Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования

«Поволжский государственный технологический университет»

# Анализ протокола HTTP

отчёт по лабораторной работе №3

# по дисциплине «Технологии программирования и создание WEB

# приложений»

Исполнитель:

студент гр. ИСТ-34

Джумаев Бобур

Проверил:

/ Чесноков Сергей Евгеньевич

(подпись) (Ф.И.О. полностью)

Йошкар-Ола

2024

### Цели работы

**Изучение основ работы протокола HTTP**: понять основные принципы функционирования HTTP, его методы (GET, POST и др.) и структуру запросов и ответов.

**Овладение навыками работы с Wireshark**: научиться использовать Wireshark для захвата и анализа сетевого трафика.

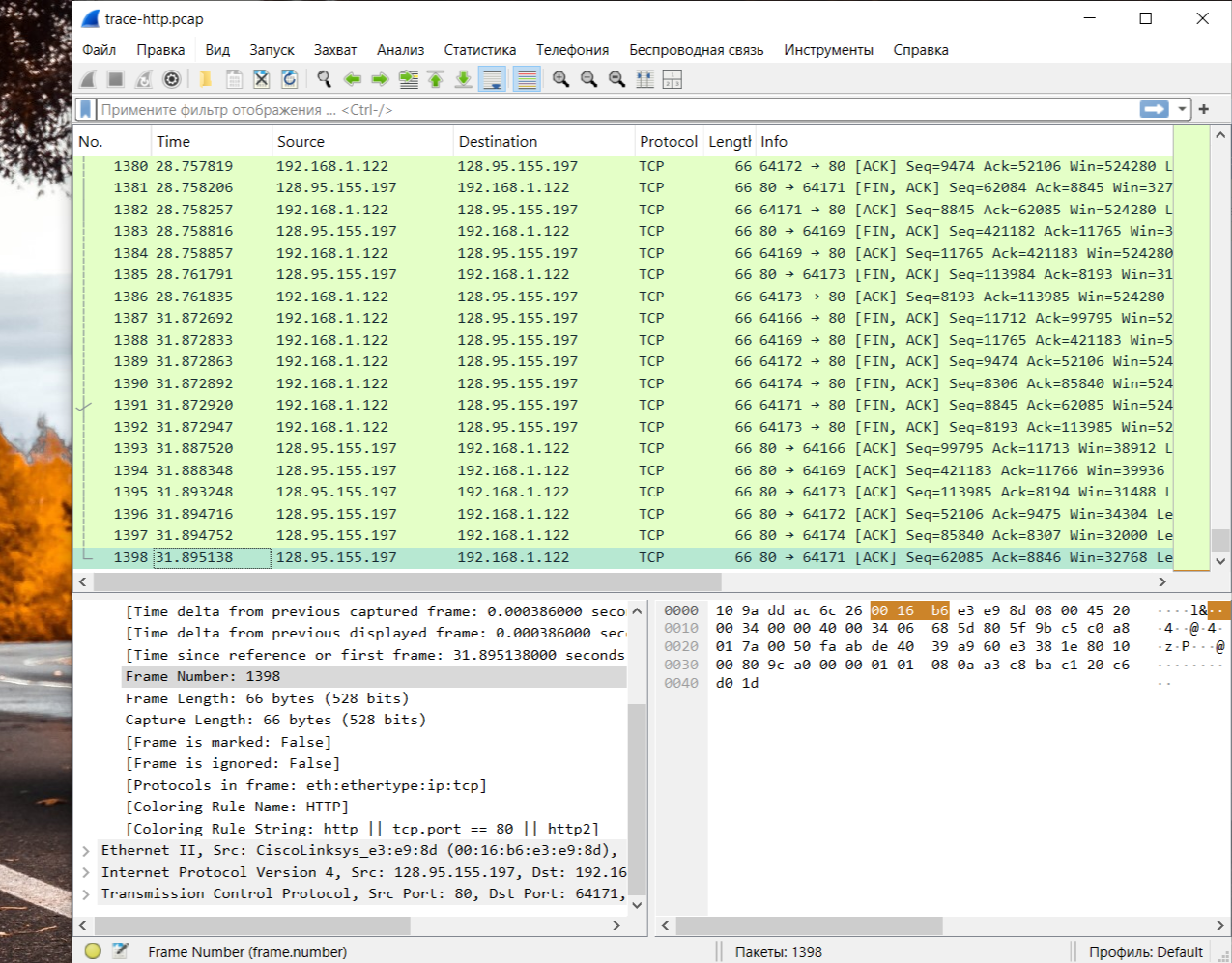
**Анализ HTTP-запросов и ответов**: научиться идентифицировать и интерпретировать HTTP-запросы и ответы в захваченном трафике.

**Изучение кодов состояния HTTP**: ознакомиться с различными кодами состояния ответов HTTP и их значениями.

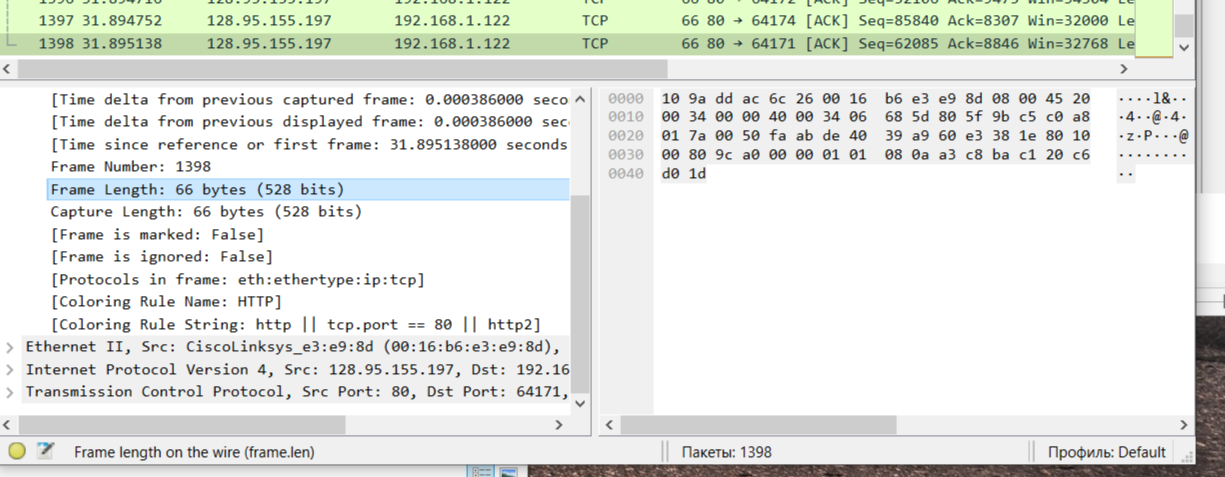
**Исследование заголовков HTTP**: понять роль заголовков в HTTP-запросах и ответах, изучить, как они влияют на поведение клиента и сервера.

## Практическая часть

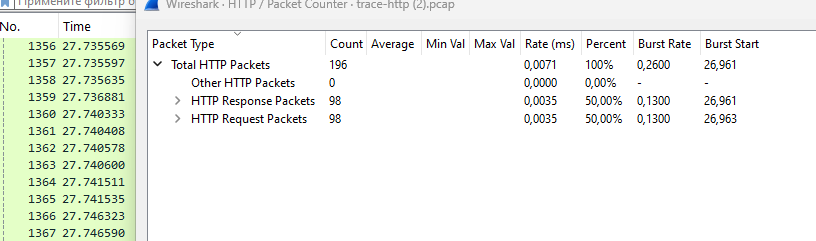
### Задание 1. Загрузка файла трассировки http



Захвачено 1398 пакетов при помощи Wireshark.



Обьем перехваченных пакетов: 66(байтов)

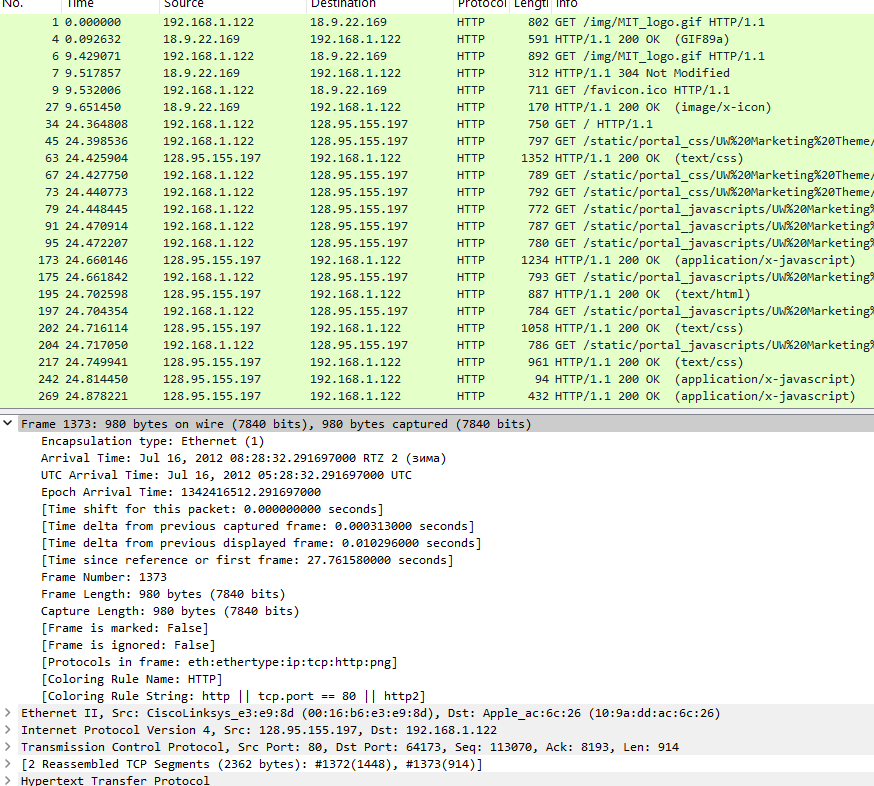


98 пакетов HTTP Request

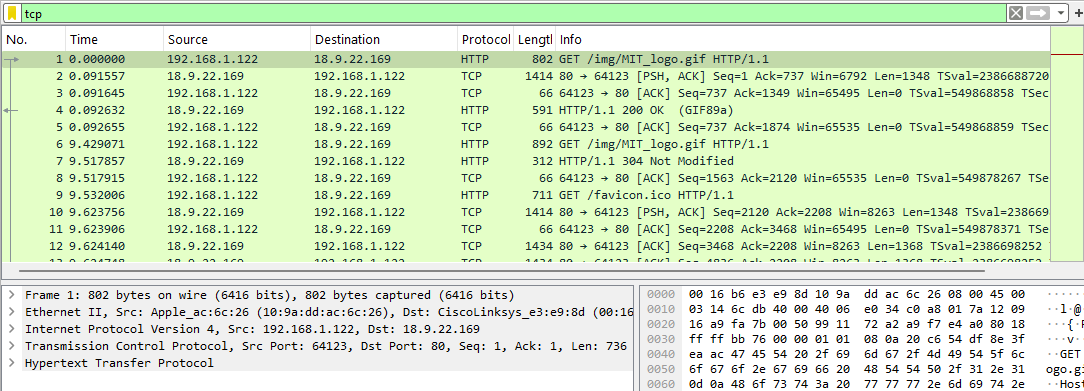
98 пакетов HTTP Response

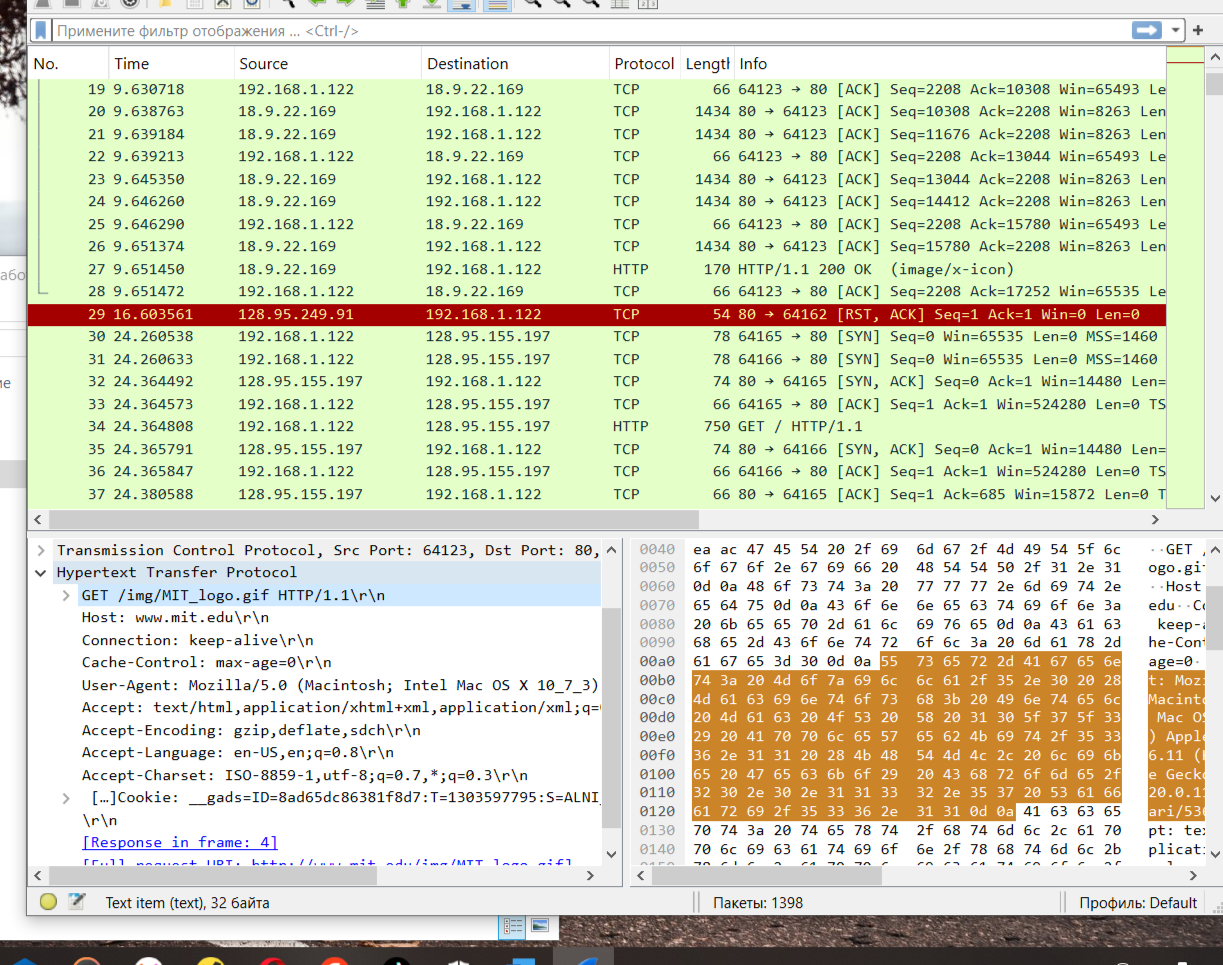
### Задание 2. Изучение трассировки запроса и ответа

Чтобы сосредоточиться на трафике HTTP, введем и применим выражение фильтра «http»:



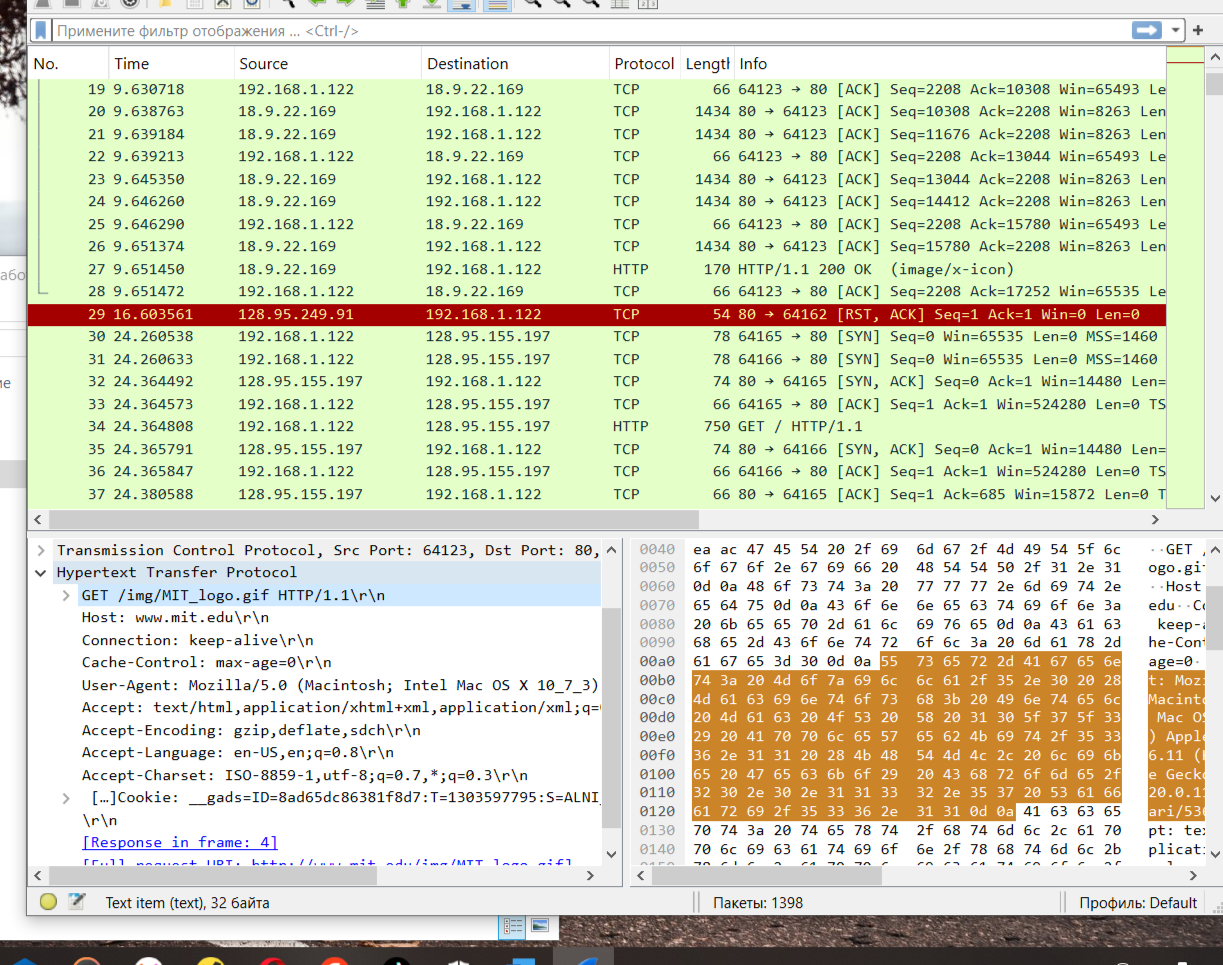
Чтобы сосредоточиться на трафике TCP, введем и применим выражение фильтра «tcp»:

Чтобы сосредоточиться на трафике TCP или UDP с использованием порта 80введем и применим выражение фильтра «tcp.port == 80 || udp.port == 80»:

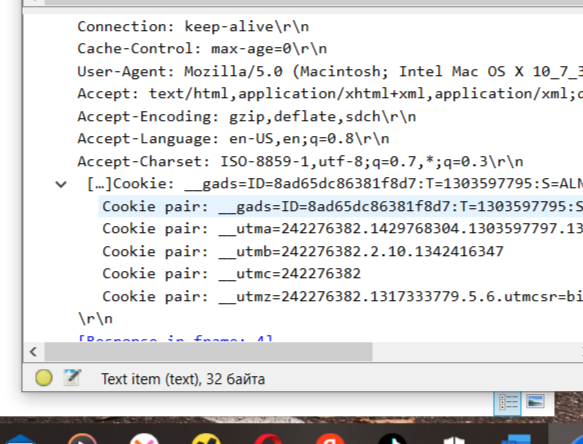


**Просмотр GET запроса и изучение HTTP-заголовка**

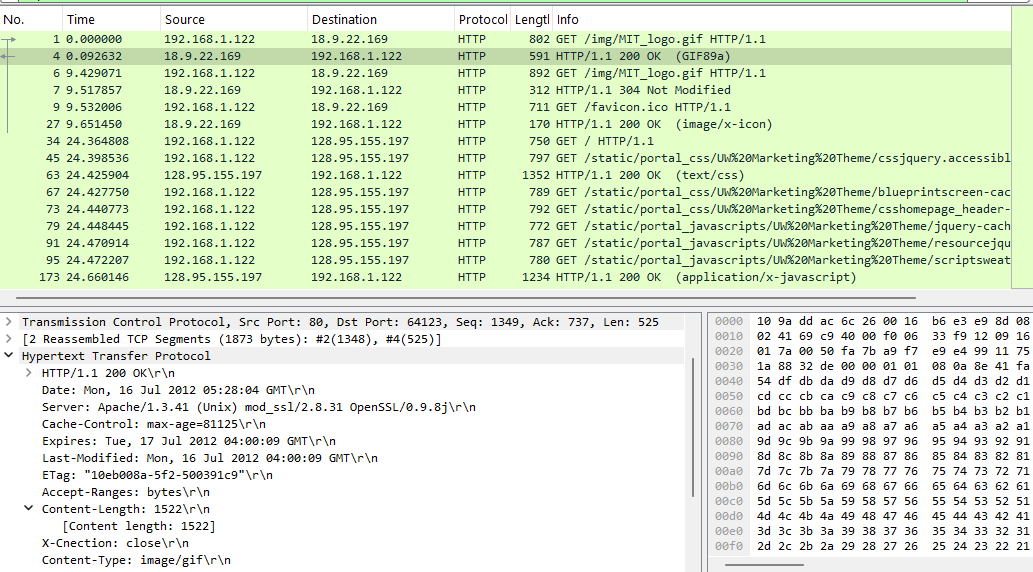
Это позволит нам изучить детали HTTP-запроса. Изучите HTTP-заголовок, развернув стрелку вниз рядом с **Протокол передачи гипертекста (Hypertext Transfer Protocol)** в нижней панели.



* **Host.** Обязательный заголовок, он идентифицирует имя (и порт) сервера.
* **User-Agent.** Вид браузера и его возможности.
* **Accept, Accept-Encoding, Accept-Charset, Accept-Language.**Описания форматов, которые будут приняты в ответе, например, text/html, включая его кодировку, например, gzip, и язык.
* **Cookie.**Имя и значение cookie-файлов, которые браузер хранит для веб-сайта.
* **Cache-Control.**Информация о том, как ответ может быть кэширован:

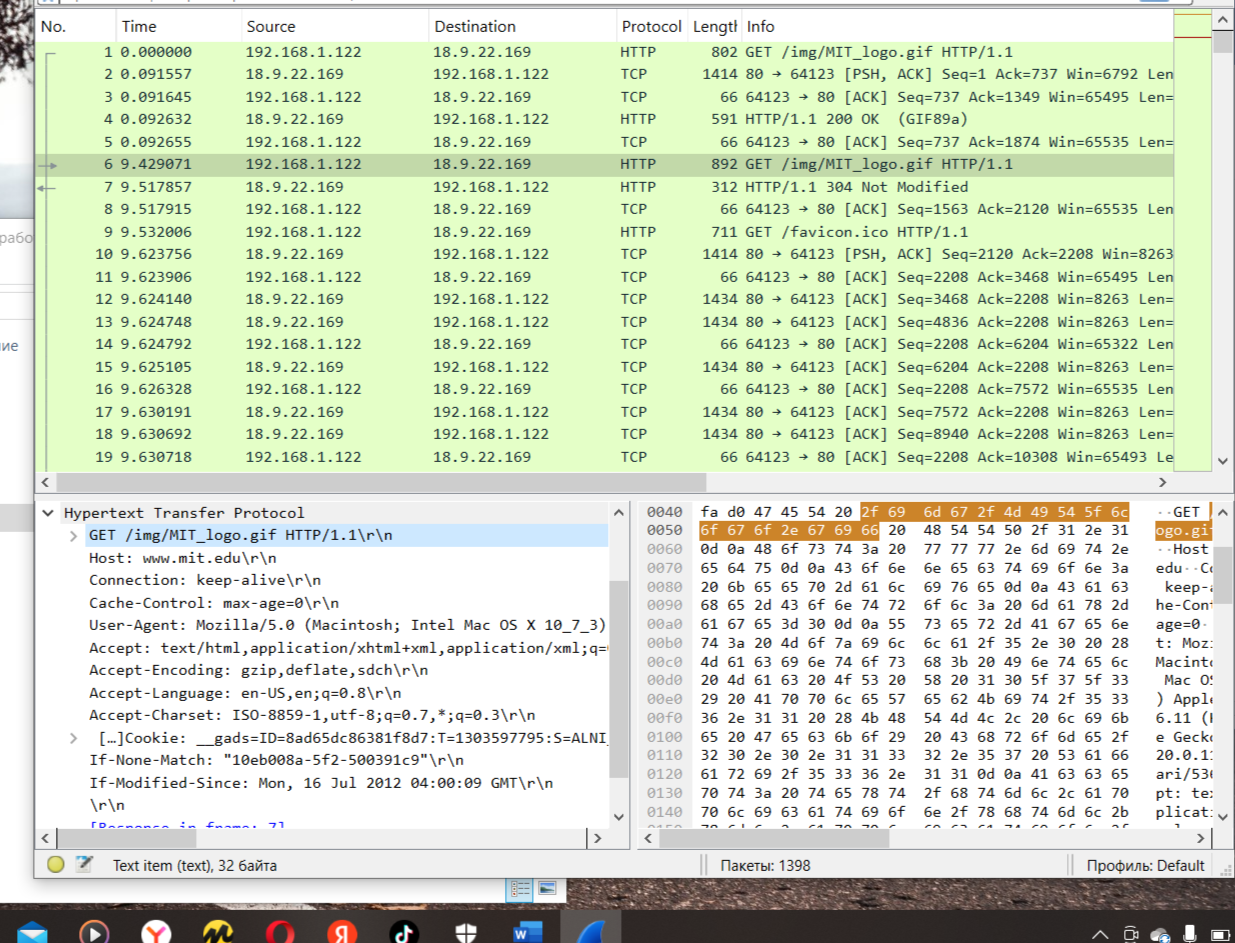


**Просмотр GET ответа**

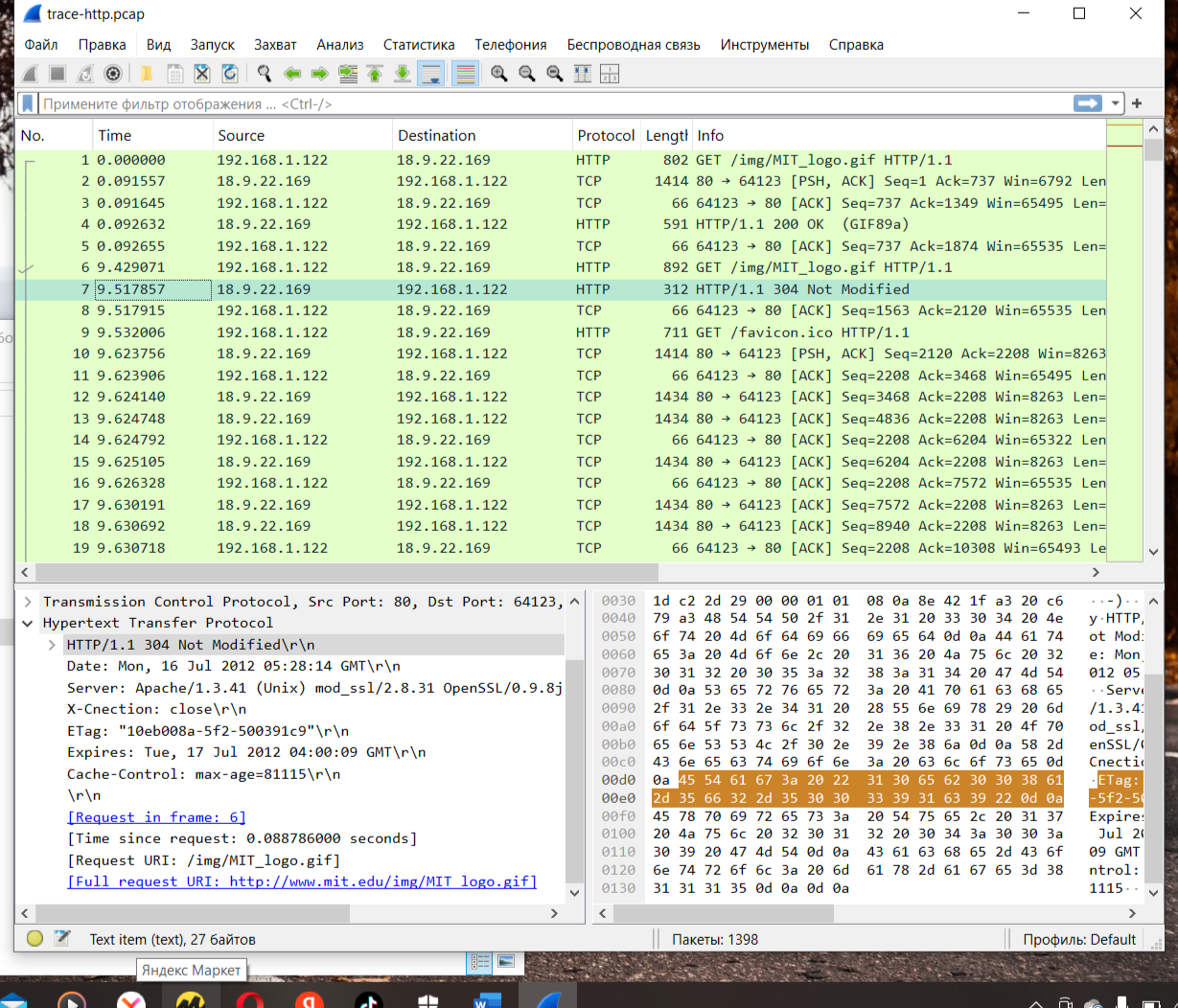


* **Server.** Тип сервера и его возможности.
* **Date, Last-Modified.** Время ответа и время последнего изменения контента.
* **Cache-Control, Expires, Etag.**Информация о том, как можно кэшировать ответ.

Задание 3. Кэширование контента



Заголовок называется «**If-Modified-Since**», т.е. он просит сервер отправить контент, если он был изменен с определенного времени.



Мы видим, что кэширование сработало, как и ожидалось, этот ответ не будет содержать контент. Вместо этогокод статуса ответа — «**304 Not Modified**». Это сообщает браузеру, что контент не изменился.

Gif изображения по которому обрабатывался запрос:

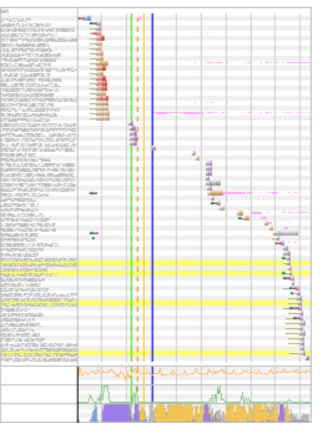


Задание 4. Сложные страницы

Тестирование WWW.WORDPRESS.COM, с которым мы работали при выполнении первой практической работы (сетевые утилиты) с помощью сервиса **PageSpeed:**

Изображение выглядит как текст, Шрифт, линия, снимок экрана

Автоматически созданное описание

****

### 

**Результаты тестирования**

**На диаграмме видно, что загрузка страницы включает следующие этапы:**

**HTML: Начальная загрузка HTML-документа.**

**CSS: Загрузка стилей, необходимых для отображения страницы.**

**JavaScript: Загрузка и выполнение скриптов.**

**Изображения: Загрузка графических элементов.**

**Шрифты: Загрузка веб-шрифтов.**

**Прочие ресурсы: Загрузка дополнительных ресурсов, таких как видео и флеш-элементы.**

**Каждый горизонтальный бар на диаграмме представляет отдельный ресурс, а его длина указывает на время, затраченное на его загрузку.**

**Анализ**

**HTML: Быстрая загрузка, что является положительным показателем.**

**CSS: Загрузка занимает больше времени, что может указывать на необходимость оптимизации стилей.**

**JavaScript: Длительное время загрузки и выполнения скриптов может замедлять отображение страницы.**

**Изображения: Некоторые изображения загружаются медленно, что может быть связано с их большим размером.**

**Шрифты: Загрузка шрифтов происходит относительно быстро.**

**Прочие ресурсы: Видео и флеш-элементы могут значительно увеличивать время загрузки страницы.**

**Заключение**

**Анализ водопадной диаграммы позволяет веб-разработчикам понять, какие ресурсы замедляют загрузку страницы и как можно оптимизировать их загрузку. Это приводит к улучшению производительности сайта и повышению удовлетворенности пользователей.**