

СиТ. Лаба 3.

Команды протокола HTTP. (отличие PATCH и PUT). Коды ответов HTTP (200, 304, 302, 401, 403, 402). Заголовки range, referer, host, expires, vary. Команды протокола FTP. Устройство NAT. Как NAT меняет пакет и сегмент. Привести пример для подмены адресов устройством NAT для различных сетей (192.168.0.0-192.168.255.255, 172.16.0.0-172.32.255.255, 10.0.0.0, 100.0.0.0, какие префиксы у данных сетей), сколько компьютеров может поддерживать NAT с узлами открывшими сколько-то соединений. Проблемы клиента за устройством NAT в активном режиме FTP. Зачем нужен активный и пассивный режим. Почему реализован переход с HTTP1 на HTTP2 и далее на HTTP3. Проблема HOL для HTTP и TCP. В чем проблема текстового протокола HTTP1 и недостатки использования base64. Есть ли у FTP проблема с base64.

<https://helpmestudyph.notion.site/3-64db570f122f4b98bb0736c823891e8c>

HTTP (англ. HyperText Transfer Protocol — «протокол передачи гипертекста») — протокол прикладного уровня передачи данных, изначально — в виде гипертекстовых документов в формате HTML, в настоящее время используется для передачи произвольных данных.

Команды протокола HTTP.

GET — Используется для запроса содержимого указанного ресурса.

HEAD — Аналогичен методу GET, за исключением того, что в ответе сервера отсутствует тело. Запрос HEAD обычно применяется для извлечения метаданных, проверки наличия ресурса (валидация URL) и чтобы узнать, не изменился ли он с момента последнего обращения.

Заголовки ответа могут кэшироваться. При несовпадении метаданных ресурса с соответствующей информацией в кэше — копия ресурса помечается как устаревшая.

POST — Применяется для передачи пользовательских данных заданному ресурсу. Например, в блогах посетители обычно могут вводить свои комментарии к записям

в HTML-форму, после чего они передаются серверу методом POST и он помещает их на страницу. При этом передаваемые данные (в примере с блогами — текст комментария) включаются в тело запроса. Аналогично с помощью метода POST обычно загружаются файлы на сервер.

PUT — Применяется для загрузки содержимого запроса на указанный в запросе URI. Если по заданному URI не существует ресурса, то сервер создаёт его и возвращает статус 201 (Created). Если же ресурс был изменён, то сервер возвращает 200 (Ok) или 204 (No Content).

PATCH — Аналогично PUT, но применяется только к фрагменту ресурса.

DELETE — Удаляет указанный ресурс.

TRACE — Возвращает полученный запрос так, что клиент может увидеть, какую информацию промежуточные серверы добавляют или изменяют в запросе.

CONNECT — Преобразует соединение запроса в прозрачный TCP/IP-туннель, обычно чтобы содействовать установлению защищённого SSL-соединения через нешифрованный прокси.

Коды ответов HTTP (200, 304, 302, 401, 403, 402)

200 OK — успешный запрос. Если клиентом были запрошены какие-либо данные, то они находятся в заголовке и/или теле сообщения. Появился в HTTP/1.0.

304 Not Modified — сервер возвращает такой код, если клиент запросил документ методом GET, использовал заголовок If-Modified-Since или If-None-Match и документ не изменился с указанного момента. При этом сообщение сервера не должно содержать тела. Появился в HTTP/1.0.

302 Found, 302 Moved Temporarily — запрошенный документ временно доступен по другому URI, указанному в заголовке в поле Location.

401 Unauthorized — для доступа к запрашиваемому ресурсу требуется аутентификация. В заголовке ответ должен содержать поле WWW-Authenticate с перечнем условий аутентификации. Иными словами, для доступа к запрашиваемому ресурсу клиент должен представиться, послав запрос, включив при этом в заголовок сообщения поле Authorization с требуемыми для

аутентификации данными. Если запрос уже включает данные для авторизации, ответ 401 означает, что в авторизации с ними отказано.

402 Payment Required — это нестандартная ошибка клиента, зарезервированная для использования в будущем. Иногда этот код означает, что запрос не может быть выполнен до тех пор, пока клиент не совершит оплату. Изначально создан для активации цифровых средств или (микро) платёжных систем и изображает, что запрошенный контент недоступен пока клиент не совершит оплату. Так или иначе, стандартизованного использования для кода нет, и он может использоваться разными элементами в разном контексте.

403 Forbidden — сервер понял запрос, но он отказывается его выполнять из-за ограничений в доступе для клиента к указанному ресурсу. Иными словами, клиент не уполномочен совершать операции с запрошенным ресурсом.

Заголовки **range**, **referer**, **host**, **expires**, **vary**

Заголовок запроса **Range** указывает серверу какую часть документа ему необходимо вернуть. Несколько частей документа может быть запрошено с помощью заголовка Range за один раз, и сервер может вернуть все эти части через многокомпонентный документ. При отправке данных отдельными частями, сервер использует код ответа 206 Partial Content. Если запрашиваемые диапазоны данных не верны, сервер возвращает ошибку 416 Range Not Satisfiable. Сервер так же может проигнорировать заголовок Range и вернуть документ целиком с кодом ответа 200.

```
Range: <единица>=<начало-диапазона>-  
Range: <единица>=<начало-диапазона>-<конец-диапазона>  
Range: <единица>=<начало-диапазона>-<конец-диапазона>, <начало-диапазона>-<конец-диапа  
зона>  
Range: <единица>=<начало-диапазона>-<конец-диапазона>, <начало-диапазона>-<конец-диапа  
зона>, <начало-диапазона>-<конец-диапазона>  
Range: <единица>=-<длина-с-конца>
```

Range: bytes=200-1000, 2000-6576, 19000-

Заголовок запроса **Referer** содержит URL исходной страницы, с которой был осуществлён переход на текущую страницу. Заголовок Referer позволяет серверу узнать откуда был осуществлён переход на запрашиваемую страницу. Сервер

может анализировать эти данные, записывать их в логи или оптимизировать процесс кеширования.

Заголовок **Host** содержит имя домена, для которого предназначен запрос и, опционально, номер порта.

Если порт не указан, то используется порт по умолчанию протокола/сервиса (например «80» для HTTP, "443" для HTTPS и т.д.).

Заголовок **Expires** содержит дату/время, по истечении которой ответ сервера считается устаревшим.

Прошедшая или невалидная дата, например 0, обозначает, что ресурс уже устарел.

Vary. Заголовок определяет, как сопоставить будущие заголовки запроса, чтобы решить, можно ли использовать кешированный ответ, а не запрашивать новый с исходного сервера.

Команды протокола FTP

File Transfer Protocol, или протокол передачи файлов, — это протокол, относящийся к прикладному уровню и отвечающий за передачу данных между двумя системами. Он работает поверх протокола *TCP*. При передаче файлов FTP использует одновременно два *TCP*-канала: один из них отвечает за управление передачей данных, а второй — передает их.

ABOR — Прервать передачу файла.

CDUP — Сменить каталог на вышестоящий.

CWD — Сменить каталог.

DELE — Удалить файл (DELE filename).

EPSV — Войти в расширенный пассивный режим. Применяется вместо PASV.

HELP — Выводит список команд, принимаемых сервером.

LIST — Возвращает список файлов каталога. Список передаётся через соединение данных.

MDTM — Возвращает время модификации файла.

MKD — Создать каталог.

NLIST — Возвращает список файлов каталога в более кратком формате, чем LIST. Список передаётся через соединение данных.

NOOP — Пустая операция.

PASS — Пароль.

PASV — Войти в пассивный режим. Сервер вернёт адрес и порт, к которому нужно подключиться, чтобы забрать данные. Передача начнётся при введении следующих команд: RETR, LIST и т. д.

PORT — Войти в активный режим. Например PORT 12,34,45,56,78,89. В отличие от пассивного режима для передачи данных сервер сам подключается к клиенту.

PWD — Возвращает текущий каталог.

QUIT — Отключиться.

REIN — Реинициализировать подключение.

RETR — Скачать файл. Перед RETR должна быть команда PASV или PORT.

RMD — Удалить каталог.

RNFR и **RNT0** — Переименовать файл. RNFR — что переименовывать, RNT0 — во что.

SIZE — Возвращает размер файла.

STOR — Закачать файл. Перед STOR должна быть команда PASV или PORT.

SYST — Возвращает тип системы (UNIX, WIN, ...).

TYPE — Установить тип передачи файла (бинарный, текстовый).

USER — Имя пользователя для входа на сервер.

Устройство NAT. Как NAT меняет пакет и сегмент. Привести пример для подмены адресов устройством NAT для различных сетей (192.168.0.0-192.168.255.255, 172.16.0.0-172.32.255.255, 10.0.0.0, 100.0.0.0, какие префиксы у данных сетей), сколько компьютеров может поддерживать NAT с узлами открывшими сколько-то соединений. Проблемы клиента за устройством NAT в активном режиме FTP.

NAT

NAT — это специальный механизм, реализованный в сетях TCP/IP, который позволяет изменять IP-адреса пересылаемых пакетов, т.е. тех внутренних IP, которые присылаются на сетевой шлюз — в глобальные для дальнейшей отправки во внешний интернет. Также, такие пакеты называют транзитными.

FTP может работать в активном или пассивном режиме, от выбора которого зависит способ установки соединения. В активном режиме клиент создаёт управляющее TCP-соединение с сервером и отправляет серверу свой IP-адрес и произвольный номер клиентского порта, после чего ждёт, пока сервер запустит TCP-соединение с этим адресом и номером порта. В случае, если клиент находится за брандмауэром и не может принять входящее TCP-соединение, может быть использован пассивный режим. В этом режиме клиент использует поток управления, чтобы послать серверу команду PASV, и затем получает от сервера его IP-адрес и номер порта, которые затем используются клиентом для открытия потока данных с произвольного клиентского порта к полученному адресу и порту. Оба режима были обновлены в сентябре 1998 г. для поддержки IPv6. В это время были проведены дальнейшие изменения пассивного режима, обновившие его до расширенного пассивного режима.

HTTP / HTTP2.0 / HTTP3.0

HTTP — протокол прикладного уровня передачи данных, изначально — в виде гипертекстовых документов в формате HTML, в настоящее время используется для передачи произвольных данных.

В HTTP/2 сервер имеет право послать то содержимое, которое ещё не было запрошено клиентом. Это позволит серверу сразу выслать дополнительные файлы, которые потребуются браузеру для отображения страниц, без необходимости анализа браузером основной страницы и запрашивания необходимых дополнений.

Среди ключевых особенностей: мультиплексирование запросов, расстановка приоритетов для запросов, сжатие заголовков, загрузка нескольких элементов параллельно посредством одного TCP-соединения.

HTTP/3 — предлагаемый последователь HTTP/2, который уже используется в Веб на основе UDP вместо TCP в качестве транспортного протокола.