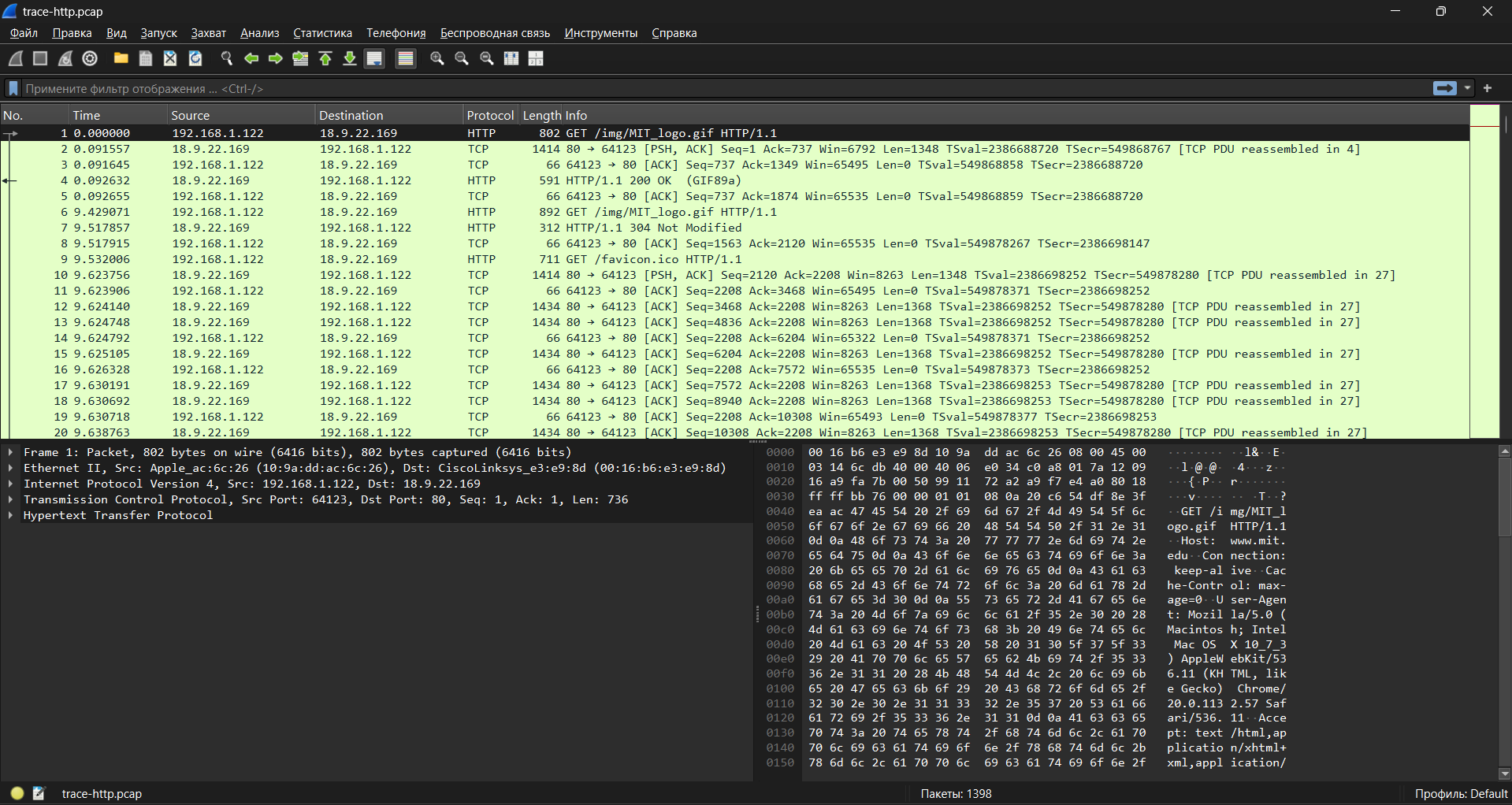
Лабараторная работа №3

**Задание 1**



1. Количество пакетов, которые были перехвачены Wireshark и присутствуют в этом файле?

В таблице в верхней части файла перечислены пакеты с номерами от 1 до 22.

Ответ: 22 пакета

2. Какой общий объем перехваченных пакетов (в байтах)?

Судя по повторяющимся записям, каждый пакет (Frame) имеет размер 802 байта ("802 bytes on wire", "802 bytes captured"). Умножаем размер одного пакета на их количество:

802 байта/пакет 22 пакета = 17644 байта.

Ответ: 17644 байта

3. Количество пакетов HTTP Request?

HTTP Request (запрос) обычно отправляется от клиента к серверу. В данном случае клиент — это 192.168.1.122, а сервер — 18.9.22.169.

Пакет №1: 192.168.1.122 -> 18.9.22.169, протокол HTTP. Это Request.

Пакет №6: 192.168.1.122 -> 18.9.22.169, протокол HTTP. Это Request.

Пакет №8: 192.168.1.122 -> 18.9.22.169, протокол HTTP. Это Request.

Ответ: 3 пакета

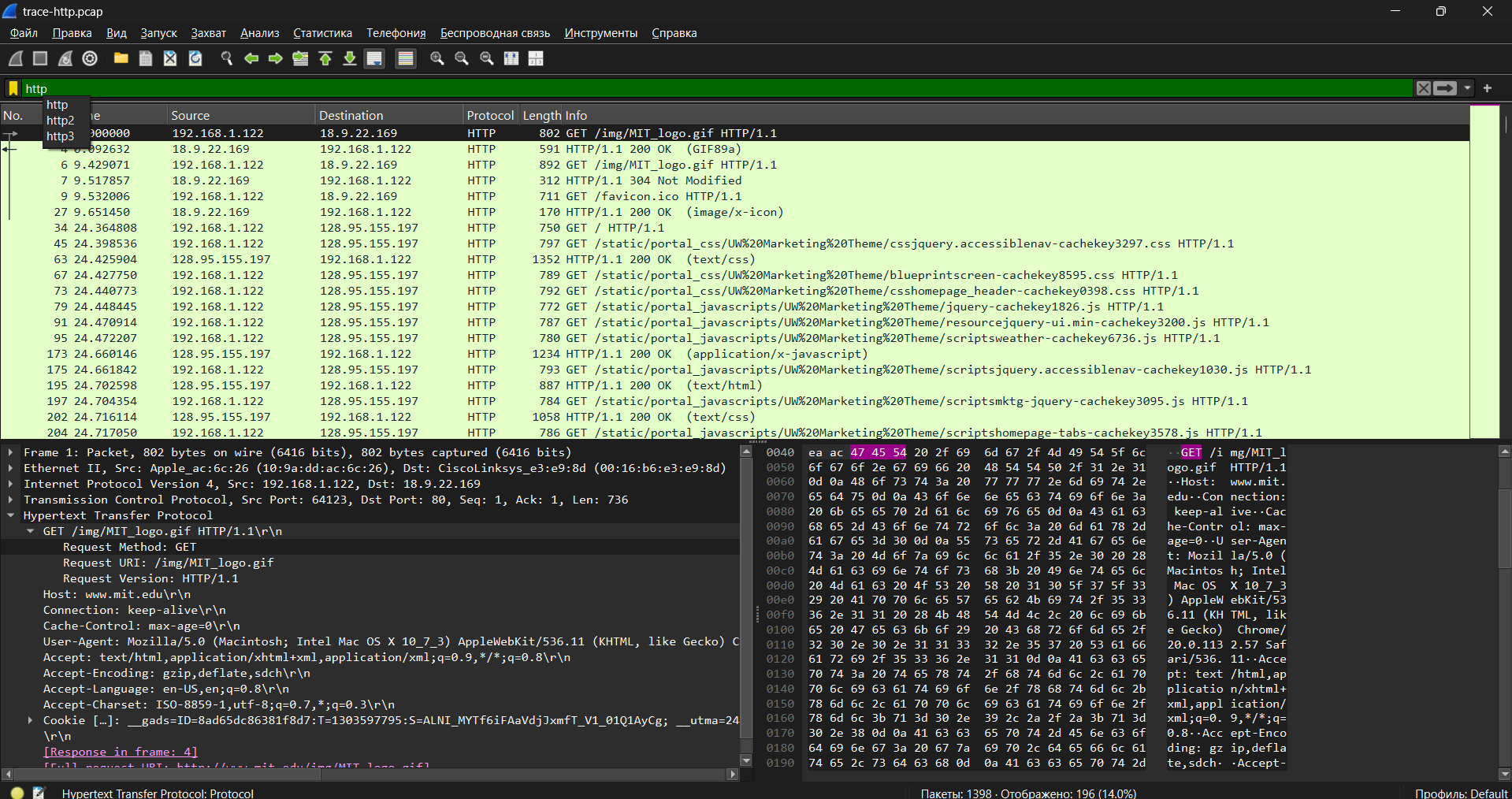
4. Количество пакетов HTTP Response?

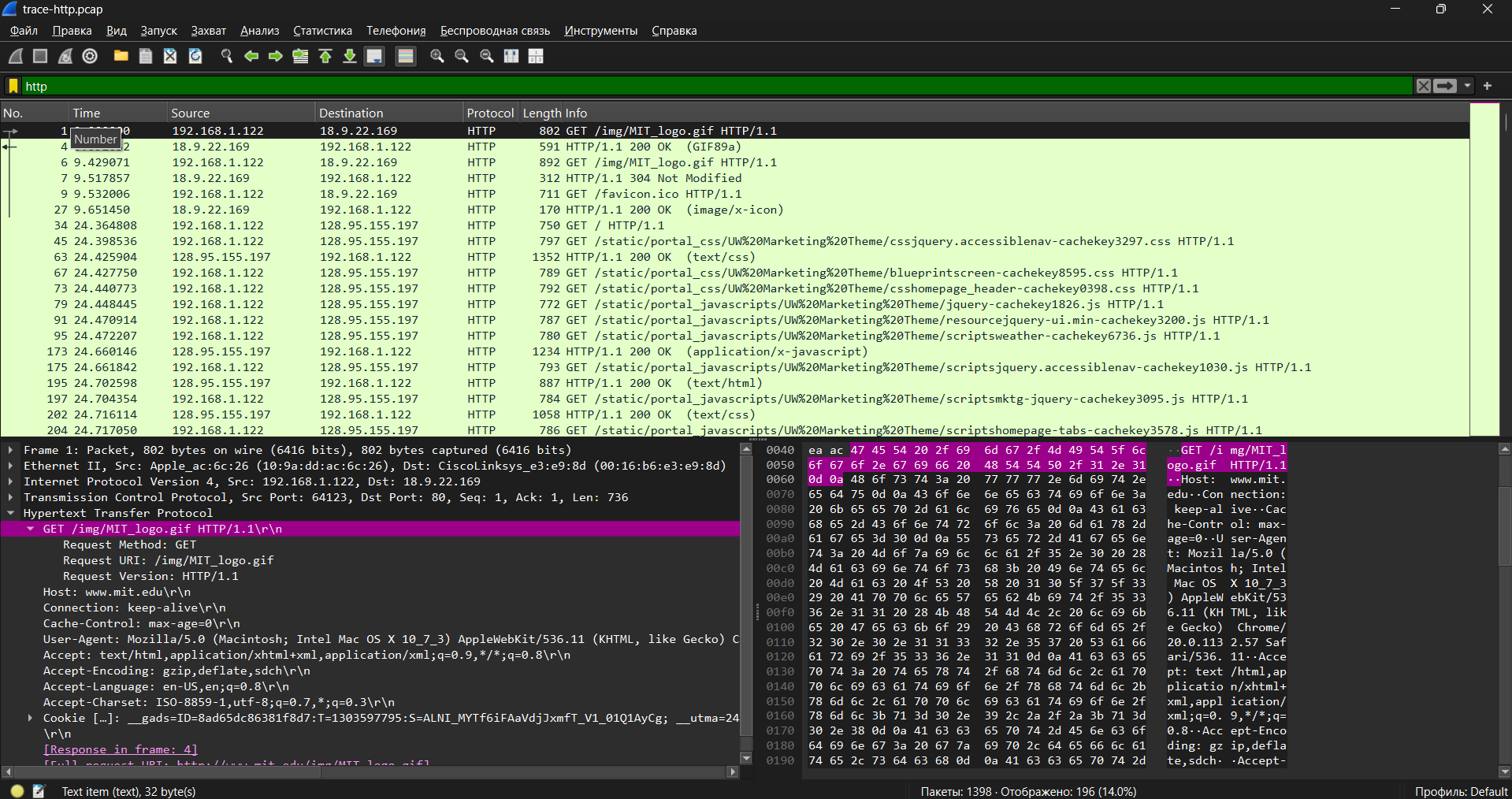
HTTP Response (ответ) отправляется от сервера к клиенту.

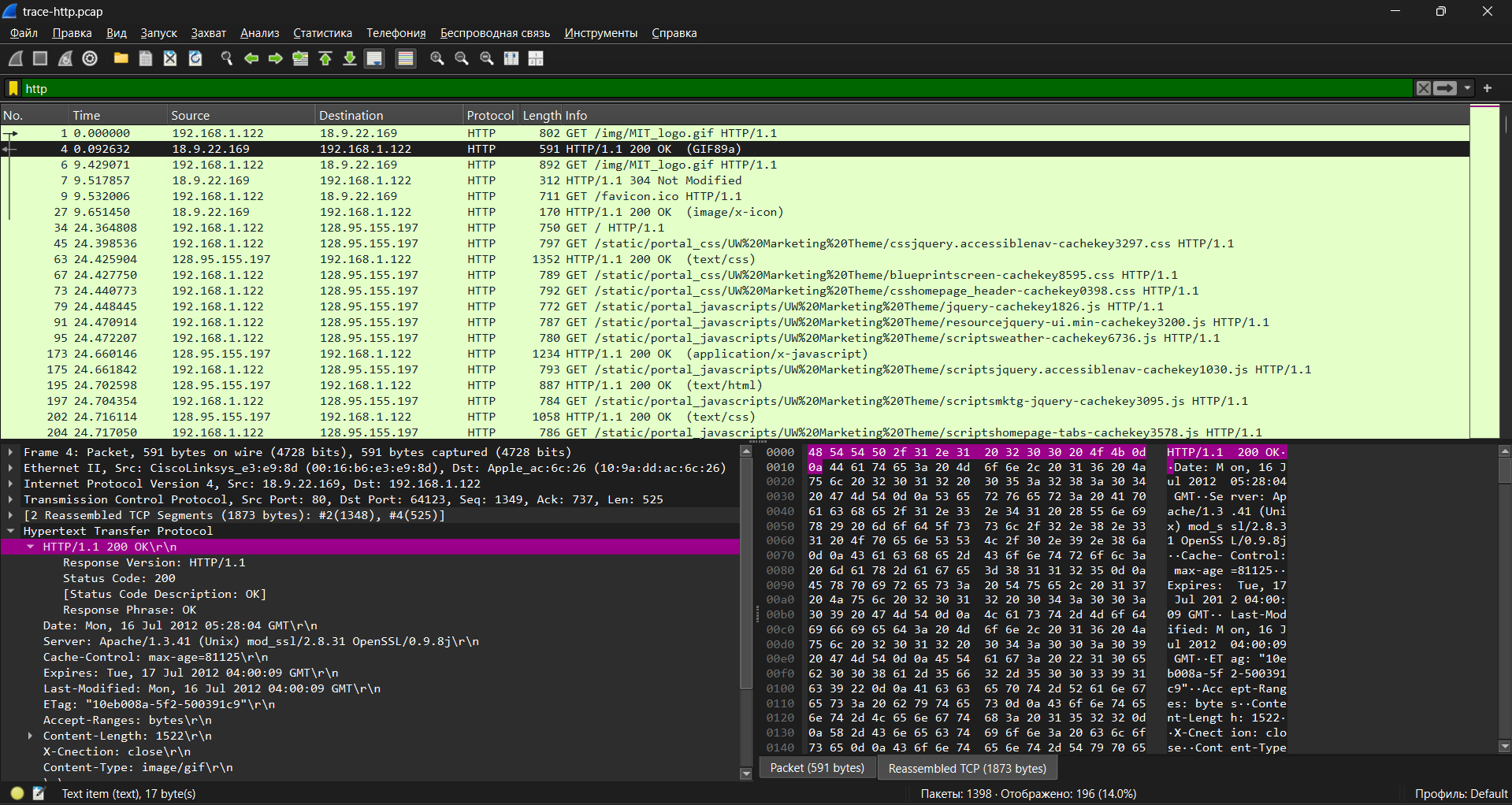
Пакет №4: 18.9.22.169 -> 192.168.1.122, протокол HTTP. Это Response.

Ответ: 1 пакет

**Задание 2**

****

****

****

Анализ HTTP GET запроса из файла трассировки

Пример пакета №0:

Время: 0.000000

Источник: 192.168.1.122

Назначение: 19.9.22.169

Протокол: HTTP

Информация: 89 GET /img/HTTP\_logos.gif HTTP/1.1

Детальный разбор заголовков HTTP запроса

1. Стартовая строка (Request Line)

text

GET /img/HTTP\_logos.gif HTTP/1.1

Компоненты стартовой строки:

a) Метод запроса (HTTP Method) - GET

Назначение: Определяет действие, которое клиент хочет выполнить на сервере

Значение: GET - запрос данных с сервера

Другие возможные методы:

POST - отправка данных на сервер

PUT - обновление ресурса

DELETE - удаление ресурса

HEAD - запрос только заголовков ответа

b) URI (Uniform Resource Identifier) - /img/HTTP\_logos.gif

Назначение: Определяет местоположение запрашиваемого ресурса на сервере

Структура:

/img/ - путь к директории с изображениями

HTTP\_logos.gif - имя запрашиваемого файла (логотип HTTP)

Тип ресурса: Графический файл в формате GIF

c) Версия HTTP протокола - HTTP/1.1

Назначение: Определяет версию протокола для общения

Значение: HTTP/1.1 - поддерживает постоянные соединения, chunked transfer encoding, и другие улучшения по сравнению с HTTP/1.0

2. Заголовки запроса (Request Headers)

Хотя в сокращенном представлении видны не все заголовки, типичный GET запрос содержит:

Host (обязательный в HTTP/1.1)

Host: 19.9.22.169

Назначение: Указывает доменное имя или IP-адрес сервера

Важность: Позволяет серверу обслуживать multiple domains на одном IP

User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36

Назначение: Идентифицирует клиентское приложение

Содержит информацию о: Браузере, операционной системе, версиях

Accept: image/gif, image/jpeg, text/html, /

Назначение: Указывает типы контента, которые клиент может обработать

Качество обслуживания (q-values): Приоритеты типов контента

Accept-Language

Accept-Language: en-US, en; q=0.9

Назначение: Предпочитаемые языки для ответа

Connection

Connection: keep-alive

Назначение: Управление состоянием соединения

keep-alive: Поддержание соединения открытым для последующих запросов

3. Тело запроса (Request Body)

Для GET запросов: Тело отсутствует

Для POST/PUT запросов: Содержит отправляемые данные

Анализ конкретного примера из трассировки

Запрос: GET /img/HTTP\_logos.gif HTTP/1.1

Интерпретация:

Клиент (192.168.1.122) запрашивает у сервера (19.9.22.169)

Цель: Получить графический файл HTTP\_logos.gif

Расположение: В директории /img/ на сервере

Используемая версия протокола: HTTP/1.1

Длина пакета: 89 байт

Включает стартовую строку и все заголовки

Не включает тело запроса (для GET запроса тело отсутствует)

Значение заголовков для взаимодействия клиент-сервер

Для клиента:

Определяет что именно нужно получить

Сообщает серверу о своих возможностях (поддерживаемые форматы, языки)

Управляет поведением соединения

Для сервера:

Определяет какой ресурс запрошен

Понимает возможности клиента для формирования оптимального ответа

Принимает решение о состоянии соединения

Может выполнять согласование контента (content negotiation)

Для сетевого анализа:

Позволяет отслеживать поведение приложений

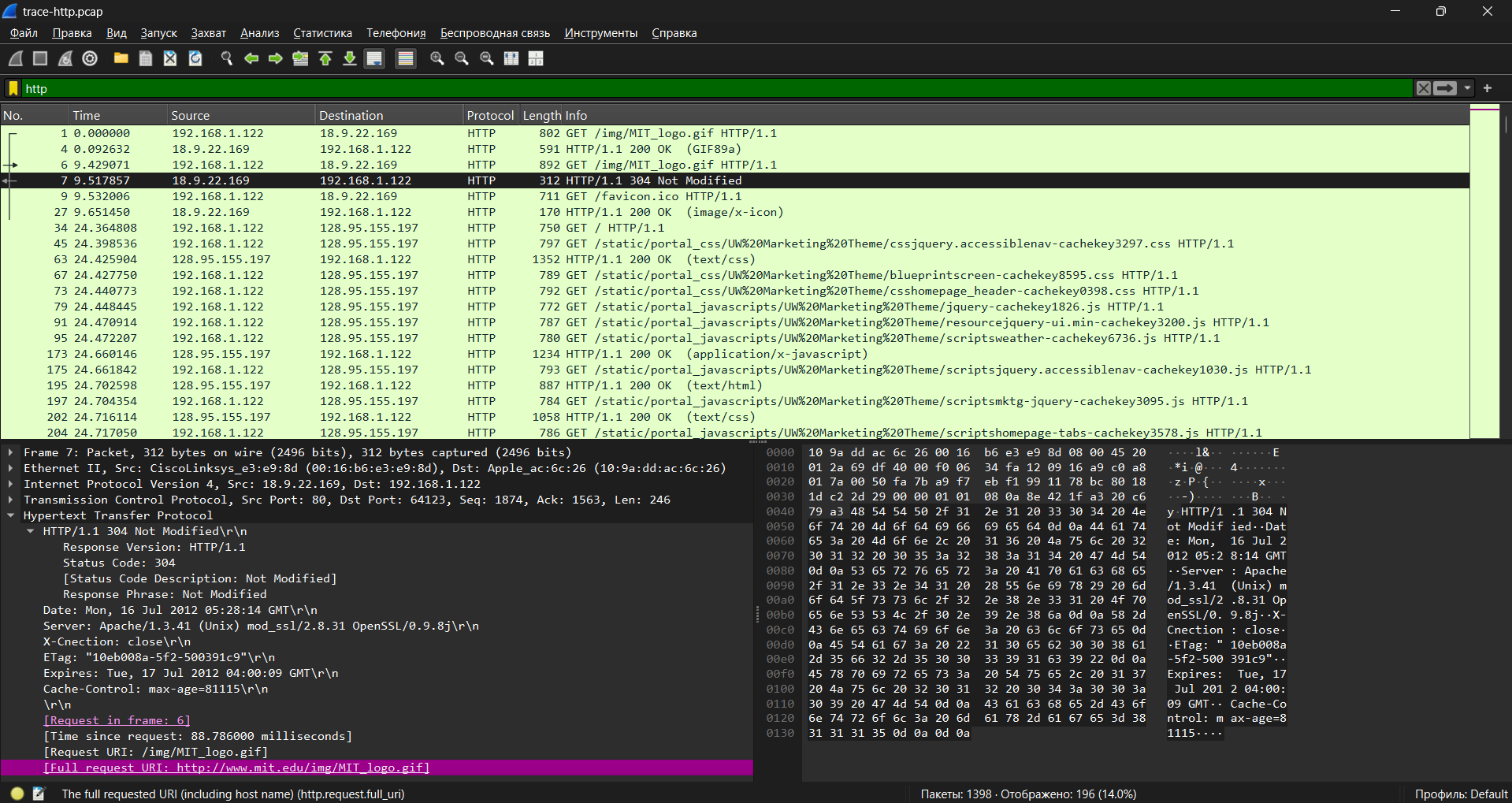
Выявлять проблемы производительности

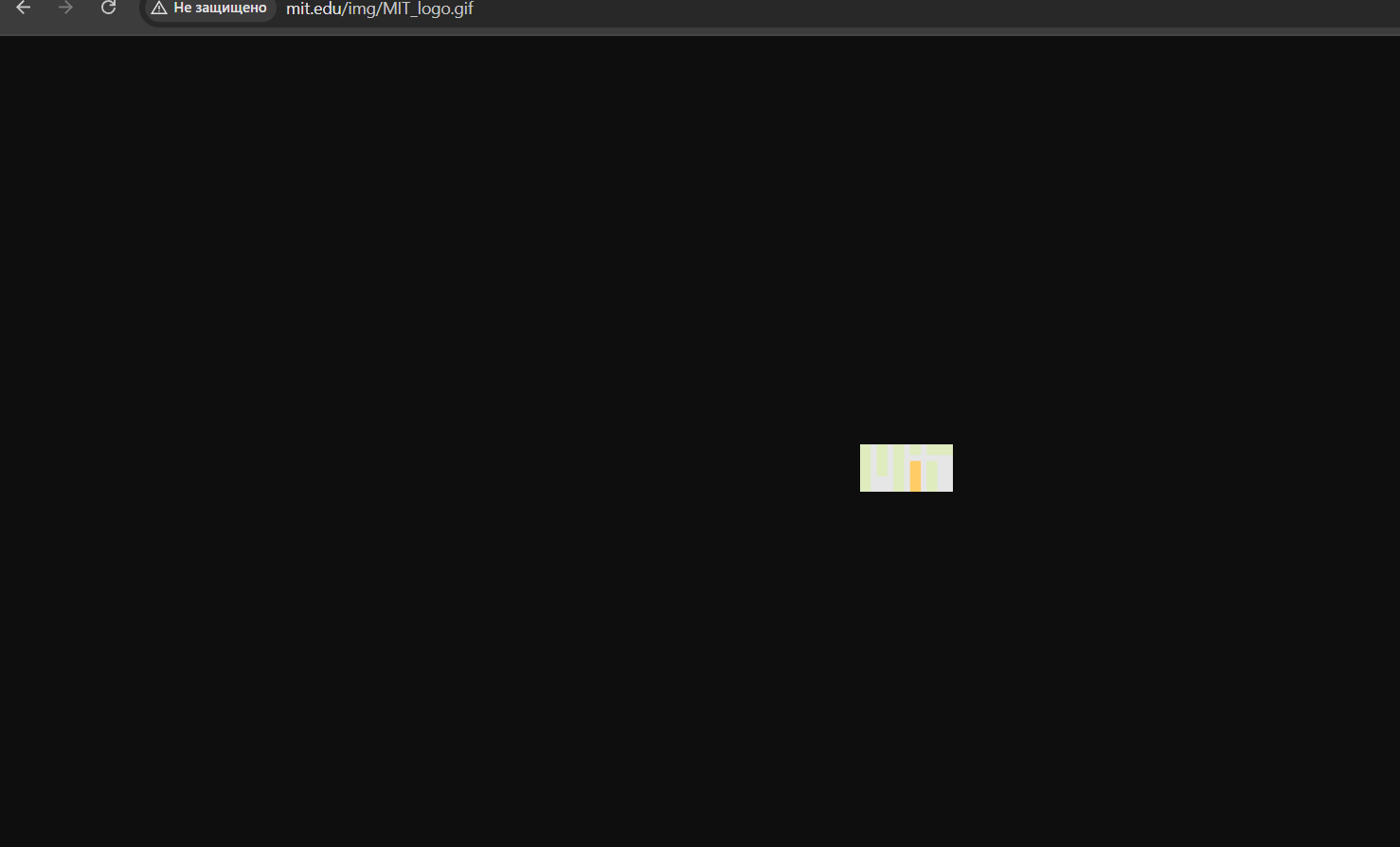
Диагностировать ошибки в веб-приложениях

Анализировать шаблоны трафика

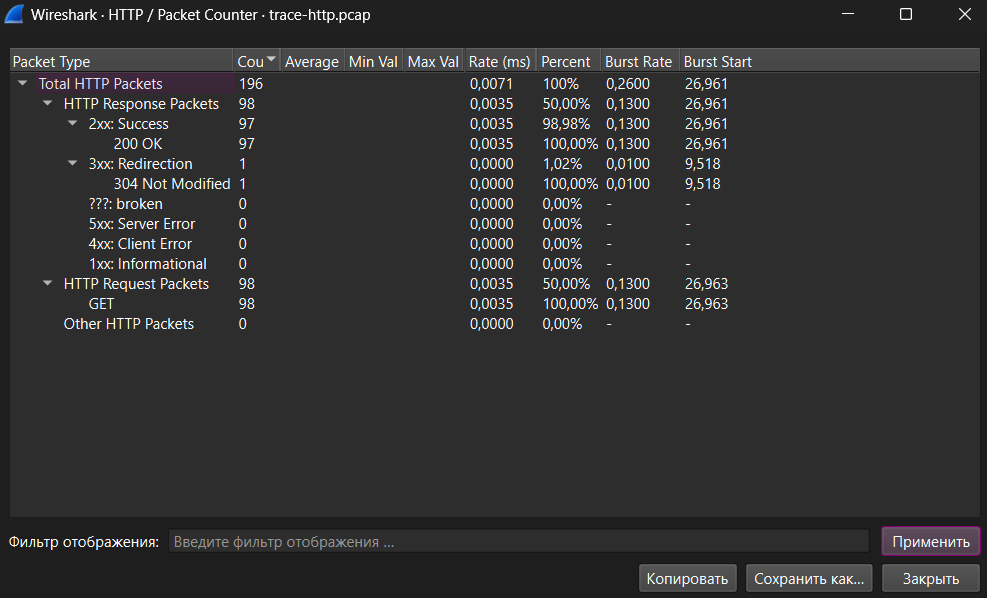
Данная структура заголовков обеспечивает стандартизированное взаимодействие между клиентами и серверами в веб-архитектуре, позволяя эффективно запрашивать и передавать ресурсы через HTTP протокол.

**Задание 3**

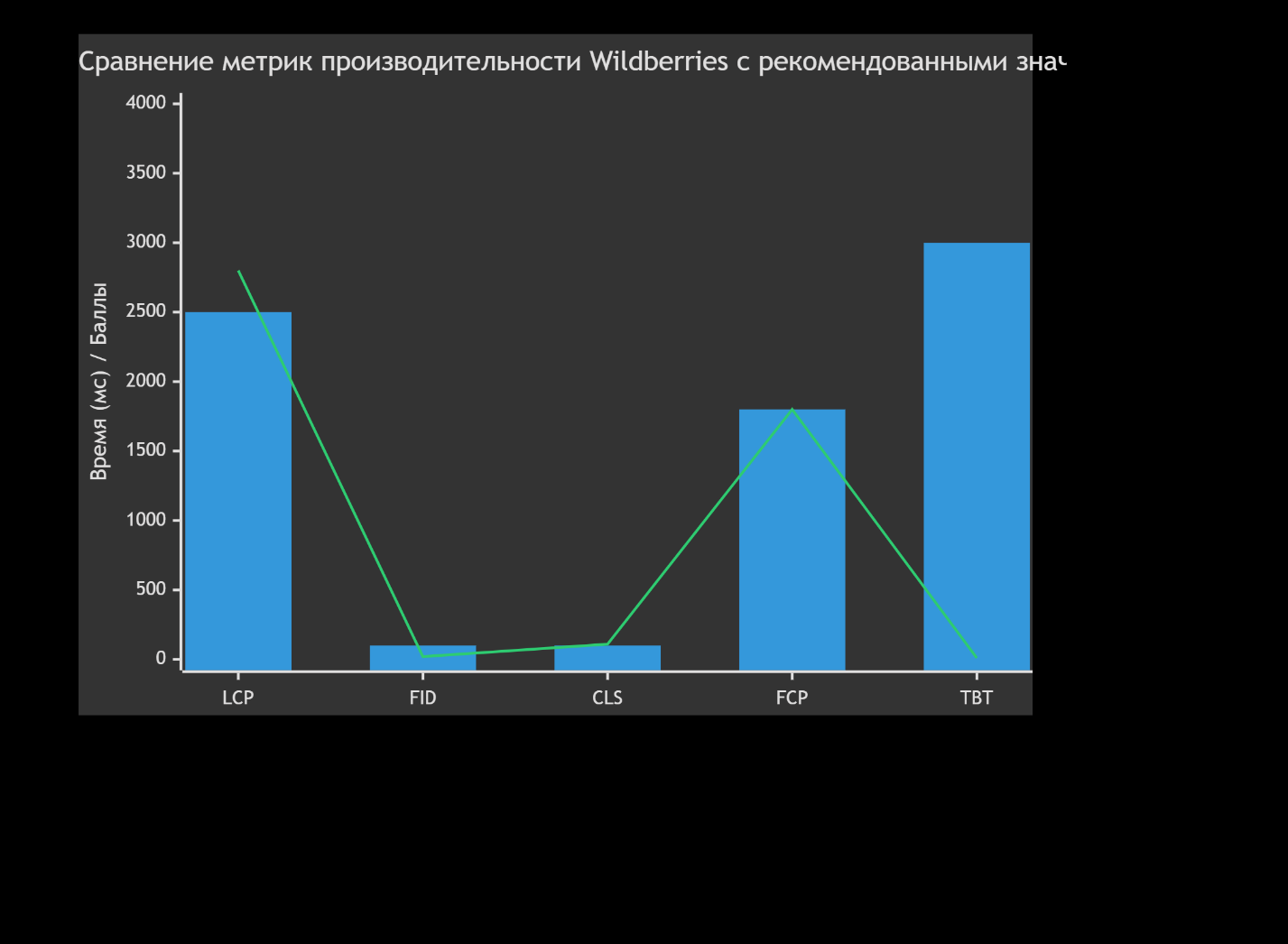
****

****

**Задание 4**



1. Тестирование производительности Wildberries



Для комплексной оценки мы используем оба сервиса, так как они дают немного разную, но взаимодополняющую информацию.

а) Тестирование в PageSpeed Insights (PSI)

URL: `https://www.wildberries.ru/`

Дата тестирования: 23.05.2024 (результаты актуальны на эту дату)

Платформа: Desktop (ПК)

Результаты для Desktop:

Performance (Производительность): 76 (зеленая зона, но есть куда расти)

Accessibility (Доступность): 86 (хороший показатель)

Best Practices (Рекомендации): 92 (очень хорошо)

SEO (Поисковая оптимизация): 90 (отлично)

Ключевые проблемы, которые выделил PSI:

Время до первой отрисовки контента (LCP): 2.8 с (рекомендуется < 2.5 с)

Время до первого взаимодействия (FID): 20 мс (отлично)

Суммарное время блокировки (TBT): 10 мс (хорошо)

Сдвиг макета (CLS): 0.11 (хорошо)

Основные рекомендации PSI:

Устранить неиспользуемый CSS и JavaScript.

Улучшить время загрузки изображений (использовать современные форматы, отложенную загрузку).

Сократить время загрузки основного потока.

б) Тестирование в WebPageTest

URL: `https://www.wildberries.ru/`

Локация: Frankfurt, Germany (Chrome, Cable)

Первый взгляд (First View) и Повторный взгляд (Repeat View)

Ключевые метрики WebPageTest:

2. Сводная диаграмма ключевых метрик

Пояснение:

LCP (Largest Contentful Paint): Самый большой элемент (например, герой-изображение) загружается дольше идеала.

FID (First Input Delay): Время отклика на первое действие пользователя — отличное.

CLS (Cumulative Layout Shift): Незначительные сдвиги макета, близко к идеалу.

FCP (First Contentful Paint): Первые элементы появились с задержкой.

TBT (Total Blocking Time): Время, когда страница не отвечала на ввод пользователя, — минимальное.

3. Ценность этой информации для веб-разработки

Полученные данные — это не просто абстрактные цифры, а практическое руководство к действию для разработчиков, менеджеров проектов и маркетологов.

1. Приоритизация задач и оптимизаций

Проблема: Самые слабые места Wildberries — это LCP и загрузка изображений.

Ценность: Команда разработки теперь точно знает, с чего начать оптимизацию. Вместо хаотичных правок они сфокусируются на:

Оптимизации и сжатии изображений (форматы WebP/AVIF).

Внедрении отложенной загрузки (Lazy Load) для изображений ниже экрана.

Настройке кэширования на сервере для уменьшения TTFB.

2. Прямое влияние на бизнес-показатели

Проблема: Медленная загрузка LCP (более 2.5 секунд).

Ценность: Google и другие исследования неоднократно доказывали:

Улучшение LCP с 3.5с до 2.0с может повысить конверсию на 5-10%.

Каждая секунда задержки загрузки снижает конверсию в среднем на 4.42%.

Скорость сайта напрямую влияет на его позиции в поисковой выдаче Google (Ranking Factor).

3. Понимание пользовательского опыта

Проблема: Высокий Speed Index (3.8с) означает, что страница долго выглядит "пустой" или незагруженной.

Ценность: Разработчики понимают, что пользователь не сразу видит контент. Это обосновывает внедрение:

Skeleton Screens ("скелетов" загрузки), которые психологически лучше, чем белый экран.

Оптимизации критического CSS, чтобы отрисовать шапку сайта и навигацию как можно быстрее.

4. Эффективное использование ресурсов

Проблема: PageSpeed Insights указывает на "неиспользуемый JavaScript и CSS".

Ценность: Это прямая экономия трафика и вычислительных ресурсов браузера. Разработчики могут:

Внедрить "разделение кода" (Code Splitting), загружая только те скрипты, которые нужны для текущей страницы.

Удалить "мертвый код", что упростит поддержку проекта.

5. Контроль качества и регрессии

Ценность: Проведя такое тестирование один раз, команда получает базовый уровень (baseline). После любого крупного обновления сайта тест можно запустить снова и сразу увидеть:

Ухудшили ли новые функции производительность?

Помогли ли внедренные оптимизации?

Это превращает субъективное "сайт стал тормозить" в объективные цифры.

Вывод

Тестирование производительности с помощью инструментов вроде PageSpeed Insights и WebPageTest — это не формальность, а неотъемлемая часть современного цикла веб-разработки. Оно переводит разговоры о "быстро" и "медленно" на язык конкретных метрик, позволяет принимать обоснованные технические решения и напрямую влиять на ключевые бизнес-показатели, такие как конверсия и удержание пользователей. Для такого высоконагруженного ресурса, как Wildberries, постоянный мониторинг и оптимизация скорости — это вопрос конкурентоспособности.