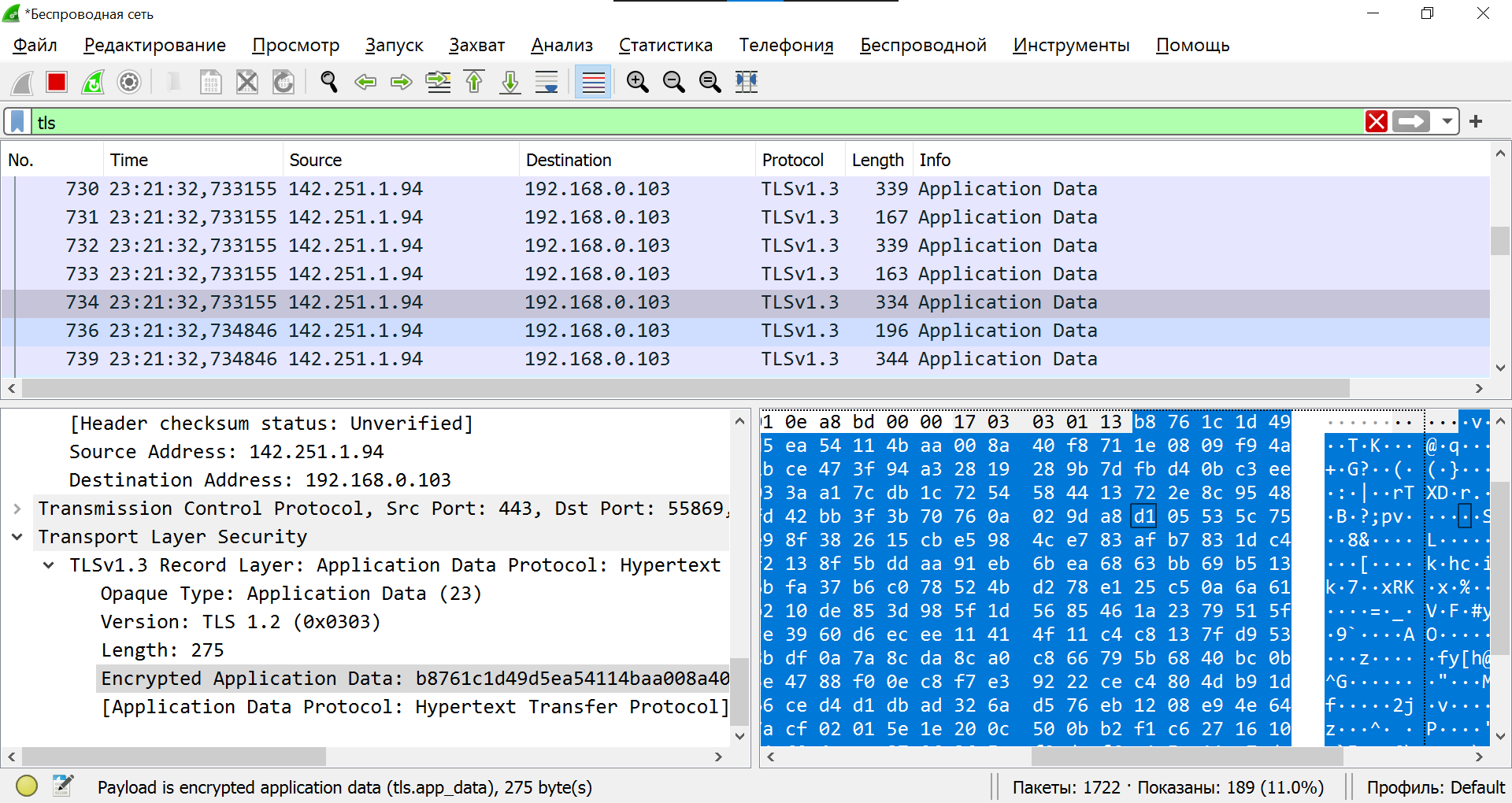


Рисунок 33 – Шифрование TLS – пакета

Но выход всё же есть. Для этого нужно подобрать ключи, которые мы возьмем из репозитория по адресу: <https://github.com/pan-unit42/wireshark-tutorial-decrypting-HTTPS-traffic>. В архиве находятся два файла:

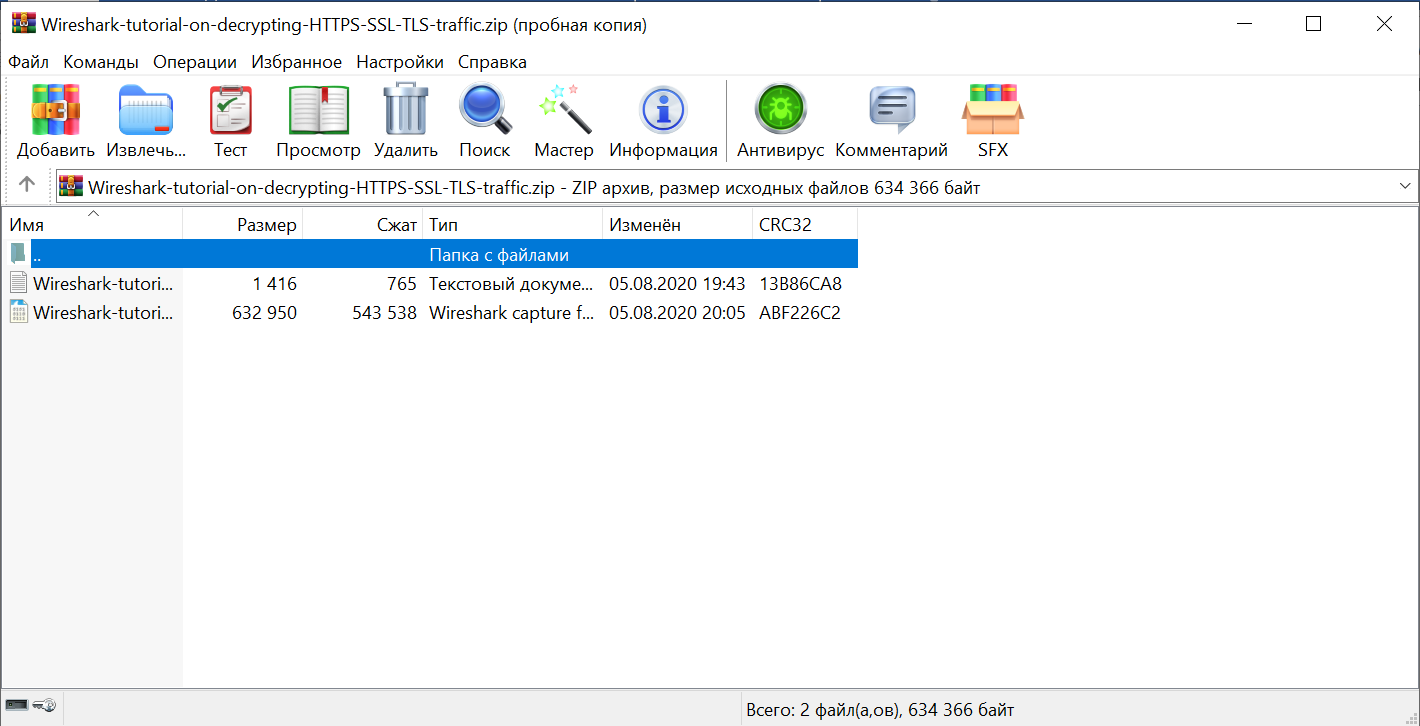


Рисунок 34 – Файлы из репозитория

Первый из них – файл дампа tcpdump. Этот файл содержит список пакетов, захваченных кем-либо до этого. Второй файл представляет из себя ключи, которые мы будем использовать для расшифровки TLS.

Запускаем второй файл, который откроется в Wireshark.

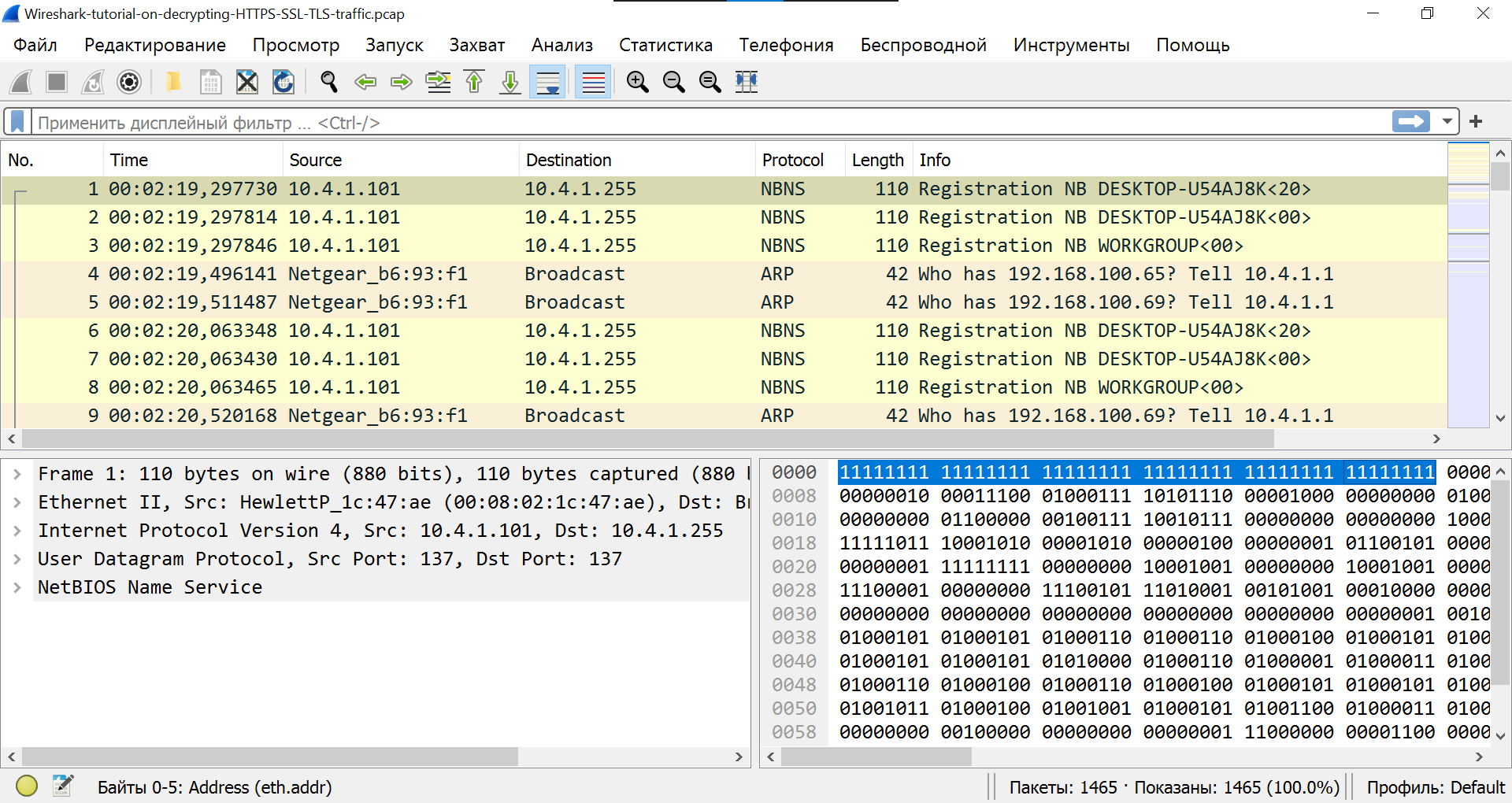


Рисунок 35 – Открытый файл

Открыв файл, мы видим в нём 1465 пакетов. Однако, отсортировав его по html, пакетов мы не увидим. Фильтрация по порту 80 результатов не дает, однако порт 443 выдает множество результатов.

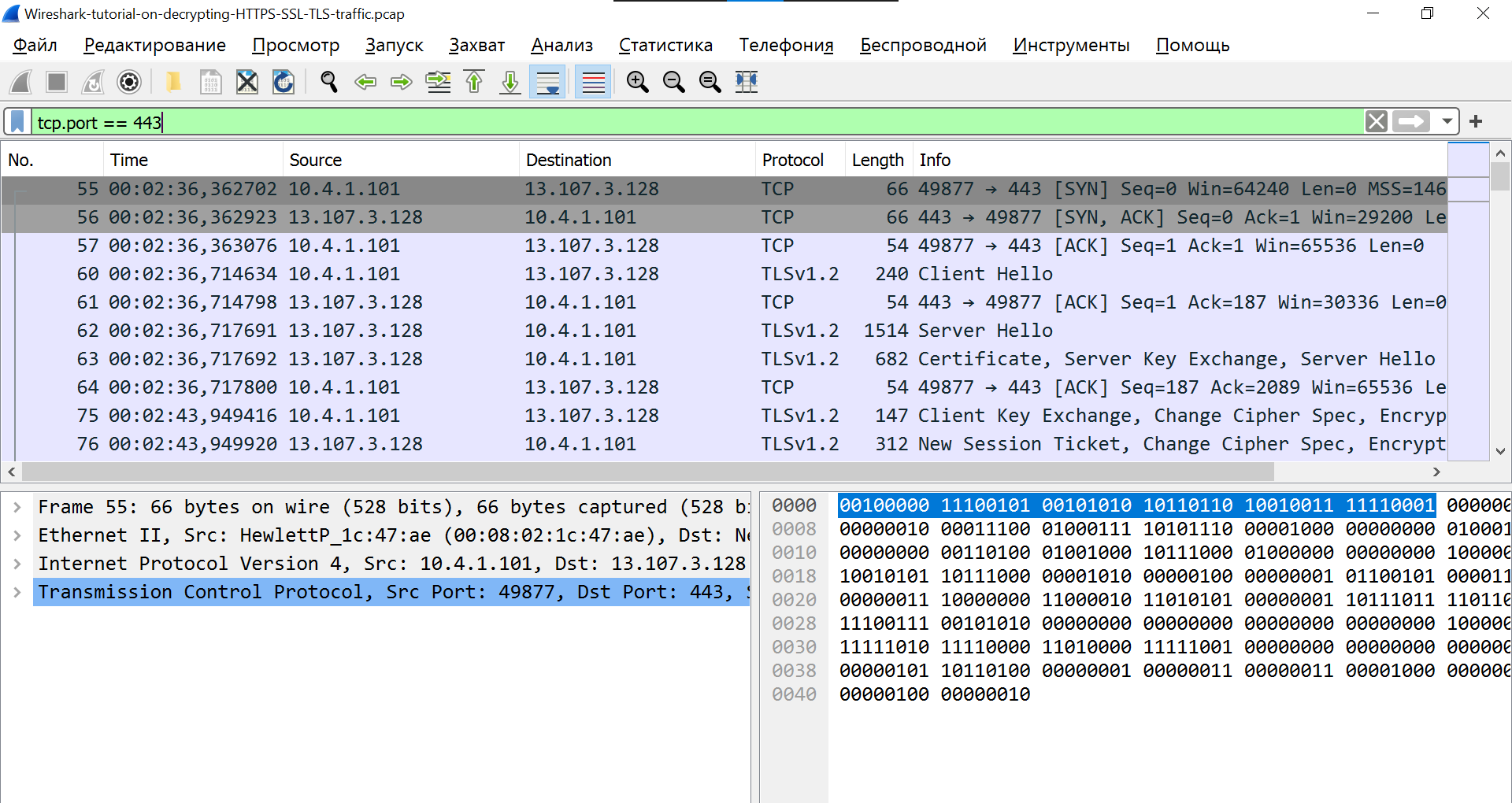


Рисунок 36 – Port 443

Теперь, нужно импортировать ключи из файла через Редактирование – Параметры – Protocols – TLS -(Pre)-Master-Secret log filename и прописываем сюда путь файла с ключами.

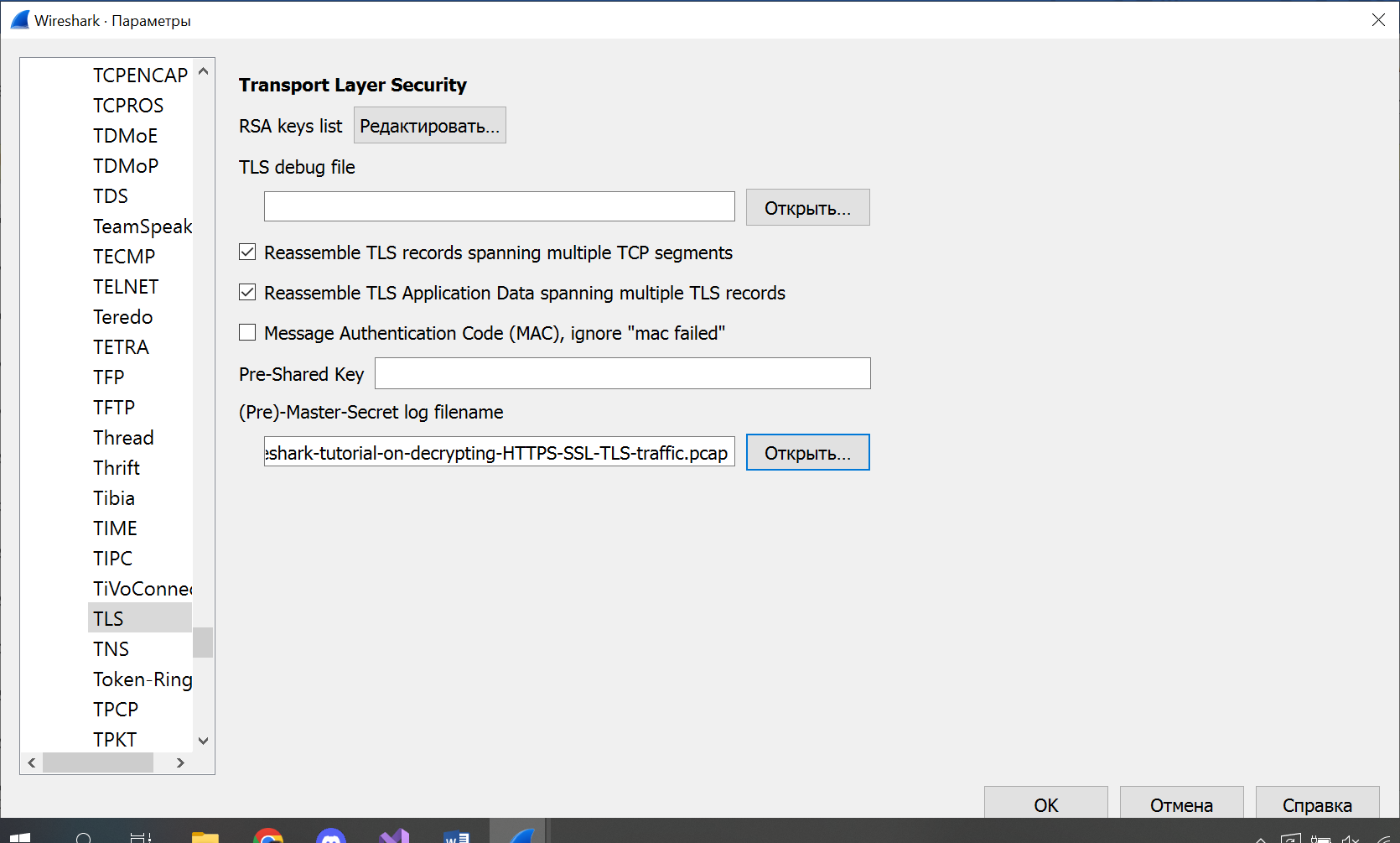


Рисунок 37 – Параметры TLS

Теперь, фильтруем пакеты по HTTP. У нас появились пакеты, из-за того, что Wireshark смог их расшифровать.

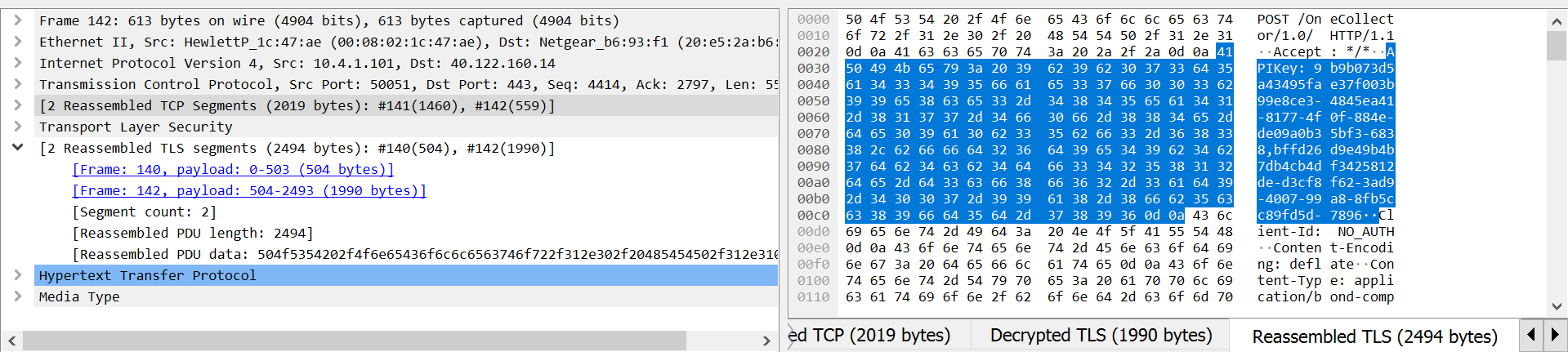


Рисунок 38 – Расшифрованные HTTP

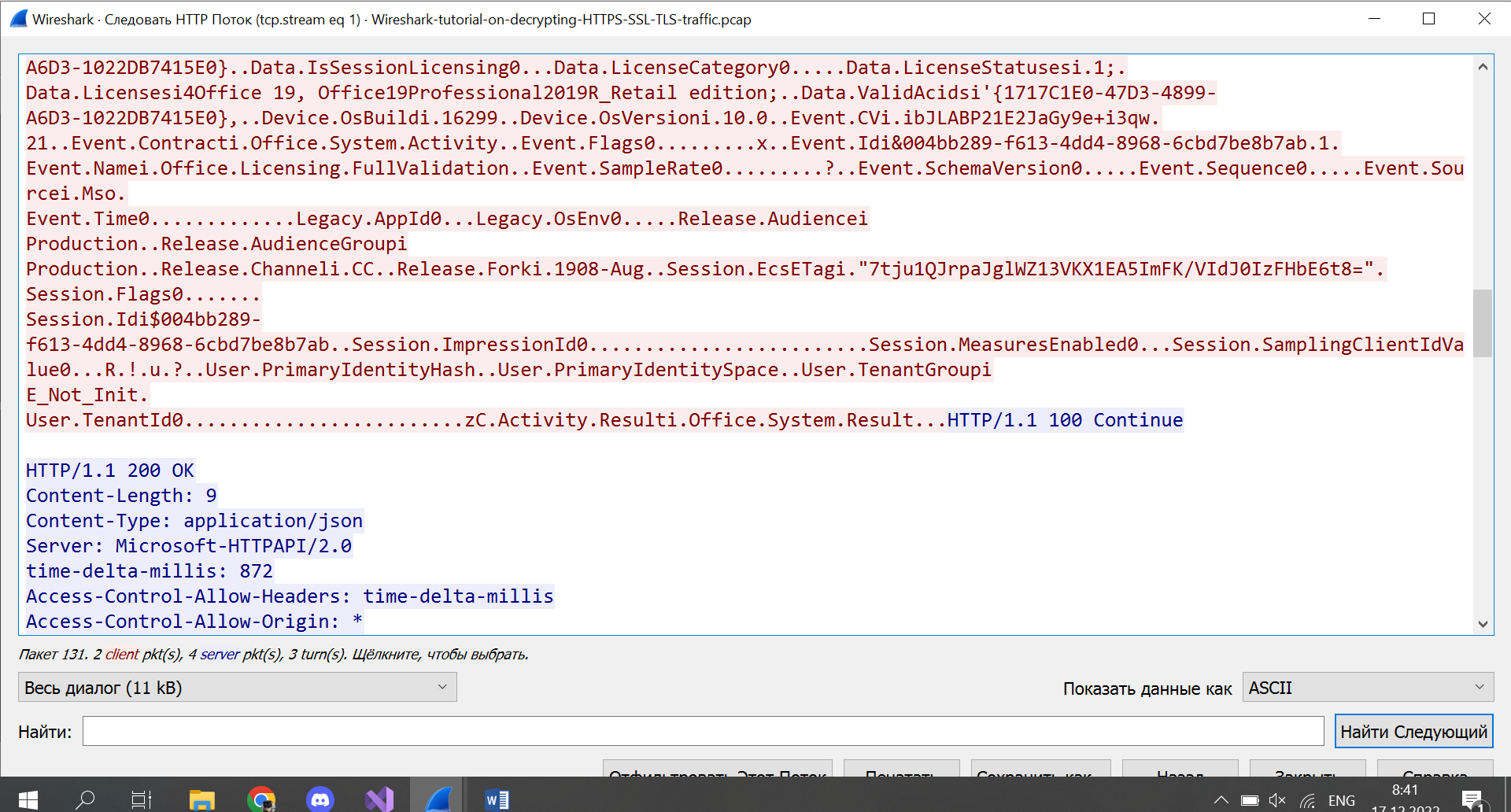


Рисунок 39 – Следование по потоку

Как мы видим, в этом пакете передавался файл вида xml, который Wireshark успешно расшифровал.

Вывод: научились использовать Wireshark для просмотра трафика, рассмотрели его функции и возможности, выполнили практическое задание.