04.09

Лекция 1. От полностью централизованных систем к вычислительным сетям

Системы пакетной обработки

В 50-х годах 20-го века вычислительные центры строились на базе больших громоздких и дорогих компьютеров универсального назначения, т. е. на базе ***мейнфреймов***. Они не были предназначены для интерактивной работы пользователей и работали в режиме пакетной обработки.

Основным носителем информации были ***перфокарты***, которые передавались операторам мейнфрейма, а те в свою очередь загружали в мейнфрейм, примерно, для однодневной обработки.

Плюсы:

- Самый эффективный режим использования вычислительной мощности

  > *Пояснение:*
  > Самая дорогая и мощная компонента мейнфрейма, а именно центральный процессор (ЦП) использовался постоянно (>90% времени)

Минусы:

- Большая цена ошибки, которая выражалась в потерянном рабочем дне, как самого мейнфрейма, так и обслуживающего персонала; в большом потреблении электроэнергии и т.п.

  > *Основная причина:* Неправильная перфокарта (Перепутан 0 и 1)

- Ненадёжный носитель информации в виде перфокарты
- Не было интерактивного режима

  > Пользователи не обращались напрямую, а через операторов

- Все информационные ресурсы были централизованы

Информационные ресурсы:

- Файлы любого формата
- База данных

- Периферийные устройства
- Вычислительная мощность (Оперативная память, внешняя память, процессор)
- Доступы (К другой сети, к внешней сети, к БД и т.д)
- Сайты, порталы, WEB-страницы

06.09

Практика 1.

Многотерминальные системы

По мере удешевлении процессоров в начале 60-х годов появились новые способы организации вычислительного процесса.

Появились ***интерактивные многотерминальные системы разделения времени***.

В таких системах мейнфрейм отдавался в распоряжение сразу нескольким пользователям посредством терминала. Реакция на действие пользователя не занимала много времени.
Таким образом, у пользователей складывалась иллюзия единоличного владения мейнфреймом через терминал. Таким образом, стоит сказать о *псевдопараллельной обработки* многотерминальной системы.

- **Централизованным оставалась вычислительная мощность, зато функция ввода/вывода стали распределённой.**

В этот период было актуально имперический *закон Гроша*, который гласил, что производительность компьютера была прямо пропорциональна квадрату его стоимости.

> *Пример:* Один компьютер за 1 миллион был лучше двух компьютеров за 500 тыс.

Многотерминальные системы стали первым шагом к созданию локальных сетей.

Глобальные сети

Сначала была решена задача попроще, а именно соединение мейнфрейма и терминала, удалённого на несколько сотен километров. Это задача решалась с помощью телефонных линий связи и модема.

> Модем: перевод аналогового сигнала в цифровой и обратно - **модуляция**

Чуть позже по той же схеме соединили два мейнфрейма между собой.

Компьютеры получили возможность обмениваться данными в автоматическом режиме, что является базовым механизмом в любой вычислительной сети.

В первых глобальных сетях были созданы три классические сетевые службы, которые используются и по сей день:

- Службы электронной почты
- Службы синхронизации БД
- Службы файлового обмена

Кроме того в первых глобальных сетях были отработаны такие концепции:

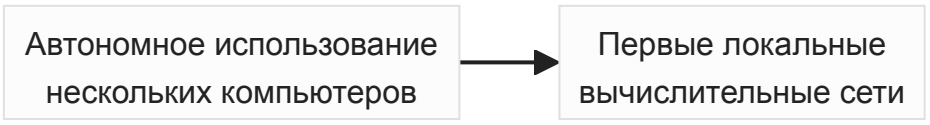
- Многоуровневые построения коммуникационных протоколов

  > *Пример:* TCP/IP

- Коммутация пакетов
- Маршрутизация пакетов в составных сетях

Таким образом, исторически первыми появились глобальные сети, а не локальные.

Локальные сети



В начале 70-х годов произошёл технологический прорыв в области производства компьютерных компонентов. **Появились большие интегральные схемы**. Их сравнительно невысокая стоимость и высокие функциональные возможности привели к созданию мини компьютеров, которые стали конкурентами мейнфреймам, закон Гроша перестал соответствовать действительности. Тем не менее первое время мини-компьютеры продолжали работать автономно. Это происходило по двум причинам: аппаратная и программная несовместимость. Однако в некоторых организациях мини-ЭВМ разных производителей связывали между собой нестандартными устройствами сопряжения. Таким образом, в то время создание сетей было творческой задачей или даже искусством.

В середине 80-х годов положение дел изменилось, поскольку появились стандартные сетевые технологии:

- Ethernet
- Token ring
- Arcnet

Мощным стимулом для появления этих технологий стало создание ***персонального компьютера(ПК)***, они стали идеальными элементами создания сетей, так как с одной стороны они являются достаточно мощными для работы сетевого ПО, а с другой

стороны явно нуждаются в объединении вычислительной мощности для решения сложных задач, а также для разделения дорогих сетевых устройств.

Таким образом, создание сети перестало быть искусством, а стало быть рутиной работой. Теперь для создания сетей было необходимо:

1. Приобрести стандартные сетевые адаптера. Например, Ethernet.
2. Приобрести стандартный кабель и соединить его с сетевыми адаптерами стандартными разъёмами
3. Установить на компьютер одну из популярных сетевых ОС (В то время: Netware от Novel).

Практика 2

Современные тенденции

Анг	Рус	Заметка
Роутер(router)	Маршрутизатор	Объединяет устройства в сети
Switch	Коммутатор	Перенаправляет пакеты данных
Репитор	Повторитель	Продлевает сигнал
Хаб(hub)	Концентратор	Простое соединение устройств
Bridge	Мосты	Соединяет разные сети \

DHCP - раздаёт IP-адреса
DNS - переводит символьный адрес в цифровой
LAN - локальные
WAN - глобальные

1.

В своём развитии глобальные сети догоняют локальные, а именно скорости в глобальных сетях становятся сопоставимыми со скоростями. Таким образом, в WAN создаются и разворачиваются сетевые службы и сервисы, которые по удобству и прозрачности сопоставимы со службами в LAN.

2.

Раннее в сетях использовалось не интеллектуальное оборудование: повторители мосты, концентраторы. На данный момент используется устройства: коммутаторы и маршрутизаторы(свичи и роутеры), которые поддерживают разнообразные сетевые

протоколы и являются специализированными мощными мультипроцессорами, которые нужно настраивать, оптимизировать и администрировать.

3.

Мейнфреймы получили вторую жизнь в виде ***серверов***, однако перестали быть универсальными, и теперь их можно классифицировать по назначению:

- ***Файловые сервера***, который в свою очередь можно разделить на сервера БД, облачные, игровые, видеохостинг и зеркальные
- ***Принт-сервера***
- ***Вычислительный сервер*** (для майнинга)
- ***Сервер унифицированных коммуникаций*** (ip-телефония, электронная почта, видео конференц связь и мессенджеры)
- ***Сервер информационной безопасности*** (СКУД, антивирус, VPN, firewall)
- ***Коммуникационный сервер***

4.

В начале передавалась только текстовая информация, по мере развития сетей в них стала передаваться мультимедийная информация, а также другие виды трафика, чувствительные к задержке. В современных сетях передаётся такая информация:

- трафик реального времени
- биометрия
- телеметрия
- ЭТМ, связанная с транзакциямим

Лекция 2

Модель комплекса программно-аппаратных сетей (одна из первых)

1. Аппаратный слой (мейнфреймы, мини-ЭВМ, ПК, ноуты, смартфоны...)
2. Коммуникационное оборудование
3. Операционные системы
4. Сетевые приложения
   На первых этапах развития сетей наибольшее значение и стоимость имел 1 слой, а именно аппаратный, однако на сегодняшний день 2 слой выходит на первый план по важности и стоимости.

Что даёт предприятию использование сетей?

Плюсы:

- Повышение эффективности работы, которое может выражаться, например, в увеличении предприятия (Например, уменьшение количества принтеров)
- Более высокая отказоустойчивость(Дублирование узлов сети или технологий | Например, дублирование сервера, чтобы распределять запросы) информационных систем и аппаратного обеспечения предприятия
- Автоматизация технологических процессов (СЭД, Система контроля доступа, Интернет вещей)
- Возможность совместного использования данных и устройств (БД, СЭД)
- Оперативный доступ к обширной корпоративной информации
- Совершенствование коммуникации (Intranet, Корпоративные мессенджеры (Microsoft Teams,))
  СЭД — система электронного документооборота.
  Intranet - внутренний интернет, некий сайт
  Минусы:
- Сложности, связанные с ПО
- Обеспечение надёжности и производительности при транспортировке сообщений
- Вопросы, связанные с обеспечением безопасности (Аттаки, вирусы)

Практика 3

Компьютерная сеть и процесс передачи информации

Компьютерная сеть - система, которая служит чтобы обеспечить обмен данными между вычислительными устройствами и/или программным обеспечением.

- Проводная: Витая пара, оптоволокно, коаксиальный кабель
- Беспроводная: Wi-Fi, Bluetooth, ИК
  Сетевой элемент <=> NE(Network Element) <=> Узел <=> Хост
  Контрольная сумма - необходимо, чтобы проверить, что данные неискажены
  TTL - количество прыжков (хоп-ов) по маршутизаторам

Процесс передачи данных

Сравним доставку реальных объектов с сетевой связью. Реальные объекты упаковываются в посылке, к которым прилагаются документы содержащие служебную информацию: квитанция, опись.

> По аналогии сетевое приложение, сгенерировав данные разбивает их на фрагменты и упаковывает каждый фрагмент в новую структуру данных, добавляя заголовок и концевик, такой процесс называется ***инкапсуляция***. В заголовке: ip получателя и отправителя, длина пакеты, контрольная сумма, TTL (время жизни пакеты, измеряемое в хопах).

Далее посылки переходят на распределительный центр, где они сортируются оп адресам назначения.

> По аналогии данные приходят на маршрутизатор, где они ***декапсулируются*** и маршрутизатор ищет соответствие ip адреса получателя со своей таблицей маршрутов. Если соответствие найдено, то **шлюз** вновь инкапсулирует сетевой пакет и отправляет его по выходному порту, который нашёл в своей таблице. Если соответствие не найдено, то есть 3 варианта:

1. Пакет удаляется
2. Пакет удаляется с уведомлением источника данных
3. Данные инкапсулируются в пакеты и отправляются по порту по умолчанию (настраиваемый выходной порт специально для таких целей) \

> > Далее пакет может пройти несколько транзитных маршрутизаторов. ***Маршрутизатор-шлюз*** отличается от ***транзитного*** тем, что стоит на границе локальной сети, тогда как транзитный маршрутизатор стоит на пути следования пакета от одной сети к другой.

Далее посылки из распределительного центра с помощью различных вариантов доставляются в распределительный центр города назначения, откуда их привозят на почту, либо курьером получателю. Получатель проверяет целостность посылки и принимает её, либо возвращает.

> По аналогии данные со шлюза с помощью разных каналов передачи данных через транзитные маршрутизаторы достигает шлюза сети назначения. Этот шлюз декапсулирует сетевой пакет, сверяется со своей таблицей маршрутизации, вновь инкапсулирует и отправляет хосту получателя. Получатель декапсулирует сетевой пакет, проверяет контрольную сумму. Если контрольная сумма корректна, то оставляет эти данные. Если же данные искажены, то уведомляет источника информации. \

Сети могут состоять из трёх уровней:

- Нижний - уровень доступа
- Чуть выше - уровень агрегации (распределения)
- Уровень ядра

Лекция 3

Типы сетей и топологий

Сети могут делиться в зависимости от классификационного признака \

По охвату

- LAN -локальные
- MAN - городские
- WAN (Wide Area Network) - глобальные
  Person Area Network (PAN) - персональная сеть, объединение устройств одного пользователя
  BAN Body Area Network - нательная сеть
  MBAN - медицинская нательная сеть
  WMBAN (Wireless medical BAN) - беспроводная медицинская нательная сеть
  Nano Network - применяется в биомедицине и в военных технологиях
  NFC (Near Field Communication) - Расстояние: 4 см, 4 Кб/с \

По функциональному назначению:

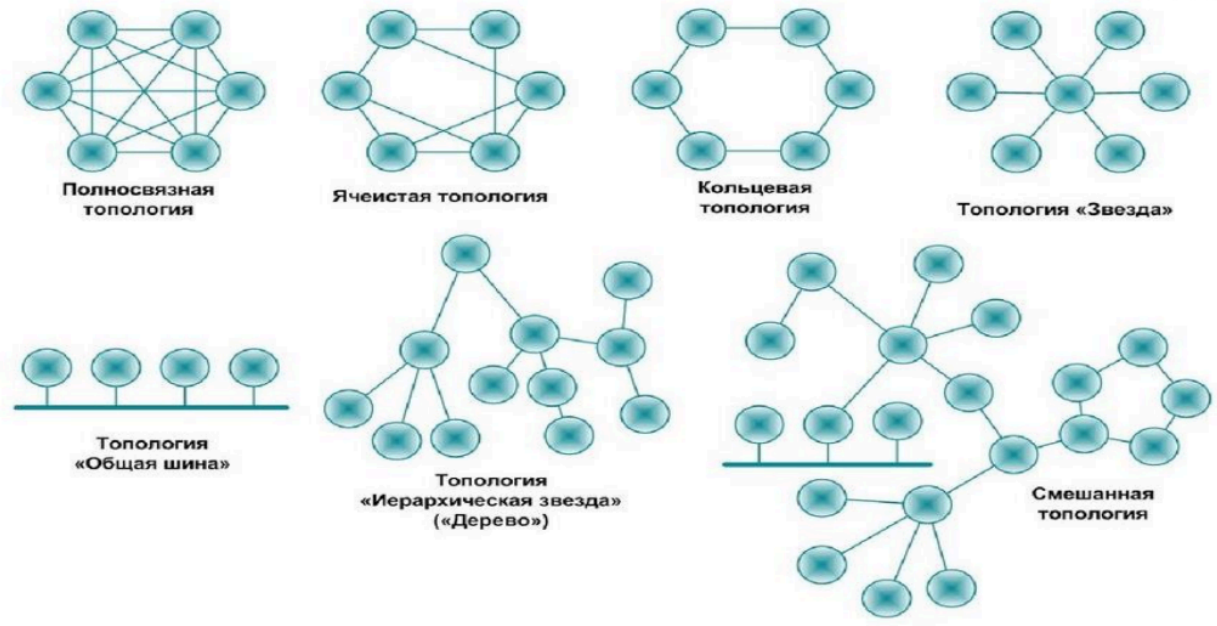
- LPWAN (Low-Power WAN) - глобальная сеть малого энергопотребления
- IAN (Internet Area Network) - сеть без привязке к физическому месту, облачная сеть
- NAN (Near Area Network) - сеть близкого размещения \

Основным критерием классификация сетей будет топология

Топологии бывают физическими и логическими.

- **Физическая** топология представляет собой граф, в котором вершинами являются конечные устройства и коммуникационное оборудование, а рёбрами являются электрические соединения в виде каналов передачи данных. По сути, это схема физического расположения устройств и кабельных систем.
- **Логическая** топология является схемой маршрутов данных в этой сети и определяется настройкой коммуникационного оборудования.
  Физическая и логическая топология могут совпадать, а могут и не совпадать. \

Топологии сетей



1. Полносвязная

> Каждый узел связан с каждым отдельным каналом передачи данных. Каждый с каждым.

Плюсы:

- Высокая отказоустойчивость (достаточно много запасных маршрутов)
- Скорость обмена данных
  Минусы:
- Экономически неэффективна (много провод и доп. сетевые адаптеры для каждого узла)
- Масштабируемость крайне низкая
  Используется крайне редко и в очень маленьких сетях.

2. Ячеистая

> Из полносвязной путём удаления избыточных связей, прямые каналы передачи данных остаются только те, по которым происходит интенсивный обмен данными. \

Плюсы и минусы такие же, как у полносвязной, но чуть ослаблены
На ранних стадиях развития сетей данная топология могла использоваться для создания глобальных сетей.

3. Общая шина

> Компьютеры подключаются к общему кабелю по схеме монтажного или.
> Передаваемая информация будет распространяться в обе стороны \

Плюсы:

- Снижает стоимость проводки
- Унифицирует подключение новых модулей (Высокая масштабируемость)
- Обеспечение почти мгновенного широковещательного(broadcast) обращения ко всем компьютерам в сети
  Минусы:
- Низкая надёжность и отказоустойчивость (любой дефект кабеля или одного из многочисленных разъёмов полностью парализует сеть)
- Низкая производительность сети
- Низкая безопасность
- Возможность образования коллизий

4. Кольцевая топология

> Соответствует сети, в которой компьютеры, соединяясь, образуют замкнутый контур, движение по кольцу происходит в одном направлении, если компьютер распознаёт данные как свои, то он копирует их в свой внутренний буфер, если как чужие, то двигает дальше по кольцу

Плюсы:

- Не имеет особых ограничений по числу абонентов, сеть может содержать более 1000 компьютеров, поскольку каждый из них усиливает сигнал, а общая длина кольца может достигать десятков км. Длина кольца будет формально ограничена только пропускной способностью каналов и времени прохождения сигнала по нему
- Лёгкая диагностика неполадок в сети (если данные вернулись к отправителю, то значит к получателю не дошли; тестовые пакеты отправляются по кольцу и отслеживается, где они теряются)
  Минусы:
- Низкая масштабируемость (Добавление новых элементов достаточно сложная)
- Низкая безопасность
- Неоднозначная отказоустойчивость и надёжность

Практика 4

Продолжение Практики 3

Коммуникационные устройства

Коммутатор

> Используется для доступа к сети и коммутации кадров данных

Коммутатор - это устройство, которое объединяет конечные устройства в локальную сеть \

- коммутация L2 (канальный)
  MAC-адрес
  Коммутатор образует широковещательный домен. ***Broadcast домен*** - это совокупность устройств, для которых справедливо следующее правило:

  > если некоторый хост A отправляет широковещательную рассылку, а другой хост B получает её, то они находятся в одном широковещательном домене

Кроме широковещательных существуют и **коллизионные домены**. Коллизионный домен образуется с помощью моста.
Коллизионный домен - это совокупность устройств, которые борются за один канал передачи данных, и при передачи данных в неё могут возникать коллизии.
Маршрутизатор не образует не один из доменов, однако он будет изолировать широковещательный домен.
`ff-ff-ff-ff-ff-ff` - MAC-адрес широковещательной рассылки
Протокол ARP - позволяет по MAC-адресу найти IP-адрес и наоборот. С помощью протокола ARP формируется ARP-таблица
Коммутатор осуществляет передачу данных на основе коммутации уровня 2 и коммутации уровня 3. Рассмотрим следующие случаи:

1. И отправитель, и получатель находятся в одном широковещательном домене
   1.1 Отправитель знает и IP-адрес и MAC-адрес
   Хост отправитель создаёт данные для отправки инкапсулируя PDU в сетевой пакет, в заголовке которого указывает: свой IP-адрес в качестве отправителя и IP-адрес получателя в соответствующем поле. Далее пакет инкапсулируется в кадр, в заголовке которого: MAC-адрес отправителя и и известный MAC-получателя. После этого кадр отправляется в сеть, где его обрабатывает коммутатор. А именно коммутатор видя MAC-адрес получателя сверяется со своей таблицей коммутации(какой MAC стоит за каким портом), где находит выходной порт, через который отправляет эти данные. На этом коммутация L2 закончена. *Вывод:* Для устройств по обе стороны от коммутатор он остаётся "прозрачным", они его не видят
   1.2 Отправитель знает только IP-адрес получателя, но не знает MAC
   Сначала формируется ARP запрос, т.е. отправитель в заголовке IP пакета указывает свой IP адрес и известный ему IP адрес получателя, инкапсулирует этот пакет в кадр, в заголовке которого указывает в отправителя свой MAC, а в качестве получателя адрес широковещательной рассылки(ff-ff-ff-ff-ff-ff). Кадр

отправляется в сеть, где коммутатор видя broadcast рассылку пересылает её во все свои порты. каждый хост, получив этот кадр декапсулирует его и вытаскивает из IP заголовка IP адрес получателя, далее он сравнивает этот адрес со своим, если они не совпадают, то хост удаляет этот кадр, если адреса совпадают, то получатель формирует кадр-ответ, котором указывает свой MAC и отправляет его источнику ARP-запросов. Получив ответ на свой ARP запрос, хост обновляет свою ARP таблицу, и дальнейшее передача данных пойдёт по сценарию. *Вывод:* Каждое устройство на пути следования кадров будет декапсулировать его, получать информацию о MAC и IP адресах, которые там указаны, и если некоторые связки IP и MAC адресов будут неизвестны этому устройству, то оно их добавит в свою ARP таблицу.

Практика 5

Продолжение практики 4....
2. Коммутация на L3, когда два хоста находятся в разных широковещательных доменах, но хотят передавать друг другу данные
Хост отправитель инкапсулирует PDU в сетевой пакет, в заголовк которого указывает в качестве адреса отправителя свой IP-адрес, в качестве адреса получателя известный ему IP адрес хоста B. Этот пакет инкапсулируется в кадр, в заголовке которого в качестве адреса отправителя хост указывает свой MAC адрес, а в качестве адреса получателя - MAC адрес того устройства, дальше которого он не видит. Это MAC адрес маршрутизатора-шлюза, который ограничивает широковещательный домен, в котором находится хост отправителя. ARP запрос в данном случае не поможет поскольку хост получателя находится в другом широковещательном домене. Кадр через коммутатор доходит до маршрутизатора-шлюза. Маршрутизатор шлюз декапсулирует кадр, вытаскивая MAC адрес получателя. Видит, что это его MAC адрес, значит кадр назначается ему. Декапсулирует из кадра сетевой пакет, сравнивает IP-адрес получателя со своим адресом и понимает, что пакет предназначен не ему. Начинает искать соответствие IP адреса получателя в своей таблице маршрутизации. После нахождения такого соответствия маршрутизатор будет знать IP-адрес следующего устройства, куда нужно будет отправить пакет. В своей ARP таблице он найдёт соответствующий MAC адрес этому IP адрес. Далее маршрутизатор, не изменяя сетевой пакет (IP адрес отправителя и получателя остаются такими же) инкапсулирует его в кадр, в заголовке которого оставит MAC адрес отправителя без изменения, а в качестве MAC адреса получателя поставит MAC адрес следующего устройства, которое нашёл в ARP таблице. После этого кадр дойдёт через коммутатор до хоста B. Если широковещательные домены соединяются не одним шлюзом, а между ними есть транзитные маршрутизаторы, то на каждом из них процедура инкаспуляции, декапсуляции и проверки таблицы будет повторяться. **Этот процесс называется коммутацией L3**

Маршрутизаторы

Маршрутизатор - это устройство, которое соединяет разные локальные сети между собой.
У маршрутизатор есть следующие функции:

- Соединение сетей одного или разных типов
- Изоляция широковещательных доменов
- Поддержка таблицы маршрутизации и работающих протоколов маршрутизации
- Выбор маршрутов и пересылка IP-пакетов
- Доступ к глобальной сети и преобразование сетевых адресов
- Функции безопасности \

Лекция 4

продолжении лекции 3...

5. Активная звезда

Топология звезда соответствует сети, в которой каждый хост будет включаться отдельным кабелем к общему устройству, которая находиться в центре сети. Чаще всего таким устройством является коммутатор. Данное устройство будет перенаправлять информацию, то есть управлять передачей данных, кроме того центральное устройство может служить интеллектуальным фильтром и при необходимости блокировать запрещённые администратором передачи данных.
Особенности:

1. Существенно больше надёжность, чем в топологии шина, однако меньше, чем в других топологиях. Если центральное устройство аварийно завершает свою работу, то вся сеть перестаёт работать.
2. Сложность и стоимость центрального устройства при простоте и дешевизне абонентских.
3. Ограничение по числу абонентов. Количество подключаемых устройств зависит от количества портов на центральном устройстве. Данное ограничение можно преодолеть посредством подключения других коммутаторов, таким образом получается топология ***иерархическая звезда или по-другому дерево.***
   Кроме активной звезды есть и ***пассивная звезда***. Она отличается тем, что в центре нет какого либо активного устройства, там просто идёт соединение проводов, что фактически нам даёт общую шину.

6. Гибридная

На данный момент самой популярной является гибридная, она же смешанная топология, которая за счет преимущества одних топологий будет снижать недостатки других.

По роли компьютера в сети

Помимо представленных существуют ещё классификации сетей. В данном случае классификационным признаком будет роль компьютера в сети: одноранговые сети и сети с выделенным сервером.

Одноранговая сеть

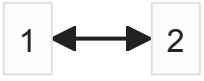
В одноранговой сети каждый хост может являться как владельцем ресурсов (сервер) так и запрашивающим ресурсы у других хостов.

Сети с выделенным сервером

Это такие сети, в котором есть чёткое разделение между хостами, где большая часть хостов являются клиентами, а сервера выполняют исключительную роль владельцев ресурсов.

Многоуровневый подход к построению сети

Протокол - формализованные правила, определяющие последовательность и формат сообщений, которыми обмениваются сетевые компоненты, лежащие на одном уровне, но в разных узлах \



Интерфейс - четко определенные правила и стандартизованные форматы сообщений с помощью которых модули, реализующие протоколы соседних уровней и находящиеся в одном узле, взаимодействуют друг с другом.

Основным приёмом при решение сложных задач является *декомпозиция*, то есть разделение сложной задачи на несколько более простых задач - *модулей*. В результате чего достигается логическое упрощение задачи, а также появляется возможность модификации отдельных модулей без изменения остальной части системы.

При декомпозиции используется многоуровневый подход, который заключается в следующий:

1. Всё множество модулей разбиваются на уровни, которые образуют иерархию
2. Множество модулей, составляющих один уровень, сформировано таким образом, что для выполнения задач они обращаются только к модулям соседнего нижележащего уровня
3. Результаты работы модулей, которые относятся к одному уровню, могут быть переданы только модулям соседнего вышележащего уровня.
   Такая иерархическая организация предполагает чёткое определение функций каждого уровня и интерфейса между ними.

- IPX/SXP
- TCP/IP модель стека
- OSI
- SDN

Практика 6

Продолжение практики 5... \

Межсетевой экран (МСЭ)

Межсетевой экран выполняет следующие задачи:

- Изоляция сетей с различным уровнем безопасности
- Реализация контроля доступа с использованием политик безопасности
- Реализация аутентификации личности пользователя (Triple A- AAA - authorization authefication accounting)
- Реализация удалённого доступа => } VPN
- Поддержка шифрования данных => } VPN
- Реализация преобразования сетевых адресов (Серый и белые адреса)
- Реализация других функций безопасности (НСД - несанкционированный доступ)

Беспроводные устройства

В широком смысле беспроводная сеть это та сеть, которая использует радиоволны, лазерные и инфракрасные сигналы, для замены некоторых или всех в локальной сети. В качестве беспроводных устройств можно использовать следующее:

1. Точки доступа,которые будут разделяться по режимам управления:
   1. Fat AP (Access port)
      Этот режим применяется в частных жилых домах, работает такая точка

доступа независимо и необходимо настраивать отдельно. Такой режим обладает простыми функциями и имеет низкие затраты

2. Fit AP
   Применяется для средних и крупных предприятий. Для работы таких точек доступа требуется контроллер доступа, через который осуществляется управление и настройка режима (Контроллер доступа необязательно аппаратный, может быть и программным)
3. Облачное управление
   Оно применяется для малых и средних предприятий. Для единого управления и настройки требуется облачная платформа. Этот режим предоставляет различный функции и поддерживает автоматическую настройку и запуск в работу (технология plug and play)

2. Контроллер доступа
   Разворачивается на уровне агрегации всей сети, чтобы обеспечить высокоскоростные, безопасные и надёжные услуги беспроводной локальной сети. Контроллер доступа предоставляет услуги беспроводного управления данными с большой ёмкостью, высокой производительностью, надёжностью, простотой установки и обслуживания. Он обеспечивает гибкую организацию сети и энергосбережения.

Стандартизация компьютерных сетей

! Нужно переписать презентацию, так как конспект отдельно от презы
RFC - технические спецификации и стандарты, широко применяемые во всемирной сети
RFC 1149
В зависимости от статуса организаций различают следующие виды стандартов:

1. Стандарты отдельных фирм
   Например, стек протоколов DecNet от фирмы Dec
2. Стандарты специальных комитетов и объединений
   Fast Ethernet - стандартизовал специальный комитет Fast Ethernet Alliance
   ATM - комитет ATM Forum

Практика 7

Продолжение практики 6 (Стандартизация)
3. Национальные стандарты
- ANSI (American National Standart Institute) -> приняла стандарт FDDI
- Росстандарт выпускает ГОСТ
4. Международный стандарт
ISO - международная организация по стандартам -> модель OSI
Некоторые стандарты могут повысить свой класс за счёт популяризации и выхода на международный уровень.

> Пример: Стандарт по архитектуре компьютера IBM (от фирменного до международного)

Существует ещё одна национальная организация, вклад которой в развитие IT и сетей в частности невозможно переоценить: DoD - Department of Depends - министерство обороны США -> (TCP/IP)

Коммутация

Коммутация - процесс соединения абонентов компьютерной сети через различные узлы связи.
Типы коммутации:

1. Коммутация каналов
   При коммутации каналов сеть образует между конечными узлами непрерывный составной физический канал, который состоит из последовательно соединённых коммутаторами промежуточных канальных участков. Условием образование единого канала является равенство скоростей в каждом из его составляющих каналов. Равенство скоростей означает, что коммутаторы не должны буферизировать передаваемые данные.
   Плюсы:
   - Постоянная и известная скорость передачи данных
     Это даёт пользователю сети на основе заранее произведённой оценке установить в сети канал нужной скорости
   - Низкий и постоянный уровень задержки передачи данных, что позволяет качественно передавать данные, чувствительные к задержкам (трафик реального времени - голос, видео)
     Минусы:
   - Нерациональное использование пропускной способности физических каналов.
     Та часть пропускной способности, которая отводится составному каналу, даётся ему на всё время соединения, однако во время передачи данных могут возникать ситуации, когда канал будет простаивать (паузы в разговоре). Невозможность динамического распределения пропускной способности является принципиальным ограничением данной технологии, так как информационный поток в целом будет являться единицей коммутации.
   - Отказ сети в обслуживание запроса на установление соединения (DoS)
     Такая ситуация может возникнуть из-за того, что на некотором участке нужно установить соединение вдоль канала, который уже нагружен максимально возможным количеством потоков (ситуация `Новый год`). И когда конечный абонент уже занят передачей данных с другим абонентом (ситуация `Занят`)
   - Обязательная задержка перед передачей данных из-за фазы установления соединения \

2. Коммутация пакетов

Практика 8

Продолжение практики 7....(Коммутация)

2. Коммутация пакетов

Это техника коммутации была специально разработана для эффективной передачи компьютерного трафика. Типичные сетевые приложения генерируют трафик очень неравномерно с высоким уровнем пульсации скорости передачи данных. При коммутации пакетов используются коммутаторы с буферизацией, а единицей коммутации являются пакеты, то есть небольшие фрагменты, на которые разбиваются исходные сообщения. Если необходимо связать двух конкретных абонентов, то необходимо использовать коммутацию каналов, однако при использовании коммутации пакетов повышается общий уровень пропускной способности сети в целом. Это доказано империческим(практическим) путём и с помощью иммитационного математического моделирования.

Плюсы:

- Высокая общая пропускная способность сети при передачи пульсирующего трафика.
- Возможность динамически перераспределять пропускную способность физических каналов между абонентами в соответствии с реальными потребностями

Сеть с коммутацией пакетов может генерировать следующие задержки:

1. На источнике данных
1. Время на на инкапсуляцию и передачу заголовков
2. Задержки, вызванные интервалами между передачами следующего пакета (джиттер)
2. На получателе
1. Время на декапсуляцию
2. Время на сборку исходного сообщения
3. Задержка на коммутаторе
1. Время на буферизацию пакетов
2. Время на коммутацию, которое складывается из времени ожидания пакета в очереди и времени перемещения пакета в исходящий порт (Зависит от состояния в сети, переменная величина)

Минусы:

- Неопределённость скорости передачи данных
- Переменная величина задержки пакетов
- Возможная потеря данных в случае переполнения буфера

В настоящее время используются методы, которые позволяют преодолеть указанные недостатки - это методы обеспечения качества обслуживания - ***QoS***.

Сети с коммутацией пакетов в наше время используется, для того чтобы одновременно передавать различные виды трафика с применением технологии QoS. Такие методы коммутации считаются сегодня самыми перспективными для создания конвергентных сетей, в которых обеспечивается комплексное, качественное

обслуживание абонентов любого типа.
Коммутации каналов, кроме традиционных телефонных сетей, широко применяется для образования высокоскоростных постоянных соединений в опорных сетях по технологиям SDH и DWDM.
3. Коммутация сообщений
По своим принципа близка к коммутации пакетов, однако есть различия:
1. Буферизация происходит на транзитных компьютерах, точнее на их жёстких дисках. Количество этих транзитных компьютерах старались уменьшить, в идеале до двух штук.
2. Сообщения имеют произвольную длину в отличии от пакетов, и определяется она не технологическими соображениями, а содержанием информации
В сетях с коммутацией сообщений транзитные компьютеры могли связываться как сетью с коммутацией пакетов, так и с сетью с коммутацией каналов.
Применялась данная технология в основном для электронной почты.
$2^{16} = 65536$

Практика 9

Заголовок пакета IPv4

Стандартный размер пакетов IPv4 20 байт без опциональной части (она 0-40 байт).

Заголовок пакетов состоит из следующих полей:

- **Версия**: длина 4 бита
- **Длина заголовка**: длина 4 бита - указывает размер заголовка в 32-битных блоках. Стандартный заголовок 20 байт будет возвращать значение 5.

- **Тип сервиса** - данное поле необходимо для дифференцирования услуг QoS. Размер поля 8 бит. Это поле разделено на два подполя. Первое - это приоритет пакета, 3 бита. Нулевой приоритет - это самый обычный, седьмой - самый высокий приоритет. Следующие 3 бита образуют подполе критерия выбора маршрута: D - по минимальной задержке, T - по максимальной пропускной способности, R - по максимальной надёжности. Выбирается обычно один бит из этих трёх, редко, когда 2, 3 - бесполезно. Оставшиеся два бита в этом поле зарезервированы и равны 0.
- **Общая длина** - размер 16 бит, означает общую длину пакета с учётом заголовка и данных, значение измеряется в байтах. Максимальный размер пакета: 65535 байт. В основном используется пакеты размером 576 байт и меньше, однако даже эти пакеты не всегда вмещаются в пропускную способностью. Следующие три поля нужны при фрагментации пакетов.
  Фрагментация - процесс разделения пакета на несколько частей, если он не вмещается в пропускную способность канала передачи данных
- **Идентификатор** - занимает 16 бит, используется для распознавании пакетов на получателе. Все фрагменты исходного пакета должны иметь одинаковое значение данного поля.
- **Флаги** - размер 3 бита, определяет контроль над фрагментацией. Первый бит зарезервирован, всегда 0. Второй бит - нефрагментировать, единица в этом бите означает, что фрагментация запрещена, а 0 - разрешена. Третий - "У пакета есть ещё фрагмент", 1 - за текущем фрагментом есть ещё фрагменты, 0 - это последний фрагмент.
- **Смещение фрагментов** - длина 13 бит, указывает на относительное положение фрагмента в исходном фрагменте. Смещение должно быть кратно 8-ми байтам.
- **TTL** - время жизни пакета, длина 8 бит, измеряется в хопах. Максимум 255 хопов.
- **Протокол** - длина 8 бит, указывает на протокол, который будет обрабатывать данные на получателе, после того, как их обработает IP протокол. (например, ICMP - 1, TCP - 6, UDP - 17)
- **Контрольная сумма** - 16 бит, проверить целостность данных в пакете
- **IP адрес источника** - 32 бита
- **IP адрес получателя** - 32 бита
- **Опции** - кратно 32 бита, поле опции является необязательным и используется при отладке сети. В опциях можно указать точный маршрут, то есть регистрировать проходимые пакетом маршрутизаторы, также можно указывать данные системы безопасности и временные метки. Если информация в поле опции не кратно 32 битам, то соответствующее количество нулей записывается в поле выравнивание.

Практика 10

IP - адресация

IPv4 адреса можно классифицировать различными способами:

1. Классы A,B,C,D,E
   Сетей класса C больше чем сетей класса A, однако хостов больше в сетях класса A. Это объясняется тем, что сетевая часть в сетях класса A 8 бит, а хостовая 24, тогда как у сетей класса C сетевая часть 24 бита, а хостовая 8
2. Частные и публичные IP адреса (Серые и белые)
   Публичный адрес (белый) используется в сети Интернет, является уникальным и дублировать его нельзя. Серые адреса используется в локальной сети, в сети Интернет их использовать нельзя, серый IP адрес может повторяться, но в разных локальных сетях. Для того чтобы отсрочить момент исчерпания IPv4 были созданы два костыльных решения. Во-первых, это ***серый адреса***, во-вторых протокол ***NAT***, который позволяет транслировать белые адреса в серые и наоборот.
3. UNICAST, MULTICAST (Групповые), BROADCAST (Широковещательные)
   Существуют специальные IP адреса, которые не относятся не к одной классификации.
4. **Ограниченный широковещательный адрес - 255.255.255.255** - Может использоваться только в качестве получателя, область действия ограничивается шлюзом
5. **Любой IP адрес - 0.0.0.0** - Может использоваться только в качестве отправителя. Если интерфейс хоста не получает свой IP адрес во время запуска, то он отправляет запрос к DCHP серверу, где в качестве адреса отправителя указывает данный IP адрес, ожидается, что DHCP сервер назначит доступный IP адрес данному хосту. Второй вариант использование в качестве статического маршрута по умолчанию. На серверах этот IP адрес даёт команду прослушивать и принимать соединение с любого IP адреса. На клиентских устройствах данный IP адрес указывает на то, что клиент не подключён к сети в TCP/IP и это устройство работает в автономном режиме.
6. **Loopback/localhost - 127.0.0.0/8** - Localhost работает с так называемым кольцевым трафиком. Этот трафик генерируется локальным хостом и не выходит за его пределы. Данные адрес может использоваться как отправитель, так и получатель. Такой кольцевой трафик нужен для тестирования ПО на самом локальном устройстве.
7. **Link-local - 169.254.0.0/16** - Данный диапазон используется для указания адреса отправителя, в том случае, если в сети нет DHCP сервера или с ним есть какие-то проблемы. Данный адрес используется для временной связи.

Протокол ICMP

ICMP - вспомогательный протокол сетевого уровня, на который возложены следующие функции:

1. Проверка доступности сетевых объектов (хосты, сети, коммуникационные устройство, веб сайты)
   Данная функция решается с помощью утилиты ping. Сообщения, которой являются самыми популярными у ICMP (эхо-запрос, эхо-ответ). Также в ICMP сообщении есть поле содержания, размер которого 32 бита, если поле не используется, то все биты заполняются нулями. В сообщении эхо-запроса данное поле содержит идентификатор и последовательный номер. Источник будет ассоциировать эхо-ответ с эхо-запросом по идентификатору и последовательному номеру.
2. Диагностика сети (traceroute)
   Функции диагностики сети реализуются с помощью команды tracert, которая работает по следующему алгоритму. Каждая итерация предполагает увеличение TTL на единицу, а значит передвижение пакета на один маршрутизатор дальше, каждый маршрутизатор отправляет пакет о том, что отправленный пакет закончил существование на этом маршрутизаторе. Там происходит, пока пакет не достигнет хоста назначения.
3. Перенаправление трафика.
   В случае перенаправления в поле содержимое указывается IP адрес шлюза, на который можно перенаправить трафик. Данная функция оптимизирует маршруты передачи данных.

   > Ситуация со слайда:
   > Хост А хочет отправлять запросы на сервер А, на данном хосте прописан шлюз по умолчанию: роутер В. Поэтому запросы для сервера А отправляются сначала на шлюз по умолчанию, то есть на роутер В. Роутер В понимает, что данный маршрут неоптимальный и создаёт ICMP сообщение (Redirect), в котором указывает в поле содержимое указывает IP адрес роутера А, и отправляет сообщение хосту А. Хост А, получив ICMP перенаправление, переписывает шлюз по умолчанию на роутер А.

Практика 11

При планировании ip-адресов нужно соблюдать следующие правила:

1. Уникальность - означает, что у каждого хоста в сети должнен быть уникальный ip-адрес
2. Непрерывность - смежные адреса можно легко суммировать в иерархических сетях. Это приводит к уменьшению размеров таблицы маршуртизации, что приводит к ускорению вычисления маршрутов.
3. Масштабируемость - адреса должны быть зарезервированны на каждом уровне, чтобы обеспечить смежное адресное пространство для суммирования маршрутов при расширении сети. (Резервировать нужно не только адреса, но и подадреса)

4. Сочетание топологий и сервисов - Правильное планирование ip-адресов помогает легко определить положение устройств и типы сервисов по значению ip-адресов, а также облегчает развертывание технологии QoS. Если знают ip-сети, то знаю ip-адрес шлюза(он следующий по ip-адреса сети)(по правилу хорошего тона сетевика). Для серверов(файловый и принт) выделяют ip-адрес по краям диапазона. 192.168.1.254 - файловый сервер. 192.168.1.253 - МФУ(эти адреса статические, так как иначе при печати нужно будет задавать новые ip-адреса). Серединка отдается обычным хостам

IPv6

Доп. инфа в презентации
Заголовок IPv6 - 40 байт
Включает в себя следующие поля:

1. Поле версии: размер 4 бита(значение: 6)
2. Класс трафика: длина 8 бит - указывает класс и приоритет пакета(похож на поле: тип сервиса в IPv4), используется в основном для управления QoS
3. Flow Label - метка потока - новое поле, которго не было в IPv4. Это поле нужно для того, чтобы различать трафик реального времени. Метка потока и ip-адрес источника вместе могут иденцифицировать уникальной поток данных.
4. Payload Length - длина полезной нагрузки(из Totoal Length - IHL(IPv4)) - длина 16 бит, указывает размер ПДУ(единица передачи информации для верхнего уровня)
5. Next Header - размер 8 бит, соответствует полю "Протокол" в IPv4.
6. Hop Limit - длина 8 бит, соответствует полю TTL в IPv4.(255 прыжков)
7. Адрес отправителя и адрес получателя - по 128 бит.

IPv6 адреса делятся на следующие категории:

1. Индивидуальные адреса(unicast) - иденцифицируют один интерфейс устройства, пакеты, отправленные на этот адрес, доставляются только на этот интерфейс. Среди индивидуальных адресов есть специальный
2. (0000:0000:0000:0000:0000:0000:0000:0000/128 - ::/128)
3. loopback - 0000:0000:0000:0000:0000:0000:0000:0001/128 - ::1 / 128
4. Многоадресные(multicast) адреса - иденцифифируют группу интерфейсов. Пакет, посылаемый на такие адреса доставляются всем интерфейсам участников группы расслыки.
5. Произвольные (anycast) адреса - позволяют адресовать группу интерфейсов, однако, в отличие от multicast, пакеты, посылаемые на произвольный адрес, доставляются хотя бы на один из адресов группы рассылки.(обычно на ближайший). BROADCAST В IPv6 НЕТ!!!
6.

Лекция 5

Модель OSI

> Пример многоуровневого подхода

Модель OSI является эталонной модели, которая в жизни не используется, но к которой все стремятся. \

1. Физический уровень

> Физический уровень имеет дело с **передачей битов по физическим каналам** связи(оптоволокно, витая пара, коаксиальный кабель, WiFi).

К этому уровню имеют отношения характеристики физических сред передачи данных такие как: полоса пропускания, помехозащищённость, волновое сопротивление и т.д. На этом же уровне определяются характеристики электрических сигналов такие как: уровни напряжения и тока, сопротивление, тип кодирование и скорость передачи данных. Также на этом уровне определяются типы разъёмов и назначения каждого контакта. Функции физического уровня со стороны компьютера будет выполнять сетевой адаптер и последовательный порт(устарел, только консольный кабель). Как таковых протоколов на физическом уровне нет, но есть спецификация: 10-BASE-T, которая предполагает использование витой пары 3 категории с волновым сопротивление 100 Ом, с разъёмом RJ-45, длиной физического сегмента 100 м и т.д.

2. Канальный уровень

На физическом уровне передаются биты, однако при этом не учитывается, что в некоторых сетях, в которых линии связи используются попеременно, физическая среда передачи данных может быть занята, **проверкой доступности среды передачи данных занимается канальный уровень**. Единицей передачи является кадр. На канальном уровне используется ***физическая адресация по MAC-адресам***.
Основным устройством на канальном уровне является **коммутатор**.
Ещё важной задачей данного уровня является реализация механизма обнаружения и коррекции ошибок. С помощью контрольной суммы на канальном уровне решается проблема искажения данных. Коррекция ошибки происходит с помощью повторной передачи кадра.
Передача кадров между хостами на данном уровне происходит с помощью протоколов, которые поддерживаются строго определённую топологию связи.
Функции канального уровня со стороны компьютера реализуются сетевыми адаптерами и драйверами. В глобальных сетях, которые редко обладают регулярной топологией, канальный уровень обеспечивает обмен данными между двумя соседними компьютерами, которые соединены индивидуальными линиями связи (P2P - Point to point)
Примером протокола канального уровня является **протокол ARP**.

3. Сетевой уровень

Единицей передачей данных является **сетевой пакет**, используется **IP адресация**, основным устройством является **маршрутизатор**.
Основная задача сетевого уровня: **решение задачи маршрутизации**, то есть определение оптимального маршрута пути следования пакета. Другие задачи:

1. Связь между сетями разных топологий и технологий. Чтобы с одной стороны сохранить простоту передачи данных типовых топологий, а с другой стороны допустить использование других топологий, вводится сетевой уровень.
2. На данном уровне может решаться задача фильтрации трафика. С помощью маршрутизатора можно создать надёжные и гибкие барьеры на пути нежелательного трафика.
3. Согласование разных типов адресации.
   На сетевом уровне существует два типа протоколов: протокол адресации (IP, ICMP), протокол маршрутизации (RIP, IS-IS, OSPF, BGP). Протоколы маршрутизации - это те протоколы, с помощью которых маршрутизаторы собирают информацию о топологии межсетевых соединений. Например, обмениваясь своими таблицами маршрутизации.
   Функции сетевого уровня будут реализовываться программными модулями операционной системы хоста и маршрутизатора.

4. Транспортный уровень

Единицей передаче данных являются сегменты и датаграммы. Используется адресация по портам. Протоколы UDP и TCP.
На пути от отправителя к получателю данные могут быть искажены или утеряны и, хотя некоторые функции по обнаружению и исправления ошибок могут быть на других уровнях. Именно транспортный уровень обеспечивает приложениям или верхним уровнем модели передачу данных с той степенью надёжностью, которая им требуется. Транспортный уровень предоставляет определённый класс сервиса, которые различаются по следующим характеристикам:

1. Срочность
2. Возможность восстановления прерванной связи
3. Наличие средств мультиплексирования нескольких соединений между различными прикладными протоколами через общий транспортный протокол. (Несколько потоков передаются одни протоколом транспортного уровня)
4. Способность к обнаружению и исправления ошибок передачи данных (Искажение данных, потеря данных, дублирования данных)
   Для того, чтобы обнаружить ошибки используются следующие средства:
5. Контрольная сумма
6. Предварительное установление соединения

7. Циклическая нумерация сегментов
8. Установление тайм-аутов доставки
9. Подтверждение получения сегментов
10. Повторная передача сегментов
    Транспортный уровень является промежуточным между сети зависимыми уровнями (нижние 3) и сети независимыми уровнями (верхние 3)

5. Сеансовый уровень

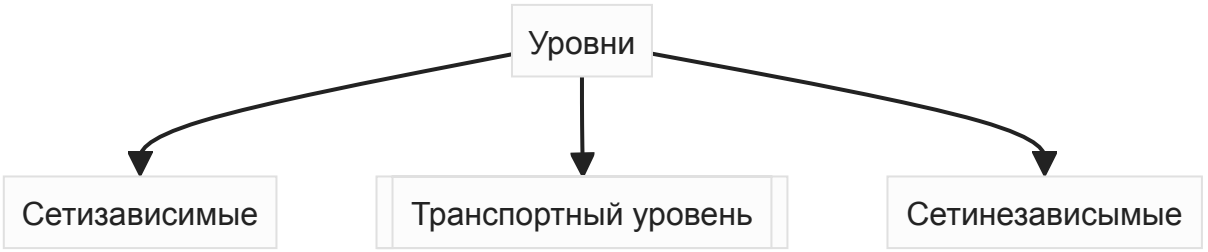
Обеспечивает управление диалогом, то есть фиксирует какая из сторон является активной, и предоставляет средства синхронизации. Средства синхронизации предоставляют собой контрольные точки, для восстановления диалога.
В жизни протоколов сеансового уровня очень мало (SMP, RPC), поскольку функции сеансового уровня берут на себя протоколы либо транспортного, либо прикладного уровня.

6. Уровень представлений

Данный уровень имеет дело с формой представления передаваемой по сети информацией, не меняя при этом её содержания. За счёт данного уровня информация, передаваемая прикладным уровнем одной системы будет всегда понятна прикладному уровню другой системы. Также на этом уровне происходит шифрование и дешифрование, повышая её безопасность. Примером протоколов является SSL.

7. Прикладной уровень/уровень приложений

Единицей передачей данных будет PDU, а также сообщения или данные. Прикладной уровень является набором разнообразных протоколов, с помощью которых пользователь получает доступ к разделяемым ресурсам, а также организует свою совместную работу.



Практика

Модель TCP/IP

Заголовок TCP состоит из следующих полей:

1. Порт источника и порт назначения по 16 бит, каждый. Эти поля идентифицирует приложения, которые отправляют и принимают сегменты. На адресном уровне происходит ***адресация по портам***. Порт представляет собой 16-разрядное без знаковое целое число (0 - 65535). Первые 1024 порта зарезервированные под наиболее популярные сетевые приложения и службы, остальные доступны для общего использования называются динамическими или эмфимерными. Зарезервированные на входящий трафик, динамические - на исходящий.
2. Порядковый номер - 32 бита. Каждый байт отправляемый через TCP соединение имеет свой порядковый номер. Значение этого поля указывает на номер первого байта в отправляемом сегменте.
3. Номер подтверждение - 32 бита. Это порядковый номер первого байта следующего сегмента, который ожидается для получения. Значение этого поля равно порядкового номеру последнего байта, полученного в предыдущем сегменте и увеличенного на единицу
4. Длина заголовка - 4 бита. Показывает количество 32-битных (4 байта) блоков в заголовке.
5. Поле резерв - 6 бит. 6 нулей
6. Биты управления - 6 бит. Показывает состояние TCP соединение. Существует следующие флаги:
    1. URG - Указатель срочности. Показывает приоритет сегмента
    2. PSH - Выталкивает данные из очереди в буфере.
    3. ACK - Подтверждает получение данных
    4. RST - Сбрасывает TCP соединения по завершению или после разрыва
    5. SYN - Синхронизирует соединение хостов
    6. FIN - Завершает TCP соединение
7. Окно - 16 бит. Используется для управления потоком TCP. Значение данного поля показывает максимальное количество байт, которое получатель разрешил отправлять ему (макс. 65535)
8. Контрольная сумма - 16 бит. Некое число, чтобы понимать искаженно ли содержимое или нет.
9. Указатель срочности - 16 бит. Данное поле действует только, если установлен флаг URG. Это поле показывает, что отправитель передаёт данные в аварийном режиме. Значение поля указывает на то, сколько срочных байт находится в сегменте. (Срочные данные обычно в начале сегмента)
10. Опции (необязательное поле) - от 0 до 40 байт.

TCP соединение состоит из трёх этапов

1. Трёхстороннее рукопожатие
2. Передача данных
3. Четырёхстороннее рукопожатие. Поскольку TCP соединение работает в режиме duplex, появляется 4-е рукопожатие.

На втором этапе используется механизм ***динамических окон***. Отправитель будет всегда настаивать на максимальном размере окне, однако последнее слово всегда за получателем, который будет ориентироваться на свой внутренний буфер. окно показывает сколько нужно передать байт на одно подтверждение от получателя. Важно соблюдать баланс размера окна, поскольку если оно будет слишком маленьким, то количество подтверждений может сильно загрузить сеть, с другой стороны, если оно будет слишком большим, то при потере хотя бы одного байта придётся повторно отправлять весь сегмент.

Установление TCP-соединения – трехстороннее квитирование

Перед отправкой данных приложению на базе TCP необходимо установить соединение через трехстороннее квитирование.

- Процесс установления TCP-соединения:
  - Инициатор TCP-соединения (ПК1 на рисунке) отправляет первый TCP-сегмент с установленным флагом SYN. Начальный порядковый номер (a) является случайно сгенерированным числом. Номер подтверждения равен 0, потому что ранее от ПК 2 не поступали сегменты.
  - После получения корректного TCP-сегмента с установленным флагом SYN, получатель (ПК2) отвечает TCP-сегментом с установленными флагами SYN и ACK. Начальный порядковый номер (b) является случайно сгенерированным числом. Поскольку сегмент является ответом для ПК1, номер подтверждения равен a+1.
  - После получения TCP-сегмента, в котором установлены флаги SYN и ACK, ПК1 отвечает сегментом, в котором установлен флаг ACK, порядковый номер равен a+1, а номер подтверждения – b+1. После того, как ПК2 получает сегмент, устанавливается TCP-соединение.

Порядковый номер TCP и номер подтверждения

- TCP использует поля Порядковый номер и Номер подтверждения для реализации надежной упорядоченной передачи данных.

- Предположим, что ПК1 необходимо отправить сегменты данных на ПК2. Процесс передачи будет следующим:
    1. ПК1 нумерует каждый байт, отправляемый через TCP-соединение. Предположим, что номер первого байта равен a+1. Номер второго байта будет равен a +2, третьего байта – a+3 и так далее.
    2. ПК1 использует номер первого байта каждого сегмента данных в качестве порядкового номера и отправляет TCP-сегмент.
    3. После получения от ПК1 TCP-сегмента ПК2 необходимо подтвердить сегмент и запросить следующий. Как определяется следующий сегмент данных? Порядковый номер (a+1) + длина полезной нагрузки = порядковый номер первого байта следующего сегмента (a+1+12)
    4. После получения TCP-сегмента, отправленного ПК2, ПК1 обнаруживает, что номер подтверждения равен a+1+12, что указывает на получение сегментов от a+1 до a+12. Порядковый номер будущего сегмента должен быть a+1+12.
- Чтобы повысить эффективность отправки, отправитель может отправлять несколько сегментов данных одновременно, которые затем по очереди будет подтверждать получатель.

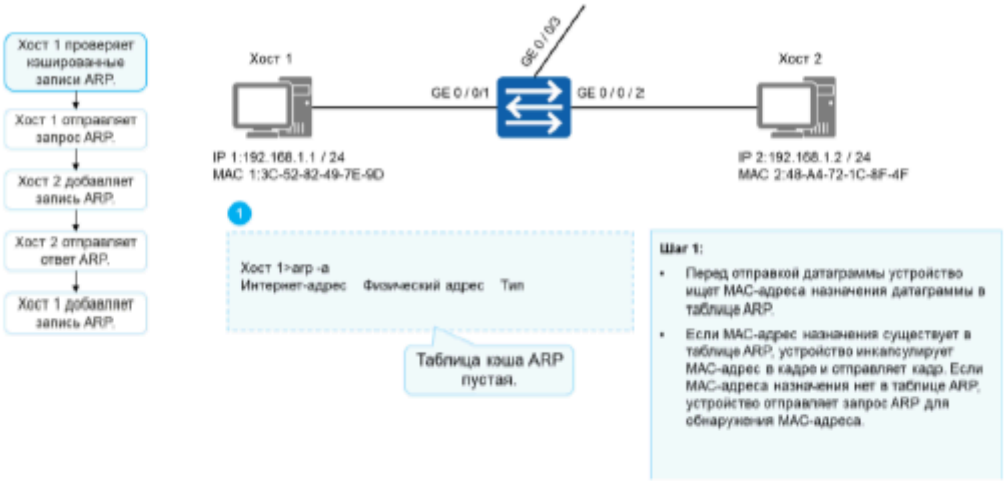
Выключение TCP - четырехстороннее квитирование

- После завершения передачи данных TCP-соединение должно выполнить четырехстороннее квитирование для разъединения и освобождения ресурсов системы.

- TCP поддерживает передачу данных в дуплексном режиме. Это означает, что данные можно передавать в обоих направлениях одновременно. Перед передачей данных TCP устанавливает соединение в обоих направлениях посредством трехстороннего квитирования. Поэтому после завершения передачи данных соединение должно быть разъединено в обоих направлениях. Это показано на рисунке.

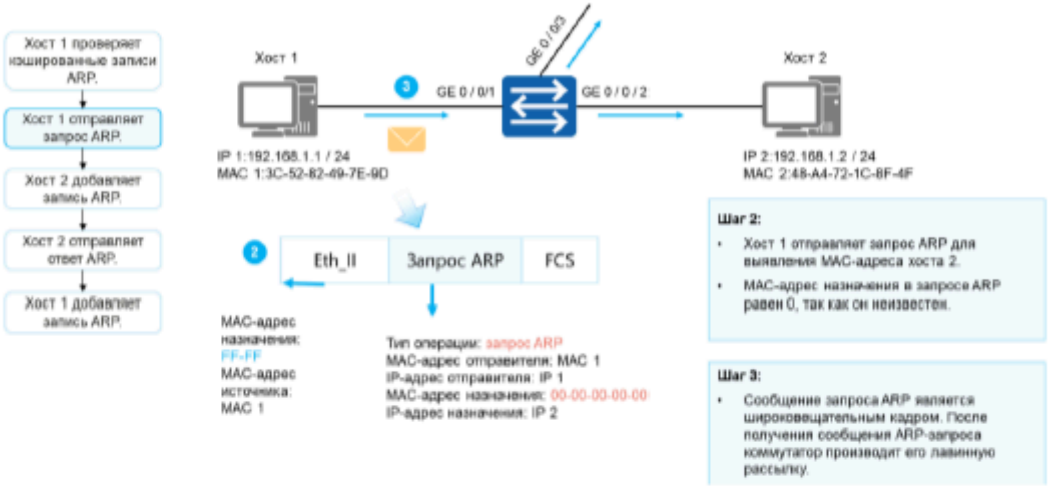
    1. ПК1 отправляет TCP-сегмент с установленным флагом FIN. Сегмент не содержит данных.
    2. После получения ПК1 сегмента TCP ПК2 отвечает сегментом TCP с установленным флагом ACK.
    3. ПК2 проверяет необходимость отправки данных. Если необходимо, ПК2 отправляет данные, а затем TCP-сегмент с флагом FIN для закрытия соединения. Если нет, ПК2 напрямую отправляет TCP-сегмент с установленным флагом FIN.
    4. После получения TCP-сегмента с установленным флагом FIN ПК1 отвечает сегментом с флагом ACK. Затем TCP-соединение разрывается в обоих направлениях.

Принципы реализации ARP (1)



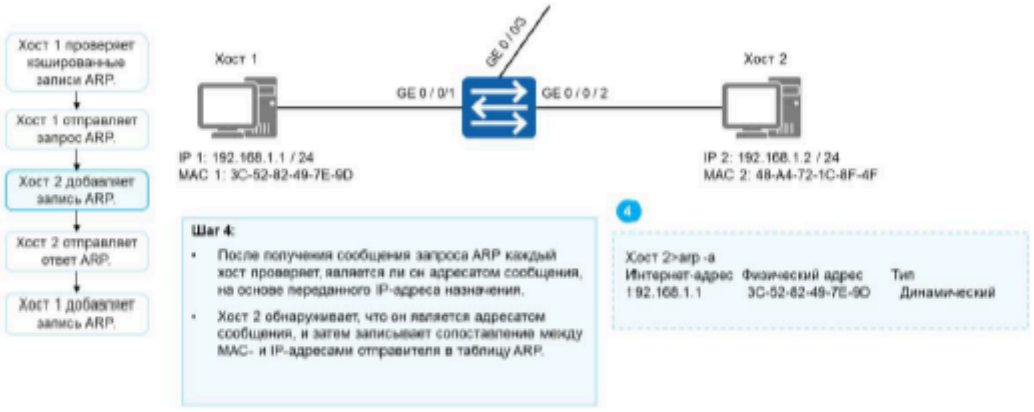
- Как правило, у сетевого устройства есть кэш ARP. В кэше ARP хранится сопоставление между IP-адресами и MAC-адресами.
- Перед отправкой датаграммы устройство выполняет поиск в своей таблице ARP. Если устройство находит соответствующую запись ARP, оно инкапсулирует соответствующий MAC-адрес в кадре и отправляет кадр. Если устройство не находит соответствующую запись ARP, оно отправляет запрос ARP для обнаружения MAC-адреса.
- Полученное сопоставление между IP-адресом и MAC-адресом хранится в таблице ARP в течение некоторого периода. В течение периода действия (180 с по умолчанию) устройство может напрямую искать в этой таблице MAC-адрес назначения для инкапсуляции данных без выполнения запроса ARP. После истечения срока действия запись ARP автоматически удаляется.
- Если устройство назначения расположено в другой сети, устройство-источник ищет в таблице ARP в поиске MAC-адрес шлюза адреса назначения и отправляет датаграмму шлюзу. Затем шлюз переадресует датаграмму устройству назначения.

Принципы реализации ARP (2)



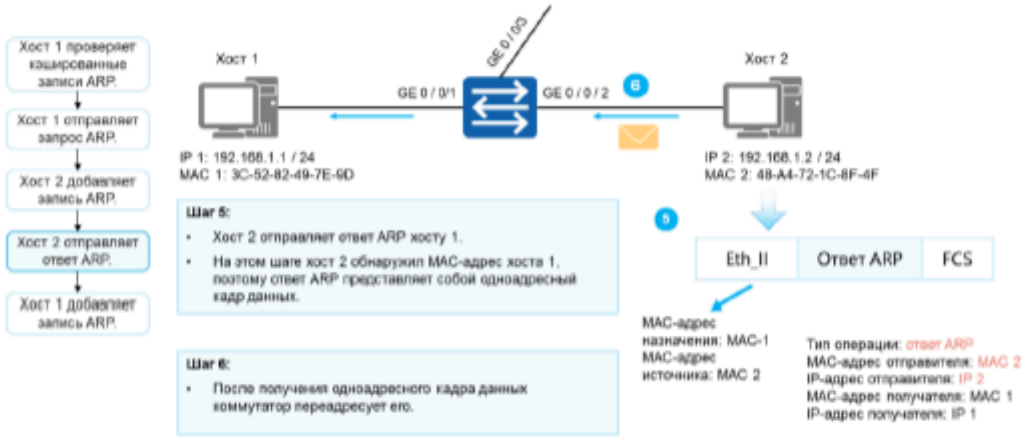
- В этом примере таблица ARP хоста 1 не содержит MAC-адрес хоста 2. Поэтому хост 1 отправляет ARP-запрос для обнаружения MAC-адреса назначения.
- Запрос ARP инкапсулируется в кадре Ethernet. MAC-адрес источника в заголовке кадра – это MAC-адрес хоста 1 на стороне передачи. Поскольку хост 1 не знает MAC-адрес хоста 2, MAC-адрес назначения является широковещательным адресом FF-FF-FF-FF-FF-FF
- Запрос ARP содержит MAC-адрес источника, IP-адрес источника, MAC-адрес назначения и IP-адрес назначения. MAC-адрес назначения - все 0. Запрос ARP передается всем хостам сети, включая шлюзы.

Принципы реализации ARP (3)

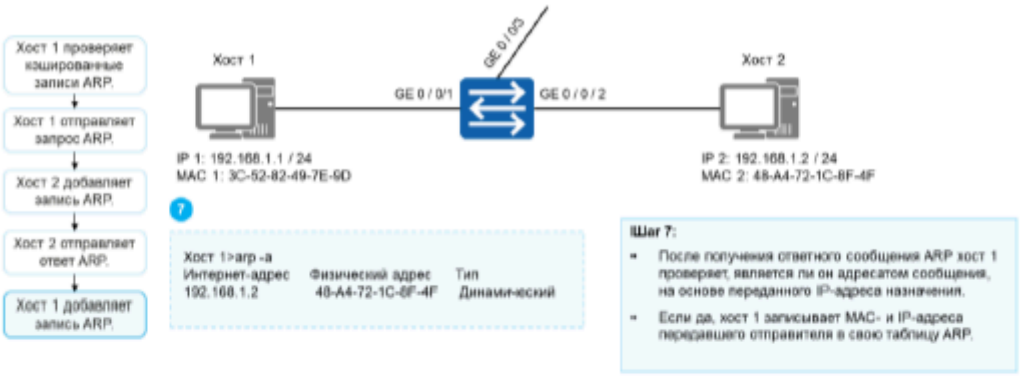


- После получения запроса ARP каждый хост проверяет, является ли он адресатом сообщения, на основе переданного IP-адреса назначения. Если нет, хост не отвечает на запрос ARP. Если да, хост добавляет MAC- и IP-адреса отправителя, содержащиеся в запросе ARP, в таблицу ARP, а затем отвечает сообщением с ответом ARP.

Принципы реализации ARP (4)



- Хост 2 отправляет ответ ARP на хост 1.
- В ответном сообщении ARP IP-адрес отправителя является IP-адресом хоста 2, а IP-адрес получателя - IP-адресом хоста 1. MAC-адрес получателя является MAC-адресом хоста 1, а MAC-адрес отправителя - MAC-адресом хоста 2. Тип операции – ответ.
- Ответные сообщения ARP передаются в одноадресном режиме.



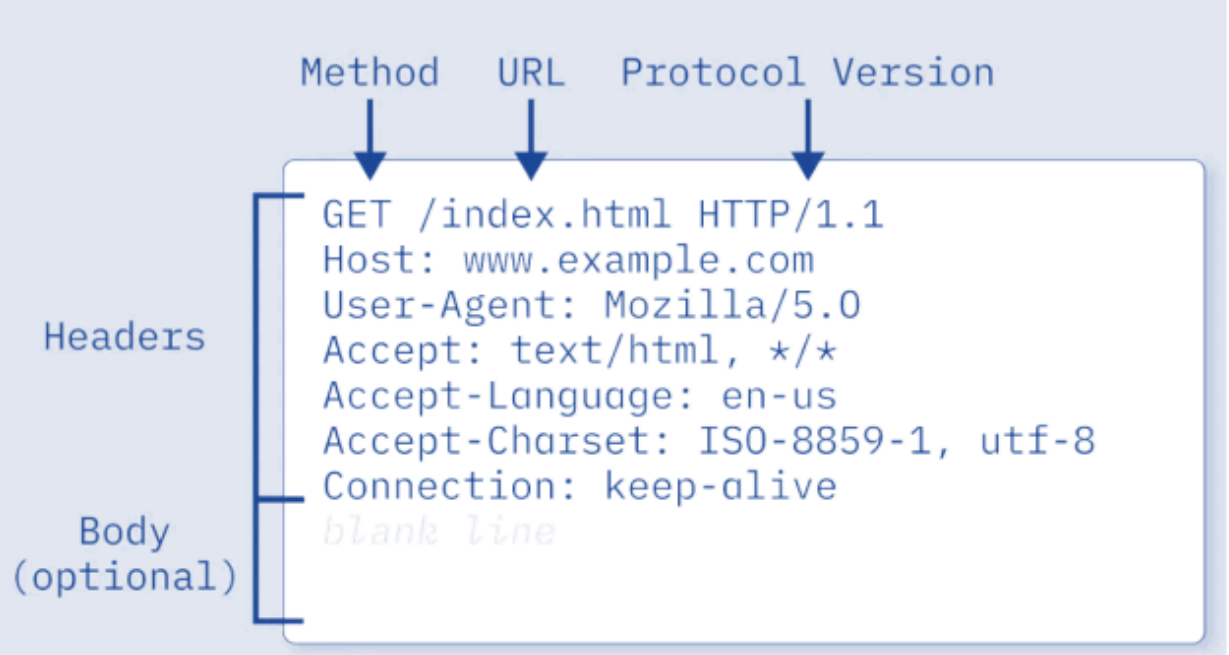
- После получения ответного сообщения ARP хост 1 проверяет, является ли он адресатом сообщения, на основе переданного IP-адреса назначения. Если да, хост 1 записывает MAC- и IP-адреса отправителя в свою таблицу ARP.

HTTP

HTTP (от англ. HyperText Transfer Protocol) — протокол передачи гипертекста. Это набор правил, по которым данные в интернете передаются между разными источниками, обычно между компьютерами и серверами. Интернет-протокол HTTP — это шаблон, по которому формируется запрос на передачу данных, а затем передаются интернетстраницы, видео, аудио и текст. Чаще всего с помощью HTTP передают веб-страницы, то есть контент сайтов, которые отображаются в интернете. Протокол HTTP нужен для стандартизации. Благодаря ему все компьютеры в интернете могут расшифровать присланные данные и отправлять их в виде, понятном другим компьютерам. Структура HTTP-сообщения всегда одинакова:

1. Стартовая строка, в которой определяется адрес, по которому отправляется запрос, и тип сообщения. Указывается метод, который определяет действия при получении этого сообщения. Это может быть чтение данных, их отправка, изменение или удаление.
2. Заголовки (Headers), в которых прописаны определённые параметры сообщения. Например, может быть напрямую задан язык.

3. Тело запроса (Request Body), текст сообщения — данные, которые передаются. Например, файлы, отправляемые на сервер.



Основные методы HTTP

В заголовках HTTP-сообщений используются методы — по ним сервер и клиент понимают, в чём именно суть сообщения. Разберём пять самых популярных методов.

Существуют и другие методы, однако они используются гораздо реже. Например, OPTIONS возвращает описание ресурса — методы, настройки кэширования, тип контента.

Преимущества и недостатки HTTP

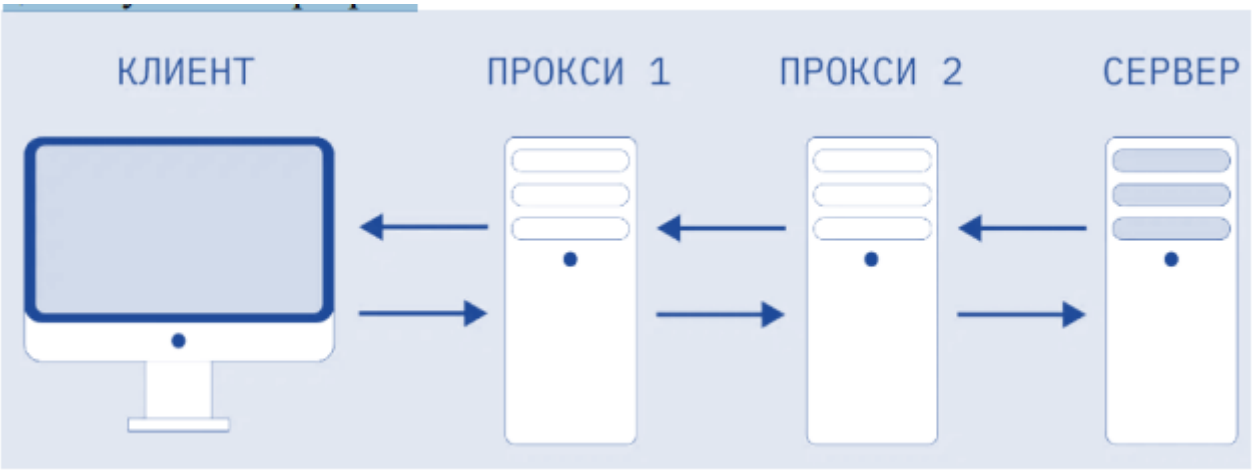
✅ Преимущества
● Расширяемость. В 1992, когда HTTP только появился, он был совсем простым. Но со временем протокол передачи гипертекста обрастал новыми методами и возможностями, и он всё ещё способен к расширению и изменению.
● Подробная документация. HTTP подробно описан на разных языках, и в документации есть ответы на большинство вопросов.
● Распространённость. HTTP — самый популярный протокол в интернете. Он считается основным и универсальным, на нём работают практически все сайты в мире.
❌ Недостатки
● Отсутствие навигации. HTTP не позволяет запросить все доступные ресурсы и их параметры. Это исправили расширением WebDAV, но в

самом HTTP такая возможность не предусмотрена.
● Проблемы с распределёнными запросами. Когда HTTP только создавали, время обработки запросов не учитывали, но сейчас с повышением нагрузки на серверы это иногда становится проблемой.
● Незащищённость. Базовый HTTP без шифрования совершенно небезопасен — любой может перехватить данные запроса и узнать всё: логины, пароли, данные банковских карт. Поэтому и появился HTTPS.
Сейчас большинство недостатков HTTP исправлены надстройками и не заметны на практике. Поэтому протокол передачи гипертекста остаётся актуальным, и прекращать применять HTTP никто не планирует.
Чем можно управлять через протокол HTTP
HTTP позволяет не только получать и передавать данные, но и управлять поведением сайта. Например, с помощью заголовков и тела сообщения можно управлять:
● Кэшем. Сервер расскажет, что и как долго кэшировать на стороне клиента. Так браузер клиента поймёт, какие элементы сайта нужно положить в кэш для будущего переиспользования.
● Аутентификацией. В заголовке HTTP можно настраивать специальные сессии и куки для быстрого входа по логину и паролю. Именно благодаря этому можно заходить на сайты, не вводя повторно логин и пароль.
● Сессиями. Текст HTTP-запроса позволяет серверу запомнить состояние сайта на стороне клиента. Например, чтобы сохранить его корзину или какие-то введённые данные даже при обновлении страницы.

Составляющие систем на HTTP

Система, которая работает на HTTP, требует минимум два участника. Один из них — клиент, который отправляет запросы. Как правило, это компьютер пользователя с браузером. Второй — сервер, который отвечает на запросы, отправляя клиенту нужную информацию. Обычно это компьютер, на котором запущен сайт: части его системы и база данных.

Иногда в этой схеме появляются дополнительные элементы — проксисерверы. Они располагаются между клиентом и сервером и обрабатывают запросы — например, дополнительно их шифруют или кэшируют. Часто прокси используют, чтобы сделать запрос анонимным: сервер отвечает не напрямую клиенту, а через указанный прокси или цепочку таких серверов.



Клиент — не всегда компьютер. Это может быть смартфон, планшет либо умное устройство: TV, колонка, часы.

Разработчику сайтов важно понимать, как именно работает HTTPпротокол и что можно делать с его помощью. Ведь именно по этому протоколу он будет настраивать передачу данных от сервера к клиенту

Порядок работы HTTP-протокола

Сама работа HTTP-протокола максимально проста — клиент передаёт запрос, сервер формирует ответ и передаёт его обратно. До отправки запроса и после получения ответа происходят фоновые задачи, незаметные для пользователя. Поэтому, чтобы понять, как работает протокол HTTP, рассмотрим этот процесс целиком:

1. Формирование URL или переход по введённой ссылке в браузере клиента. Браузер анализирует URL и понимает, что по этому адресу нужно отправить HTTP-запрос.
   Этот пункт актуален, если клиент переходит по ссылке браузера. Но это не обязательно: иногда HTTP-запрос формируется после каких-то действий пользователя автоматически. Либо всё происходит вообще не в браузере — например, в случае с умными устройствами. Тогда первый пункт просто пропускается, а запрос инициируется после некоторого триггера.
2. Клиент формирует и отправляет запрос. Для этого генерируется стартовая строка, заголовки и тело запроса в зависимости от того, что именно будет запрошено у сервера. Это может быть как просто отображение страницы, так и какие-то действия, например обновление или удаление данных с сервера.
3. Запрос направляется напрямую на сервер либо через прокси. Движение запроса регулируется другими протоколами, которые управляют отправкой данных по сети. Обычно это TCP/IP. Они формируют пакеты данных из запросов по своим правилам.
   Пример:

Здесь четыре элемента: метод — «GET», URI — «/», версия HTTP — «1.1» и адрес хоста. Давайте разберём каждый из них подробнее.
Метод — это действие, которое клиент ждёт от сервера. Например, отправить ему HTML-страницу сайта или скачать документ. Протокол HTTP не ограничивает количество разных методов
URI расшифровывается как «унифицированный идентификатор ресурса» (или Uniform Resource Identifier) — это полный адрес сайта в Сети. Он состоит из двух частей: URL и URN. Первое — это адрес хоста. Например, www.vk.com. Второе — это то, что ставится после URL и символа / — например, для URI www.vk.ru/media URN-адресом будет /media. URN ещё можно назвать адресом до конкретного файла на сайте.
Версия HTTP указывает, какую версию HTTP браузер использует при отправке запроса. Если её не указывать, по умолчанию будет стоять версия 1.1. Она нужна, чтобы сервер вернул HTTP-ответ с той же версией HTTP-протокола и не создал ошибок с чтением у клиента.
Адрес хоста нужен, чтобы указать, с какого сайта клиент пытается получить данные. Адрес указывают в виде домена, но он сразу же меняется на IP-адрес перед отправкой запроса с помощью DNS.

4. Сервер получает запрос, обрабатывает и формирует ответ. Он также заполняет стартовую строку и заголовок, а в тело поместит то, что клиент запросил. Обычно это веб-страница в формате HTML, которую браузер сможет расшифровать и отобразить. Одной из зон ответственности веб-разработчика является формирование и генерация этого HTML-кода — при помощи различных подходов и средств, в частности библиотек, движков-генераторов и фреймворков