

Министерство цифрового развития, связи и массовых коммуникаций РФ
Уральский технический институт связи и информатики (филиал) ФГБОУ ВО
"Сибирский государственный университет телекоммуникаций и
информатики" в г. Екатеринбурге (УрТИСИ СибГУТИ)



Уральский технический
институт связи
и информатики

ОТЧЕТ
По дисциплине «Сетевое программирование»
Практическое занятие №7
««Основы маршрутизации. Клиент-серверная архитектура»

Выполнила: студентка группы ПЕ-216

Морос Е.Е.

Проверил: преподаватель

Бурумбаев Д.И.

1 Цель работы:

1.1. Закрепление знаний по теме «Клиент-серверная архитектура».

2 Подготовка к работе:

2.1. Изучить теоретический материал по теме «Основы маршрутизации. Клиент-серверная архитектура».

3 Задание:

3.1 Ответить письменно на вопросы тестового задания.

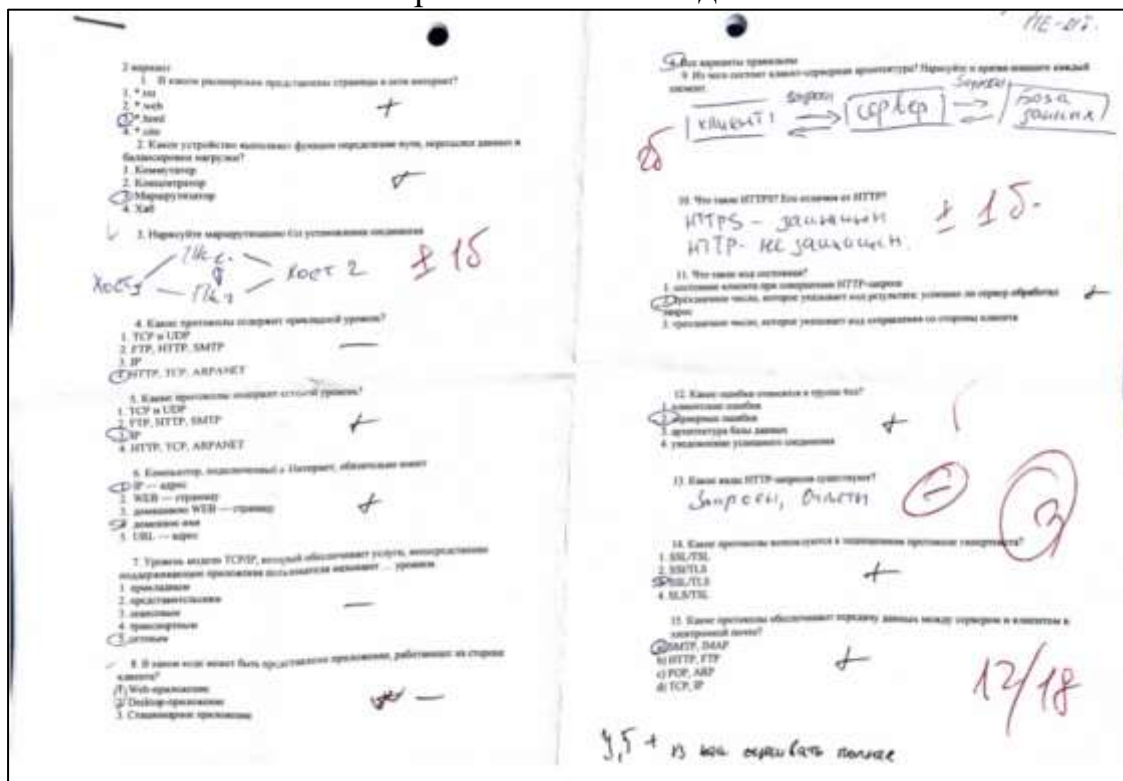


Рисунок 1 – Решение теста

4 Обобщенные вопросы тестового задания:

4.1 На каком уровне используется HTTP протокол?

HTTP (HyperText Transfer Protocol) работает на прикладном уровне модели OSI и TCP/IP. Это самый верхний уровень, обеспечивающий взаимодействие между приложениями пользователя.

4.2 Где обрабатывается информация (обрабатывается запрос) в архитектуре клиент-сервер?

В архитектуре "клиент-сервер" обработка запроса происходит на серверной стороне. Клиент формирует и направляет запрос, сервер получает его, обрабатывает (выполняет логику, обращается к БД и т.д.), формирует ответ и отправляет его клиенту.

4.3 На какой сервер требуется отправить запрос, чтобы узнать адрес нужного ресурса?

Для определения IP-адреса нужного ресурса используется DNS-сервер (Domain Name System). Клиент отправляет запрос на DNS-сервер, который возвращает соответствующий IP-адрес.

4.4 В какой топологии сети все хосты подключены последовательно (последний соединяется с первым)?

Такая топология называется кольцо (ring). В ней каждый узел соединён с двумя соседями, а последний узел замыкает круг, соединяясь с первым.

4.5 Какой протокол, использующийся на транспортном уровне, требует подтверждения доставки пакетов?

Протокол TCP (Transmission Control Protocol) требует подтверждения доставки пакетов, обеспечивая надёжную доставку данных между узлами.

4.6 Какие HTTP-коды информируют об ошибке на стороне сервера?

Коды состояния HTTP, начинающиеся с 5xx (от 500 до 599), информируют об ошибках на стороне сервера.

Основные коды:

- 500 Internal Server Error — внутренняя ошибка сервера
- 501 Not Implemented — сервер не поддерживает функционал запроса
- 502 Bad Gateway — неверный ответ от вышестоящего сервера
- 503 Service Unavailable — сервис недоступен
- 504 Gateway Timeout — превышено время ожидания ответа от сервера
- 505 HTTP Version Not Supported — версия протокола не поддерживается

4.7 Клиент-серверная архитектура: описание, виды пользовательских интерфейсов.

Клиент-серверная архитектура — это модель взаимодействия между двумя сторонами: клиентом (запрашивает услуги) и сервером (предоставляет ресурсы, данные, сервисы). Клиент инициирует коммуникацию, сервер её обрабатывает.

Виды пользовательских интерфейсов:

- Графический интерфейс (GUI) — окна, кнопки, меню (например, браузеры, почтовые клиенты).
- Командный интерфейс (CLI) — текстовый, через консоль (например, telnet, ssh).
- Веб-интерфейс — взаимодействие через веб-браузер.
- Мобильные интерфейсы — приложения для смартфонов/планшетов.

4.8 Клиент-серверная архитектура: назначение блоков, описание технических устройств клиентской и серверной части.

Назначение блоков:

- Клиент: инициирует запросы к серверу, отображает пользователю результат работы.
- Сервер: принимает запросы от клиентов, обрабатывает их, отправляет ответы, управляет ресурсами и БД.

Технические устройства:

- Клиентская часть:
- Компьютеры, ноутбуки, смартфоны, планшеты, рабочие станции с ПО-клиентом (браузер, почтовый клиент и т.п.).

Серверная часть:

- Специализированные серверы (физические или виртуальные), облачные сервисы, серверные кластеры. Обычно оснащены большими объёмами оперативной памяти, дискового пространства и мощными сетевыми картами.

4.9 Модель TCP/IP: назначение уровней, протоколы.

Модель TCP/IP состоит из 4 или 5 уровней (варианты описания):

Прикладной уровень (Application):

- Протоколы: HTTP, FTP, SMTP, DNS, POP3 и др.

Транспортный уровень (Transport):

- Протоколы: TCP, UDP.

Сетевой уровень (Internet):

- Протокол: IP (IPv4, IPv6), ICMP, ARP.

Канальный уровень (Network Access, Link):

- Протоколы: Ethernet, Wi-Fi, PPP, Frame Relay и др.

(иногда выделяют физический уровень отдельно)

Назначение: координатора передачи данных между компьютерами, обеспечение адресации, доставки и маршрутизации пакетов, управление соединениями, преобразование данных для передачи.

4.10 Маршрутизация: назначение, классификация, функции.

Маршрутизация обеспечивает выбор оптимального пути передачи данных между отправителем и получателем в компьютерных сетях.

Классификация:

- Динамическая — маршруты пересчитываются автоматически с учётом изменений.
- Статическая — маршруты настраиваются вручную и не изменяются автоматически.

Функции:

- Выбор оптимального маршрута для передачи пакетов
- Предотвращение петель маршрутизации
- Управление таблицами маршрутов
- Перенаправление трафика при отказе узла/канала

4.11 Нарисуйте маршрутизацию с установлением соединения

```
[Клиент] --(Запрос на соединение: SYN)--> [Маршрутизатор] --(SYN)--> [Сервер]
[Сервер] --(Ответ: SYN+ACK)--> [Маршрутизатор] --(SYN+ACK)--> [Клиент]
[Клиент] --(Подтверждение: ACK)--> [Маршрутизатор] --(ACK)--> [Сервер]
Данные могут передаваться между клиентом и сервером через маршрутизатор.
```

4.12 Какие группы кодов состояния бывают?

- HTTP-коды состояния группируются по первой цифре:
- 1xx — Информационные (Informational) — приняли, продолжаем работать (пример: 100 Continue)
- 2xx — Успешные (Success) — успешно обработано (пример: 200 OK)
- 3xx — Перенаправление (Redirection) — дальнейшие действия (пример: 301 Moved Permanently)
- 4xx — Ошибки клиента (Client Error) — ошибка в запросе клиента (пример: 404 Not Found)
- 5xx — Ошибки сервера (Server Error) — ошибка на сервере (пример: 500 Internal Server Error)