Санкт-Петербургский политехнический университет Петра Великого
Институт компьютерных наук и технологий
Высшая школа программной инженерии

Компьютерные сети и телекоммуникации.

Выполнил
студент гр. в3530904/00030 В.С. Баганов

Руководитель
доцент, к.т.н. Б.М. Медведев

«____»_____ 202__ г.

Санкт-Петербург
2023

Содержание

1. Общие сведения о компьютерных сетях

1.1. Классификация сетей

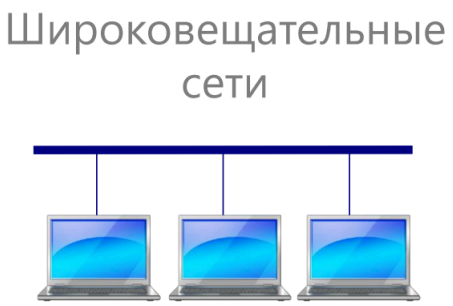
1.1.1. Классификация сетей

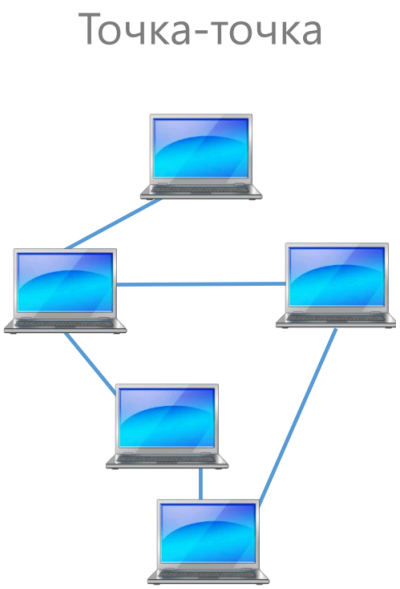
Тип коммутации
Технологии передачи
Протяженность

1.1.2. Тип коммутации

Коммутация каналов
Коммутация пакетов

1.1.3. Технология передачи





1.1.4. Протяженность

Название	Протяженность	Расположение
Персональная	1 м	На столе
Локальная	10 м – 1 км	Комната, здание, кампус
Муниципальная	10 км	Город
Глобальная	100 – 1000 км	Страна, континент
Объединение сетей	10 000 км	Весь мир

1.1.5. Сети с коммутацией пакетов

- Компьютерные сети

1.1.6. Технологии передачи данных

- Широковещательные (классический Ethernet, Wi-Fi)
- Точка-точка (коммутируемый Ethernet)

1.1.7. Протяженность

- Локальные сети (Ethernet, Wi-Fi)
- Объединение сетей (стек протоколов TCP/IP)

1.2. Топологии компьютерных сетей

Топология - это «схема» соединения компьютеров в сети.
Более строго, топология сети – конфигурация графа:
- Вершины – узлы сети (компьютеры и сетевое оборудование)
- Ребра – связи между узлами (физические или информационные)

1.2.1. Популярные базовые топологии компьютерных сетей:

- Полносвязная
- Ячеистая
- Кольцо
- Дерево
- Общая шина

На практике чаще всего используется смешанная топология, при которой разные сегменты сети используют разные базовые топологии.

1.2.2. Типы топологии:

- Физическая топология – соединения устройств в сети.
- Логическая топология – правила распространения сигналов в сети.

1.2.3. Логическая и физическая топологии могут отличаться:

- Классический Ethernet: физическая топология звезда, а логическая - общая шина.
- Коммутируемый Ethernet: физическая топология звезда, а логическая топология полносвязная.
- Wi-Fi: проводных сеодинений между компьютерами нет, а логическая топология - общая шина.

1.3. Стандарты компьютерных сетей

1.3.1. Стандарты в области компьютерных сетей.

1.3.2. Типы стандартов.

1.3.3. Организации, принимающие стандарты.

1.3.4. Стандарты IEEE.

1.3.5. Документы RFC.

1.3.6. Рекомендации W3C.

1.4. Основы организации компьютерных сетей

Создание крупных компьютерных сетей – сложная задача. Для ее решения используют метод декомпозиции: разбиение крупной сложной задачи на несколько более простых. В компьютерных сетях используется композиция на уровни сети.

Базовые понятия организации компьютерных сетей:

Сервис – описывает какие функции реализует уровень (что делает уровень).

Протокол – правила и соглашения, используемые для связи уровня N одного компьютера с уровнем N другого компьютера (как уровень это делает).

Интерфейс – набор примитивных операций, которые нижний уровень предоставляет верхнему (как получить доступ к уровню).

Архитектура компьютерной сети – набор уровней и протоколов сети.

Стек протоколов - Иерархически организованный набор протоколов, достаточный для организации взаимодействия по сети.

Инкапсуляция - Включение сообщения вышестоящего уровня в сообщение нижестоящего уровня.

Сообщение в компьютерных сетях состоит из трех частей: заголовок + данные + концевик (не обязателен).

Можно придумать много разных архитектур сетей и стеков протоколов, но, если мы хотим строить большие сети, которые можно объединять между собой, необходимо иметь стандарты на уровни сети, функции уровней, и протоколы, которые на них используются. Такие стандарты называются эталонными моделями компьютерных сетей. Сейчас популярны две эталонные модели:

- Модель взаимодействия открытых систем.

- Модель TCP/IP.

1.5. Модель OSI

На практике модель OSI не используется, но с ее помощью удобно описывать, как должны быть устроены компьютерные сети.

Модель взаимодействия открытых систем (Open Systems Interconnection) - это одна из двух популярных эталонных моделей организации компьютерных сетей. Модель является юридическим стандартом, принятым Международной организацией по стандартизации (ISO) в 1983 г.

Модель включает 7 уровней:

1. Физический.
2. Канальный.
3. Сетевой.
4. Транспортный.
5. Сеансовый.
6. Представления.
7. Прикладной.

1.6. Модель и стек протоколов TCP/IP

Стек протоколов TCP/IP – наиболее популярный набор сетевых протоколов в настоящее время. Он является основой интернет и широко используется.

Модель TCP/IP – одна из двух эталонных моделей организации сетей, которые популярны в настоящее время. Это фактический (de facto) стандарт на основе стека протоколов TCP/IP. Модель TCP/IP описывает, как нужно строить сети на основе разных технологий, чтобы в них работал стек TCP/IP.

Модель содержит 4 уровня:

- Сетевых интерфейсов
- Интернет
- Транспортный
- Прикладной

По назначению уровни похожи на уровни модели взаимодействия открытых систем ISO OSI. Уровень сетевых интерфейсов обеспечивает интеграцию стека TCP/IP с существующими сетевыми технологиями Ethernet, Wi-Fi и др. Уровень интернет (аналог сетевого уровня OSI) нужен для объединения сетей, построенных на основе разных сетевых технологий, и поиска маршрута в крупной составной сети. Транспортный уровень, как и в модели OSI, обеспечивает связь между процессами на разных компьютерах сети.

В прикладном уровне модели TCP/IP сочетаются функции уровней сеансового, представления и прикладного модели OSI. Считается, что если приложению TCP/IP нужны возможности сеансового уровня, или уровня представления, то оно должно самостоятельно их реализовать.

2. Физический уровень

2.1. Физический уровень

Физический уровень – первый уровень модели взаимодействия открытых систем (ISO OSI). Задача физического уровня: передача потока бит по среде передачи данных. Физический уровень не вникает в смысл передаваемой информации.

Единица передачи информации на физическом уровне – бит.

Основной вопрос, который решается на физическом уровне: как представить биты информации в виде сигналов, передаваемых по среде. Основная сложность в том, что при передаче по среде происходят искажения сигналов. Сигнал, который доходит до получателя, часто очень сильно отличается от сигнала, который передал отправитель. Поэтому важно выбрать правильную форму представления, чтобы можно было распознать сигнал без ошибок.

Мы воспользуемся преимуществом модели открытых систем: изоляция решений на разных уровнях. Будем считать, что физический уровень передает поток бит, а как он это делает, нам не важно.

2.1.1. Канал связи имеет следующие характеристики:

- Пропускная способность (бит/с) – сколько данных можно передать за единицу времен
- Задержка – сколько времени проходит от отправки сигнала до его получения.
- Количество ошибок – как часто в канале связи возникают ошибки.

2.1.2. Типы каналов связи по направлению передачи данных:

- Симплексный – данные можно передавать только в одном направлении
- Дуплексный – возможна передача данных в двух направлениях одновременно
- Полудуплексный – данные можно передавать в двух направлениях, но по очереди.

2.1.3. Среды передачи данных:

1. Кабель
- Телефонный кабель (“лапша”)
- Коаксиальный кабель
- Витая пара
- Оптический кабель
- Провода электропитания 220В
2. Беспроводные технологии
- Радиоволны
- Инфракрасное излучение
3. Спутниковые каналы
4. Беспроводная оптика (лазеры)

2.1.4. Для представления сигналов можно использовать два подхода:

- Цифровые сигналы. Метод представления сигналов: кодирование. Используется в медных кабелях.

- Аналоговые сигналы. Метод представления сигнала: модуляция. Используется в беспроводной среде и в оптоволокне.

3. Канальный уровень

3.1. Канальный уровень

3.1.1. Основные задачи канального уровня:

1. Передача сообщений по каналам связи – кадров (frame). Определение начала/конца кадра в потоке бит
2. Обнаружение и коррекция ошибок
3. Множественный доступ к каналу связи:
- Адресация
- Согласованный доступ к каналу связи

3.1.2. Методы выделения кадров:

- Указатель количества байт
- Вставка байтов (byte stuffing)
- Вставка битов (bit stuffing)
- Средства физического уровня

3.1.3. Обнаружение и исправление ошибок:

1. Обнаружение ошибок
- Контрольная сумма
2. Исправление ошибок
- Коды исправляющие ошибки (с избыточной информацией)
- Позволяют обнаруживать и исправлять ошибки
3. Повторная отправка данных
- Если в кадре обнаружена ошибка, его можно отправить заново.
- Повторная отправка кадра, который не дошел до получателя

3.1.4. Типы повторной отправки:

1. Остановка и ожидание.
2. Скользящее окно.
Канальный уровень в модели OSI состоит из двух подуровней:
1. Подуровень управления логическим каналом (LLC)
- Отвечает за передачу данных (создание кадров, обработка ошибок и т.д.)
- Общий для разных технологий
2. Подуровень управления доступом к среде (MAC):
- Совместное использование разделяемой среды
- Адресация
- Специфичный для разных технологий
- Не является обязательным

3.1.5. Технологии канального уровня:

- Ethernet, Wi-Fi (современные)
- Token Ring, FDDI, ATM, 100VG-AnyLAN (устаревшие)

3.2. Технология Ethernet

Ethernet - самая популярная сейчас технология создания локальных сетей.
Рассматривается история развития Ethernet, различные варианты технологии:
- Ethernet
- Fast Ethernet
- Gigabit Ethernet
- 5G Ethernet
- 10G Ethernet
- 100G Ethernet

Рассматриваются отличия классического и коммутируемого Ethernet.
Описаны варианты создания сети классического Ethernet:
- Общая шина в виде коаксиального кабеля.
- На основе концентраторов (hub).

Рассматривается формат кадра Ethernet.

3.3. MAC адреса

Служат для идентификации сетевых интерфейсов узлов сети.
MAC-адреса применяются в технологиях канального уровня:
- Ethernet (IEEE 802.3)
- Wi-Fi (IEEE 802.11)

Регламентированы стандартом IEEE 802. Длина 6 байт (48 бит)
Форма записи – шесть шестнадцатеричных чисел:
1C-75-08-D2-49-45

3.4. Ethernet. Метод доступа к разделяемой среде CSMA/CD

Классический Ethernet использует разделяемую среду передачи данных. В такой среде данные в одно время может передавать только один компьютер, иначе возникнет коллизия и данные исказятся.

Классический Ethernet использует метод доступа к среде CSMA/CD:

- Carrier Sense Multiple Access with Collision Detection.
- Множественный доступ с прослушиванием несущей частоты и обнаружением коллизий.

3.5. Коммутаторы Ethernet

Коммутируемый Ethernet - совершенно другая сетевая технология по сравнению с классическим Ethernet. Основное отличие: отказ от разделяемой среды передачи данных. Поэтому коллизии не возникают.

Для коммутируемого Ethernet необходимо новое устройство: коммутатор (switch).

Коммутатор использует полносвязную топологию.

Коммутатор работает на канальном уровне: он анализирует заголовок канального уровня и передает данные только компьютеру получателя, а не всем компьютерам в сети, как это делает концентратор.

3.5.1. Основные особенности работы коммутаторов:

1. Таблица коммутации - Соответствие MAC-адресов портам коммутатора.
2. Алгоритм обратного обучения - Заполнение таблицы коммутации.

3. Алгоритм прозрачного моста - Передача кадров коммутатором.

3.5.2. Преимущества коммутаторов:

- высокая производительность и масштабируемость
- высокая безопасность.

3.6. VLAN

VLAN (Virtual Local Area Network, Виртуальная локальная сеть) - технология разделения единой сети на несколько логических сетей, изолированных друг от друга.

В модели OSI и TCP/IP технология VLAN находится на канальном уровне и реализуется на коммутаторах.

3.6.1. Преимущества изоляции локальных сетей:

- Безопасность.
- Распределение нагрузки.
- Ограничение широковещательного трафика.

Для описания VLAN используются номера. Компьютеры, подключенные к VLAN с разными номерами, не могут передавать данные друг другу.

3.6.2. Типы VLAN:

- На основе коммутаторов (нетегированные). Номер VLAN задается в таблице коммутации. Работает только внутри одного коммутатора.
- Тегированные - номер VLAN передается в кадре Ethernet (стандарт IEEE 802.1Q). Может использоваться в сети с несколькими коммутаторами.

3.7. Протокол STP

Протокол связующего (остовного) дерева (Spanning Tree Protocol, STP) позволяет автоматически отключать дублирующие соединений в Ethernet, чтобы в сети не образовалось кольца и широковещательного шторма.

Протокол STP определен в стандарте IEEE 802.1D.

3.7.1. Преимущества STP:

- Надежность соединений между коммутаторами.
- Защита от ошибок конфигурации.

Протокол STP работает в 3 этапа:

1. Выбор корневого коммутатора
2. Определение кратчайших путей до корневого коммутатора
3. Отключение всех остальных соединений

Для реализация STP коммутаторы каждые 2 секунды отправляют управляющие сообщения Bridge Protocol Data Units (BPDU) на групповой адрес STP 01:80:C2:00:00:00.

3.7.2. Расширения протокола STP.

RSTP (Rapid Spanning Tree Protocol):

- Срабатывает быстрее при подключении оборудования и изменении конфигурации сети
- Стандарт IEEE 802.1w

STP и VLAN:

- Multiple Spanning Tree Protocol (MSTP), 802.1s
- Отдельное связующее дерево для каждого VLAN

3.8. Wi-Fi

Wi-Fi – торговая марка, принадлежит компании Wi-Fi Alliance.

Технология Wi-Fi описана в стандартах серии IEEE 802.11.

В модели OSI Wi-Fi находится на канальном уровне.

Wi-Fi может работать в двух режимах:

- инфраструктурном
- одноранговом.

3.8.1. Стандарты Wi-Fi

Название Год принятия Скорость Частота

Wi-Fi использует для передачи данных электромагнитное излучение:

2,4 ГГц – 802.11, 802.11b, 802.11g, 802.11n

5 ГГц – 802.11a, 802.11n, 802.11ac

Современные стандарты Wi-Fi используют метод

Orthogonal Frequency Division Multiplexing (OFMD, мультиплексирование с ортогональным частотным разделением). При этом данные передаются параллельно на разных частотах.

В диапазоне 2,4 ГГц используется 14 каналов передачи данных с разной частотой.

Wi-Fi может использовать каналы разной ширины: 20, 40, 80 или 160 МГц.

Wi-Fi позволяет менять скорость при разном качестве сигнала за счет изменения метода модуляции, ширины канала, и времени между передачей отдельных символов (Guard Interval).

3.9. Wi-Fi. Метод доступа к разделяемой среде CSMA/CA

Wi-Fi использует для передачи данных разделяемую среду - радиоэфир. В разделяемой среде возможны коллизии. Чтобы их избежать, необходим метод доступа к среде, который бы обеспечивал, что в один момент времени данные передает только один компьютер.

3.9.1. Особенности беспроводной среды:

- Вероятность ошибки передачи выше, чем в проводной среде
- Мощность передаваемого сигнала намного выше, чем принимаемого
- Ограниченный диапазон распространения сигнала – не все компьютеры в сети получают данные (проблема скрытой станции и проблема засвеченной станции).

Так как ошибки при передаче данных возникают часто, то в Wi-Fi на канальном уровне используется подтверждение доставки.

Коллизии в Wi-Fi обнаруживаются по отсутствию подтверждений.

Коллизия в Wi-Fi обходится очень дорого, т.к. требуют больших временных затрат на обнаружение: время передачи кадра и тайм-аут ожидания подтверждения.

Поэтому в Wi-Fi используется метод CSMA/CA - ножественный доступ с прослушиванием несущей частоты с предотвращением коллизий.

Другой метод доступа к среда в Wi-Fi: протокол Multiple Access with Collision Avoidance (MACA). Он позволяет решить проблему скрытой и засвеченной станции. Однако на практике метода CSMA/CA почти всегда достаточно, поэтому поддержка протокола MACA в оборудовании Wi-Fi не обязательна.

В MACA перед передачей данный компьютер отправляет короткое сообщение Request To Send (RTS). Принимающий компьютер в ответ передает сообщение Clear To Send (CTS). После этого отправитель может передавать данные.

3.10. Wi-Fi. Формат кадра

Wi-Fi использует два формата кадра. При передаче по беспроводной среде на уровне MAC используется формат кадра IEEE 802.11. Однако при поступлении на компьютер на уровне LLC происходит преобразование в формат кадра Ethernet.

Формат кадра IEEE 802.11 сильно отличается от формата Ethernet. Во-первых, в кадре Wi-Fi используется 4 MAC-адреса следующего назначения:

Адрес 1 - Адрес устройства, которое принимает данные из беспроводной среды (Receiver Address)

Адрес 2 - Адрес устройства, которое передает данные в беспроводную среду (Transmitter address).

Адрес 3 - Адрес устройства получателя (Destination Address).

Адрес 4 - Адрес устройства отправителя (Source Address).

Размер поля данных в Wi-Fi 2304 байта (в Ethernet размер поля данных 1500 байт).

Поле "Управление кадром" содержит большое количество управляющих флагов.

3.10.1. Типы кадров Wi-Fi:

- Кадры данных (передача данных)
- Кадры контроля (control frames). Служебные кадры (RTS, CTS, ACK)
- Кадры управления (management frames). Реализация сервисов Wi-Fi (например, ассоциация с точкой доступа).

Технология IEEE 802.11 Power Saving Mode позволяет экономить питание при работе Wi-Fi.

Фрагментация в Wi-Fi позволяет передавать данные, даже если ошибки возникают очень часто. При этом кадры делятся на отдельные части (фрагменты) и передаются отдельно.

3.11. Сервисы Wi-Fi

3.11.1. Наборы сервисов Wi-Fi:

- Базовый набор сервисов (basic service set).
- Расширенный набор сервисов (extended service set).

3.11.2. Сервисы Wi-Fi:

- Ассоциация
- Аутентификация

- Передача данных
- Защита информации (шифрование)
- Роуминг

3.11.3. Идентификаторы наборов сервисов Wi-Fi:

- BSSID - идентификатор базового набора сервисов, MAC-адрес точки доступа.
- SSID - идентификатор набора сервисов в понятном для человека виде, текстовая строка.

4. Сетевой уровень

4.1. Сетевой уровень

Сетевой уровень (network layer) объединяет сети, построенные на основе разных технологий канального уровня:

- Ethernet
- Wi-Fi
- 5G/4G/3G
- MPLS
- ATM, TokenRing, FDDI (устаревшие)

Сетевой уровень - основа Интерне. Винтона Серфа и Роберта Кана, предложивших идею сетевого уровня, называют "Отцами Интернета".

4.1.1. Задачи сетевого уровня:

- Объединение сетей, построенных на основе разных технологий канального уровня (internetworking).
- Маршрутизация - поиск маршрута пакета в крупной составной сети.
- Обеспечение качества обслуживания

Рассматриваются ограничения Ethernet, не позволяющие построить на его основе крупную глобальную сеть.

Устройство сетевого уровня - маршрутизатор.

4.1.2. Протоколы сетевого уровня стека TCP/IP

- IP
- ICMP
- ARP
- DHCP

4.2. IP-адреса

IP-адреса - это глобальные адреса, используемые в стеке протоколов TCP/IP. Широко используются в Интернет.

Используются для уникальной идентификации компьютеров в составной сети.

4.2.1. Две версии протокола IP:

- IPv4: адрес 4 байта
- IPv6: адрес 16 байт

Длина адреса IPv4 – 4 байта, 32 бита.

4.2.2. Форма представления:

4 десятичных числа 0-255, разделенных точками.

Подсеть (IP-сеть, сеть, subnet) – множество компьютеров, у которых старшая часть IP-адреса одинаковая.

4.2.3. Структура IP-адреса:

- Номер подсети – старшие биты
- Номер хоста (компьютера в сети) – младшие биты

Маска подсети показывает, где в IP-адресе номер сети, а где хоста. Структура маски:
- Длина 32 бита.
- Единицы в позициях, задающих номер сети.
- Нули в позициях, задающих номер хоста.

4.2.4. Формы записи маски:

- Десятичное
- В виде префикса

4.2.5. Типы IP-адресов:

- Индивидуальный
- Групповой
- Широковещательный

Рассматриваются правила образования широковещательного адреса и два типа широковещания в IP:
- Направленное широковещание.
- Ограниченное широковещание.

Все IP-адреса в интернет должны быть уникальными. Поэтому нельзя использовать любые адреса, нужно получать разрешение у Internet Assigned Numbers Authority. Сейчас эту роль выполняет корпорация ICANN (Internet Corporation for Assigned Names and Numbers).

Без обращения в ICANN можно использовать IP-адреса из диапазонов частных адресов:
- 10.0.0.0 /8
- 172.16.0.0 /12
- 192.168.0.0 /16

Эти адреса не маршрутизируются в интернет.

Рассматривается проблема исчерпания IP-адресов. Максимальное количество адресов IPv4 четь больше 4 млрд. Этого было достаточно во времена разработки протокола IP, но не сейчас. В настоящее время почти все адреса IPv4 распределены и получить новый не реально. Решение проблем исчерпания IP-адресов:
- Переход на IPv6.
- Использование технологии NAT

4.3. Протокол IP

4.3.1. Задачи IP:

объединение сетей
маршрутизация
обеспечения качества обслуживания.
Формат заголовка IP.
RFC 791 - Internet Protocol
https://tools.ietf.org/html/rfc791

4.4. Протокол IP: маршрутизация

Маршрутизация – поиск маршрута доставки пакета между сетями через транзитные узлы – маршрутизаторы

4.4.1. Этапы маршрутизации:

- Изучение сети
- Продвижение пакетов на маршрутизаторе

4.4.2. Столбцы таблицы маршрутизации:

- Адрес сети
- Маска подсети
- Шлюз
- Интерфейс
- Метрика.

Маршрутизатор по умолчанию - маршрутизатор, на который отправляются пакеты, для которых неизвестен маршрут доставки.

4.5. Протокол IP: фрагментация

Фрагментация в протоколе IP.

Поля заголовка IP, используемые для фрагментации: идентификатор пакета, флаги (DF и MF), смещение фрагмента.

Сборка пакета из фрагментов.

Запрет фрагментации с помощью флага DF.

4.6. Управляющие протоколы сетевого уровня

Протокол IP используется для передачи данных на сетевом уровне. Но для корректной работы крупной составной сети одного протокола IP недостаточно.

Для обеспечения работы крупной сети используются управляющий протоколы сетевого уровня:

- DHCP - для автоматического назначения IP адресов.
- ARP - для связи сетевого уровня с канальным (автоматического определения MAC-адреса компьютера по его IP-адресу).
- ICMP - для передачи сообщений об ошибках и диагностики работы сети

4.7. Протокол DHCP

Протокол DHCP позволяет назначать IP-адреса компьютерам в сети автоматически.

Протокол использует архитектуру клиент-сервер, которые обмениваются сообщениями в режиме запрос-ответ.

4.7.1. Сообщения DHCP:

- DISCOVER - Поиск DHCP сервера
- OFFER - Предложение IP-адреса DHCP сервером клиенту
- REQUEST - Запрос IP-адреса DHCP клиентом
- ACK - Подтверждение назначения IP-адреса DHCP клиенту
- NACK - Запрет использования запрошенного DHCP клиентом IP-адреса

- RELEASE - Освобождение IP-адреса
- INFORM - Запрос и передача дополнительной конфигурационной информации

Для получения IP-адреса используются следующие сообщения: DISCOVER, OFFER, REQUEST, ACK (сокращенно DORA).

DHCP сервер выдает IP-адрес DHCP клиенту на ограниченное время, которое называется время аренды (lease time). После его окончания IP-адрес освобождается, но DHCP клиент может продлить использование IP-адреса при необходимости.

Кроме IP-адреса, по DHCP также назначаются дополнительные параметры конфигурации сети. Для передачи этих параметров служат DHCP опции. Примеры часто используемых опций:

- Маска подсети
- Маршрутизатор по умолчанию
- Адреса DNS-серверов
- Имя домена
- Адреса серверов времени
- Маршруты

4.8. Протокол ARP

Address Resolution Protocol (ARP) – протокол разрешения адресов.

ARP позволяет определить MAC-адрес компьютера по его IP-адресу. Формат ARP-запроса и ARP-ответа. ARP-таблица. Статические и динамические записи в ARP-таблице. Gratuitous ARP.

RFC 826 - An Ethernet Address Resolution Protocol https://tools.ietf.org/html/rfc826

4.9. Протокол ICMP

Протокол ICMP.

4.9.1. Формат ICMP-пакета.

4.9.2. Тип и код сообщения.

4.9.3. Примеры популярных типов и кодов сообщений.

Утилита ping - проверка доступности компьютера в сети.

Утилита traceroute - определение маршрута к получателю.

4.9.4. Типы и коды сообщений ICMP:

http://www.iana.org/assignments/icmp-parameters/icmp-parameters.xhtml

4.10. Передача пакетов на сетевом и канальном уровнях

Канальный уровень используется для передачи данных в одном сегменте сети, а сетевой - для объединение сетей в одну крупную составную сеть.

В каждом пакете содержится 2 адреса:

IP (сетевого уровня) и MAC (канального уровня).

Рассматриваются различные варианты передачи пакетов: в рамках одной сети и через маршрутизатор.

На сетевом уровене IP-адреса в пакете сохраняются постоянными.

На канальный уровень MAC-адреса постоянно меняются. Канальный уровень работает по принципу звеньев цепи. Если получатель находится в другой сети, и нет возможности передать данные ему напрямую, то в качестве MAC-адреса получателя указывается MAC-адрес маршрутизатора.

Дополнительные средства, которые используются при передаче пакетов:

- Протокол ARP
- Маска подсети
- Таблица маршрутизации

5. Транспортный уровень

5.1. Транспортный уровень

5.1.1. Транспортный уровень модели взаимодействия открытых систем.

Порты.
Протоколы TCP и UDP.
Интерфейс сокетов.

5.2. Протокол UDP

5.2.1. Протокол UDP.

5.2.2. Особенности UDP.

5.2.3. Формат заголовка UDP.

5.3. Протокол TCP

. TCP (Transmission Control Protocol, протокол управления передачей) – протокол транспортного уровня стека TCP/IP. Он предоставляет сервис надежной передача потока байт (reliable byte stream). TCP предоставляет следующие гарантии:

- Доставка данных.
- Сохранения порядка следования сообщений.

Транспортная подсистема получает от приложения данные в виде потока байт. Поток разбивается на отдельные части, которые называются сегменты. Сегменты передаются от отправителя к получателю независимо друг от друга. Получатель собирает сегменты и передает принимающему приложению поток байт.

Для гарантии доставки TCP использует подтверждение получения данных. Получатель, после приема очередной порции данных, передает отправителю подтверждения о получении. В случае, если подтверждение не пришло, отправитель передает данные еще раз.

В TCP подтверждается не получение каждого сегмента, а получение нескольких сегментов. Это сделано для увеличения скорости передачи данных: отправитель может передать без остановки несколько сегментов, не дожидаясь прихода подтверждения. Такой тип подтверждения называется кумулятивный: подтверждается получение последнего сегмента, и всех предыдущих. Количество сегментов, которые отправитель может передать без подтверждения, называется размер скользящего окна.

Однако только подтверждения и повторной отправки недостаточно для надежной передачи потока байт. В дополнение к потере данных возможна и другая проблема: нарушение порядка следования сообщений:

- Сегменты приходят в неправильном порядке.
- Сегменты дублируются.

Для сохранения порядка следования сообщений используется нумерация сообщений. Особенность протокола TCP в том, что он нумерует не сегменты, а байты в сегментах. Нумерация сообщений позволяет расставить перепутанные сегменты в правильном порядке, а также не учитывать дублирующийся сегменты.

Перед отправкой данных по TCP необходимо установить соединение.

5.3.1. Задачи соединения:

- Убедиться, что отправитель и получатель хотят передавать данные друг другу.
- Договориться о нумерации потока байт.
- Договорится о параметрах соединения (максимальный размер сегмента и т.п.).

После завершения передачи данных соединение TCP разрывается.

5.4. Протокол TCP: скользящее окно

5.4.1. Варианты подтверждения передачи сообщений:

остановка и ожидание, скользящее окно.
Влияние скользящего окна на скорость передачи данных на транспортном уровне.
Размер окна.
Кумулятивное и выборочное подтверждение.

5.5. Протокол TCP: соединение

5.5.1. Установка соединения в TCP.

5.5.2. Трехкратное рукопожатие (SYN, SYN + ACK, ACK).

5.5.3. Разрыв соединения: FIN, RST.

5.6. Протокол TCP: формат заголовка

Описание формата заголовка протокола TCP.

5.7. Протокол TCP: управление потоком

Управление потоком в TCP.
Быстрый отправитель и медленный получатель.
Затопление. Буфер получатель на транспортном уровне.
Поле "Размер окна" в заголовке TCP.
Zero Window Probe.

5.8. Протокол TCP: управление перегрузкой

Перегрузки в сети.
Окно управления потоком и окно перегрузки.
Скорость передачи данных в TCP.
Аддитивное увеличение мультипликативное уменьшение (AIMD) и медленный старт.
Сигналы о перегрузке: задержка сегмента, сообщение от маршрутизатора.
Технологии Random Early Detection и Explicit Congestion Notification.

5.9. Интерфейс сокетов

Сокеты - это интерфейс для взаимодействия с транспортным уровнем. В отличие от протоколов транспортного уровня TCP и UDP, которые используются для связи между транспортными уровнями разных хостов, интерфейс сокетов используется для взаимодействия приложения с транспортным уровнем внутри одного компьютера.

Интерфейс сокетов был впервые предложен в Berkeley UNIX 4.2 BSD. Это файл специального вида, при записи данных в которой они передаются по сети.

Сокеты оказались удобным интерфейсом, поэтому различные варианты сокетов реализованы в разных операционных системах (в том числе в Windows и Linux) и языках программирования.

5.9.1. Операции сокетов:

socket - создание новыго сокета
bind - установка связи сокета с IP-адресом и портом
listen - объявление о желании принимать соединения
accept - прием запроса на установку соединения
connect - установка соединения
send - отправка данные по сети
receive - получение данные из сети
close - закрытие соединения
Рассматривается пример использования сокетов на Python.

Программист взаимодействует с транспортным уровнем через интерфейс сокетов, поэтому протоколы TCP и UDP скрыты от программиста. Следовательно, при изменении протоколов транспортного уровня программу менять не придется.

Сокеты в Python - https://docs.python.org/3/library/socket.html

5.10. Протоколы, интерфейсы и сервисы. Примеры

Что такое сервисы, интерфейсы и протоколы на примере транспортного уровня TCP/IP.

Сервис – описывает какие функции реализует уровень

Интерфейс – набор примитивных операций, которые нижний уровень предоставляет верхнему

Протокол – правила и соглашения, используемые для связи уровня N одного компьютера с уровнем N другого компьютера

5.10.1. Примеры для транспортного уровня:

- Протоколы: TCP и UDP.
- Интерфейс: сокеты.
- Сервисы: надежная передача потока байт и ненадежная передача коротких сообщений.

Разделять протоколы и интерфейсы необходимо для изоляции решений. Это общий принцип проектирования в ИТ: описание и реализация должны быть отделены друг от друга.

Сервис - это абстрактное описание того, что делает уровень. Оно позволяет сформулировать, что требуется от уровня, не вдаваясь в детали реализации.

Интерфейс - это конкретное описание методов, процедур и функций, которые нужно вызвать, чтобы получить доступ к сервису. На транспортном уровне через один интерфейс сокетов можно получить доступ к двум типам сервиса. Для этого при создании сокета нужно указать разные константы (в Python socket.SOCK_STREAM или socket.SOCK_DGRAM).

В TCP/IP отсутствует сервис надежной доставки коротких сообщений. Многим приложениям, например, службе имен DNS, приходится самим реализовывать эту функциональность.

5.11. Трансляция сетевых адресов (NAT)

NAT - технология преобразования IP-адресов внутренней (частной) сети в IP-адреса внешней сети (Интернет).

Цель создания – преодоление нехватки адресов IPv4.

5.11.1. Типы NAT:

Статический NAT - отображение один к одному.

Динамический NAT - отображение внутренних адресов на группу внешних адресов.

Один ко многим (masquerading) - отображение внутренних адресов на один внешний адрес.

5.11.2. Преимущества NAT:

- Позволяет преодолеть нехватку адресов IPv4.
- Легко развернуть и использовать.
- Скрывает структуру сети от внешнего мира.

5.11.3. Недостатки NAT:

- Нарушение фундаментального принципа построения IP-сетей: каждый компьютер может соединиться с любым другим.
- Нет возможности подключиться к компьютерам во внутренней сети из внешнего мира.
- Плохо работают протоколы не устанавливающие соединения.
- Некоторые прикладные протоколы работают неправильно (FTP).
- Нет единого стандарта NAT, много разных вариантов.

5.12. Межсетевые экраны

Межсетевой экран - это устройств или программная система, предназначенное для отделения сетей друг от друга. Другое название межсетевого экрана: брандмауэр или firewall.

Межсетевые экраны используются для защиты сетей от проникновения злоумышленников.

Межсетевые экраны перехватывают все пакеты, которые поступают в сети, и проверяют их на соответствие политике безопасности по таблице правил.

Межсетевые экраны работают на сетевом и транспортном уровне, они анализируют заголовки пакетов протоколов этих уровней. Чаще всего используются IP-адреса, порты транспортного уровня, флаги, а также состояние соединение.

Преимуществом использования межсетевых экранов является увеличение безопасности.

5.12.1. Недостатки межсетевых экранов:

- Неправильная конфигурация может привести к неработоспособности сети.
- Возможно снижение производительности сети.

6. Прикладной уровень

6.1. Прикладной уровень

Прикладной уровень - самый верхний уровень моделей OSI и TCP/IP. Протоколы прикладного уровня используются для свези между различными сетевыми приложениями.

В стек TCP/IP входят следующие протоколы прикладного уровня:

- протокол передачи Web-страниц HTTP (Hypertext Transfer Protocol)
- протоколы электронной почты SMTP, POP3 и IMAP.
- протокол системы доменных имен DNS (Domain Name System)
- протокол передачи файлов FTP (File Transfer Protocol)
- и многие другие.

В стеке протоколов TCP/IP прикладной протокол должен реализовывать функции сеансового уровня и уровня представления модели OSI, если они ему нужны.

Протокол HTTP использует HTTP keep alive для загрузки различных элементов Web-страницы (гипертекста, стилевого файла, картинок и т.п.) через одно соединение TCP для увеличения производительности (функция сеансового уровня).

Протокол HTTPS использует технологии TLS/SSL для шифрования передаваемых по сети данных в целях безопасности (функция уровня представления).

Некоторые современные сетевые устройства работают на прикладном уровне. Например, контент-фильтр, который может ограничивать доступ к некоторым Web-страницам. Для этого контент-фильтр просматривает HTTP трафик, анализирует адреса, к которым обращается пользователь, и блокирует доступ к запрещенным адресам и доменам.

6.2. Система доменных имен DNS

Систем DNS предназначена для отображения символьных доменных имен компьютеров в IP-адреса.

6.2.1. Преимущества DNS:

- Понятные человеку имена
- Возможность менять сетевую инфраструктуру

6.3. Протокол DNS

Протокол DNS используется в системе доменных имен DNS.

Итеративный и рекурсивные режимы работы DNS.

Сервер разрешения имен DNS (DNS resolver).

Открытые серверы разрешения имен DNS.

Кэширование DNS ответов.

Типы ответов DNS: заслуживающий доверия (authoritative) и не заслуживающий доверия (non-authoritative).

Формат пакета DNS.

6.4. Типы записей DNS

A - адрес IPv4

AAAA - адрес IPv6

CNAME - псевдоним для доменного имени

MX - адрес почтового сервера
SRV - адреса и порты сетевых сервисов
NS - адреса DNS-серверов, ответственных за зону
PTR - доменное имя для IP-адреса

6.4.1. Типы DNS-зон:

Прямая - определение IP-адреса по доменному имени
Обратная - определение доменного имени по IP-адресу.
Домен для обратных зон in-addr.arpa.
Запрос записей разных типов:
nslookup -type=XXX yandex.ru

6.5. Протокол HTTP

Hypertext Transfer Protocol (HTTP) – протокол передачи гипертекста, основа World Wide Web Тим Бернерс-Ли в ЦЕРН предложил концепцию Web в 1989 году.

Uniform Resource Locator (URL) – уникальное положение ресурса.

В стеке TCP/IP протокол HTTP находится на прикладном уровне. Используется протокол транспортного уровня TCP, порт сервера 80. HTTP работает в режиме запрос-ответ. Данные передаются в текстовом виде.

6.5.1. Версии протокола HTTP:

HTTP 0.9 – экспериментальная версия ЦЕРН, 1991
HTTP 1 – первая официальная версия протокола, 1996
HTTP 1.1 – расширение первой версии HTTP, 1997. Кэширование, постоянное соединение, аутентификация. Используется сейчас
HTTP 2 – современная версия HTTP, 2015. Вводится в эксплуатацию

6.5.2. Структура пакета HTTP:

- Метод запроса/статус ответа
- Заголовки (не обязательно)
- Тело сообщения (не обязательно)

6.5.3. Методы HTTP:

GET – запрос Web-страницы
POST – передача данных на Web-сервер
HEAD – запрос заголовка страницы
PUT – помещение страницы на Web-сервер
DELETE – удаление страницы с Web-сервера
TRACE – трассировка страницы
OPTIONS – запрос поддерживаемых методов HTTP для ресурса
CONNECT – подключение к Web-серверу через прокси

6.5.4. Статусы HTTP:

1XX – информация
2XX – успешное выполнение (200 OK)
3XX – перенаправление (301 – постоянное перемещение, 307 – временное перенаправление)
4XX – Ошибка на стороне клиента (403 – доступ запрещен, 404 – страница не найдена)
5XX – Ошибка сервера (500 – внутренняя ошибка сервера)

6.6. Постоянное соединение в HTTP

Постоянное соединение HTTP (HTTP keep-alive, HTTP persistent connection) - использование одного TCP-соединения для загрузки нескольких ресурсов

6.6.1. Преимущества постоянного соединения:

- Сокращение накладных расходов на установку TCP-соединения.
- Нет необходимости каждый раз начинать передачу данных с маленьким размером окна TCP (медленный старт).

В стандарте HTTP 1.0 возможности использовать постоянное соединение нет. После публикации стандарта HTTP 1.0 был предложен заголовок "Connection: keep-alive".

Пример использования заголовка "Connection: keep-alive":
HTTP-запрос:
GET /courses/networks HTTP/1.0
Host: www.asozykin.ru
Connection: keep-alive
HTTP-ответ:
HTTP/1.0 200 OK
Server: nginx
Content-Type: text/html; charset=UTF-8
Content-Length: 5161
Connection: keep-alive

В стандарте HTTP 1.1 по умолчанию все соединения постоянные.

Технология конвейерной обработки HTTP (pipelining) - отправка сразу нескольких HTTP запросов через постоянное соединение не дожидаясь ответа. Сервер также передает сразу несколько запрошенных ресурсов.

Для ускорения загрузки Web-страниц браузеры открывают сразу несколько соединений с Web-сервером. Каждое такое соединение может быть постоянным и использовать конвейерную обработку.

6.7. Кэширование в HTTP

Современные Web-браузеры поддерживают кэширование редко меняющихся ресурсов, что позволяет сократить время загрузки web-страниц. Поддержка кэширования встроена в протокол HTTP.

Для определения, можно ли использовать копию ресурса из кэша, используется заголовок Expires, например:

Expires: Sun, 12 Jun 2016 10:35:18 GMT

Также можно использовать эвристику на основе заголовка Last-Modified.

Запрос GET с условием (Conditional GET) – запрос Web-серверу передать ресурс, если он изменился с указанного времени.

Ответы на запрос GET с условием:

- Ресурс не изменился: короткое сообщение со статусом 304 Not Modified.

- Ресурс изменился: полная передача измененной версии ресурса, статус ответа 200 OK.

Определение изменения ресурса в запросе Get с условием:

- По дате последнего изменения ресурса: заголовок If-Modified-Since (HTTP версия 1.0).

- По тэгу (хэшу) ресурса: заголовок If-None-Match (HTTP версия 1.1)

В HTTP версии 1.1 появился новый заголовок для управления кэшем Cache-Control. Возможные значения:

- no-store
- no-cache
- public
- private
- max-age=X (время в секундах)

Кроме кэша браузера, ресурсы могут быть сохранены на прокси-серверах и обратных прокси-серверах.

6.8. Электронная почта

Электронная почта (email) – технология передачи и получения электронных сообщений через компьютерную сеть.

6.8.1. Основные компоненты электронной почты:

Агент пользователя (MUA, Mail User Agent) - клиент электронной почты.

Агент передачи почты (MTA, Mail Transfer Agent) - программа передачи почты от клиента на сервер и между почтовыми серверами.

Агент доставки почты (MDA, Mail Delivery Agent) - программа записи сообщений в почтовые ящики получателей.

Адрес электронной почты состоит из двух частей: идентификатор пользователя и доменное имя. В качестве разделителя используется символ @ (коммерческая эт). Пример: networkcourse@mail.ru.

6.8.2. Основные протоколы электронной почты:

SMTP – Simple Mail Transfer Protocol

POP3 – Post Office Protocol 3

IMAP – Internet Message Access Protocol

Для определения почтового сервера получателя используется система DNS, запись типа MX

6.9. Протокол SMTP

SMTP (Simple Mail Transfer Protocol) – простой протокол передачи почты.

6.9.1. Версии SMTP:

- Протокол SMTP был создан в 1982 году.
- Расширение SMTP (ESMTP, Extended SMTP) - 2008 год.

В стеке TCP/IP протокол SMTP находится на прикладном уровне.

6.9.2. Порты SMTP:

Порт 25 – передача почты между почтовыми серверами
Порт 587 – прием почты от клиентов
Электронное письмо состоит из трех частей:
- Конверт.
- Заголовок.
- Тело письма.
SMTP работает в текстовом режиме, используется взаимодействие типа запрос-ответ.

6.9.3. Команды SMTP:

- HELO/EHLO - установка соединения
- MAIL FROM - адрес отправителя
- RCPT TO - адрес получателя
- DATA - начало передачи письма
- QUIT - выход

6.10. Протокол POP3

POP (Post Office Protocol) – протокол почтового отделения. Текущая версия протокола третья, первая и вторая считаются устаревшими.

Протокол POP3 используется для чтения сообщений из почтового ящика пользователя.

Протокол POP использует подход «загрузить и удалить». Почтовый ящик на сервере считается временным хранилищем сообщений. Все сообщения должны быть переписаны на почтовый клиент. После загрузки на клиент сообщение удаляется с сервера.

Протокол POP3 использует транспортный протокол TCP, порт 110.

POP3 работает в текстовом режиме.

6.10.1. Стадии сеанса POP3:

1. Авторизация. Клиент представляется и подтверждает, что он тот, за кого себя выдает.

2. Транзакция. Клиент загружает почту и помечает загруженные сообщения на удаление.

3. Обновление. Сервер удаляет помеченные сообщения и закрывает соединение.

6.10.2. Команды протокола POP3:

USER Указать имя пользователя
PASS Указать пароль
STAT Количество писем на сервере
LIST Передача информации о сообщениях
RETR Передать сообщение на клиент
TOP Передать на клиент заголовок сообщения
DELE Пометить сообщение на удаление
QUIT Закрытие транзакции, удаление сообщений и отключение

6.10.3. Статус ответов сервера:

+OK - команда выполнена успешно
-ERR - произошла ошибка.
Рассматривается пример сеанса POP3.

6.11. Протокол IMAP

IMAP (Internet Message Access Protocol) – протокол доступа к электронной почте.

Протокол IMAP, также как и POP3, используется для чтения сообщений из почтового ящика пользователя. В отличие от POP3, в IMAP сообщения хранятся на сервере и с почтовым ящиком может одновременно работать несколько почтовых клиентов.

IMAP использует транспортный протокол TCP, порт 143.

IMAP позволяет использовать несколько почтовых ящиков (mailbox) или папок, папка по умолчанию называется INBOX.

В IMAP письму можно установить флаги, которые бывают системные и пользовательские.

Системные флаги заданы в стандарте IMAP и начинаются с обратного слеша:

/Seen - сообщение прочитано
/Answered - отправлен ответ на сообщение
/Flagged - сообщение помечено как важное
/Draft - сообщение является черновиком
/Deleted - сообщение помечено на удаление
/Recent - сообщение новое.

Сеанс IMAP состоит из четырех этапов:

1. Клиент не аутентифицирован (Not Authenticated). Клиент только что подключился к серверу и должен пройти аутентификацию

2. Клиент аутентифицирован (Authenticated). Клиент успешно прошел аутентификацию.

3. Папка выбрана (Selected). Выбрана папка на сервере, с которой будет производиться работа.

4. Выход (Logout). Разрыв соединения.

Приводится пример сеанса IMAP.

6.12. Протокол FTP

FTP (File Transfer Protocol) – протокол передачи файлов.

FTP использует архитектуру клиент-сервер. На сервере находится файловая система. Клиент может подключаться к серверу и работать с файловой системой: загружать файлы, создавать и удалять каталоги, копировать и перемещать файлы между каталогами и т.п.

Протокол FTP использует URL для адресации файлов. Пример URL: `ftp://ftp-server.ru/pub/documents/latex/example1.tex`

На транспортном уровне используется протокол TCP.

В отличие от большинства протоколов прикладного уровня, FTP использует два соединения:

- Управляющее соединение
- Соединение для передачи данных.

6.12.1. Команды протокола FTP:

USER - Указать имя пользователя
PASS - Указать пароль
LIST - Просмотр содержимого каталога
CWD - Смена текущего каталога
RETR - Передать файл с сервера на клиент
STOR - Передать файл с клиента на сервер
TYPE - Установить режим передачи
DELE - Удалить файл
MKD - Создать каталог
RMD - Удалить каталог
PASV - Использовать пассивный режим
QUIT - Выход и разрыв соединения

FTP передает логин и пароль пользователя, а также все данные по сети в открытом виде, поэтому безопасность очень низкая. Вместо FTP сейчас используются более безопасные протоколы на основе SSH:

SFTP и SCP.

7. Защищенные сетевые протоколы

7.1. Введение в TLS/SSL

Рассматриваются основы работы безопасных сетевых протоколов транспортного уровня:

- TLS – Transport Layer Security, протокол защиты транспортного уровня
- SSL – Secure Sockets Layer, уровень защищенных сокетов (устарел)

7.2. Шифрование в TLS/SSL

7.3. Целостность данных в TLS/SSL

7.4. Инфраструктура открытых ключей в TLS/SSL

7.5. Протокол TLS

7.6. Установка соединения в TLS

7.7. Анализируем протокол TLS в Wireshark.

7.8. Расшифровка TLS в WireShark.

7.9. Протокол TLS 1.3

7.10. Протокол TLS 1.3 в WireShark.

7.11. Протокол HTTPS

7.12. Протокол HTTPS в WireShark.

8. Продвинутые темы

8.1. Web сокеты

Web сокеты - это протокол прикладного уровня стека TCP/IP, предназначенный для создания Web приложений реального времени.

Основное отличие Web сокетов от HTTP заключается в том, что в Web сокетах создается двунаправленное соединение между клиентом и сервером. По этому соединению клиент и сервер могут отправлять данные друг другу в любое время.

Web сокеты определены в стандарте RFC 6455 "The WebSocket Protocol" - https://tools.ietf.org/html/rfc6455

Работа Web сокетов состоит из двух этапов:

- Установка соединения. В целях поддержки существующей инфраструктуры Web используется подход HTTP со сменой (urgrade) протокола на Web сокеты.

- Передача данных. Для передачи данных используется постоянное TCP соединение между клиентом и сервером. Данные передаеются в виде кадров (frames), имеющих бинарные заголовки.

Рассматривается процесс установки соединения Web сокетов, формат кадра Web сокетов и другие темы, относящиеся к работе Web сокетов.

8.2. Протокол IPv6.

IP версии 6 (IPv6) - это протокол передачи данных сетевого уровня, созданный с целью заменить протокол IP версии 4.

Проблема IPv4 - недостаточное количество адресов. Длина адреса IPv4 - 4 байта, максимальное количество адресов - 4.3 млрд.

В IPv6 длина поля адреса - 16 байт. Таким образом, IPv6 предоставляет необходимое количество адресов для подключения устройств к интернет как сейчас, так и в будущем.

Дополнительные цели создания IPv6: упрощение протокола и обеспечение безопасности.

Рассматривается формат заголовка протокола IPv6.

Всемирный запуск протокола IPv6 (IPv6 World Launch) состоялся 6 июня 2012 г. - http://www.worldipv6launch.org

Компания Google предоставляет статистику запросов к своим сервисам по протоколу IPv6 - https://www.google.com/intl/en/ipv6/statistics.html#tab=per-country-ipv6-adoption

8.3. Адреса IPv6.

Длина адреса IPv6 - 16 байт. Для таких длинных адресов был разработан новый формат записи в виде набора из 8 групп шестнадцетиричных чисел, разделенных двоеточиями.

Пример адреса IPv6:

2a02:06b8:0892:ad61:59a2:3149:c5a0:67a4

Рассматриваются правила вычисления префиксов IPv6 (адресов сети).

Для сокращения IPv6 адресов используются два правила:

- Ведущие нули в каждой группе шестнадцитиричных цифр можно удалить.

- Две и более подряд идущие группы нулей можно пропустить.

8.3.1. Типы адресов IPv6:

- Индивидуальный (unicast)
- Групповой (multicast)
- Произвольный (anycast)

8.3.2. Область действия адресов IPv6:

- Глобальные (global unicast address):
- Локальные (unique local address):
- Локальные канала связи (link-local address):

8.4. Автоматическое назначение адресов IPv6.

В IPv6 предусмотрены следующие способы назначения IP-адресов:
- Ручная настройка
- DHCPv6
- Stateless Address Auto Configuration (SLAAC, RFC 4862)

SLAAC используется для полностью автоматической настройки IPv6 адреса. При этом конфигурационная информация получается от следующих источников:
- Префикс IPv6 и шлюз - от маршрутизатора по протоколу NDP
- Идентификатор интерфейса генерируется автоматически хостом по алгоритму EUI-64. Также возможно использование временных IPv6 адресов и Stable Privacy.
- Адреса DNS и др. - от сервера DHCP без фиксации состояния.

8.4.1. Ссыдки на стандарты RFC:

- RFC 4862 IPv6 Stateless Address Autoconfiguration - https://tools.ietf.org/search/rfc4862
- RFC 2373 IP Version 6 Addressing Architecture (описание процедуры создания EUI-64) - https://www.ietf.org/rfc/rfc2373.txt
- RFC 4941 Privacy Extensions for Stateless Address Autoconfiguration in IPv6 (временные IPv6 адреса) - https://tools.ietf.org/html/rfc4941
- RFC 7217 A Method for Generating Semantically Opaque Interface Identifiers with IPv6 Stateless Address Autoconfiguration (SLAAC) - https://tools.ietf.org/html/rfc7217

8.5. Протокол NDP.

Протокол NDP используется совместно с протоколом IPv6. Назначение протокола NDP:
- Определение адреса маршрутизатора и префикса IPv6 (SLAAC).
- Определение MAC-адреса компьютера по его IPv6 адресу (замена ARP для IPv4).
- Настройка маршрутизации (router redirect).
- Проверка доступности узлов сети (соседей).
- Определение конфликта IP адресов.

Протокол NDP определен в RFC 4861 - https://tools.ietf.org/html/rfc4861

Протокол NDP расширяет протокол ICMP: добавлены новые типы сообщений и описан их формат.

8.5.1. Типы сообщений NDP:

133 - Router Solicitation
134 – Router Advertisement
135 – Neighbor Solicitation
136 – Neighbor Advertisement
137 – Redirect

8.6. Протоколы маршрутизации.

Маршрутизация - поиск маршрута доставки пакета между сетями через транзитные узлы (маршрутизаторы)

8.6.1. Этапы маршрутизации:

- Изучение сети (routing)
- Продвижение пакетов на маршрутизаторе (forwarding)

Типы маршрутизации.
1. Статическая маршрутизация:
- Маршруты настраиваются вручную администраторами
- Просто и удобно в небольшой сети
2. Динамическая маршрутизация:
- Маршруты определяются автоматически с помощью протоколов маршрутизации
- Масштабирование сети
- Автоматическая перенастройка в случае сбоя
- Балансировка нагрузки на каналы связи

8.6.2. Типы протоколов маршрутизации.

1. Дистанционно-векторные протоколы (distance-vector protocols):
- Децентрализованный распределенный алгоритм
- Итерационный расчет стоимости путей при неполной информации о сети
- Пример: RIP (Routing Information Protocol)
2. Протоколы с учетом состояния канала (link-state protocols):
- Децентрализованный глобальный алгоритм
- Расчёт стоимости путей после получения полной информации о сети
- Пример: OSPF (Open Shortest Path First)

8.7. Протокол маршрутизации RIP.

RIP - это самый первый протокол маршрутизации для сетей IP. Это дистанционно-векторный протокол, для расчета расстояний используется алгоритм Беллмана – Форда.

Передача данных выполняется через UPD, порт 520.

Вектор расстояния (distance vector) содержит адреса сетей, известных маршрутизатору, и расстояние до них.

Расстояние в RIP – количество промежуточных маршрутизаторов. Максимальное расстояние – 16 (бесконечность).

Проблема RIP - медленная сходимость при изменениях в сети, например, отказе маршрутизатора. В результате возникают петли и возможна ситуация "счет до бесконечности".

Решения проблемы счета до бесконечности:

- Расщепление горизонта (split horizon)
- Отравление маршрута (route poisoning)
- Задержка (holddown)

Сейчас на практике RIP почти не используется. Вместо него применяется модифицированный дистанционно-векторный протокол EIGRP от Cisco и протокол на основе состояния канала OSPF.

8.8. Протокол маршрутизации OSPF.

8.8.1. Протокол OSPF (Open Shortest Path First):

- Протокол с учетом состояния канала (link-state protocol)
- Используется алгоритм Дейкстры (Shortest Path First)
- Передача данных через IP, код протокола 89

8.8.2. Особенности OSPF:

- Децентрализованный глобальный алгоритм
- Расчёт стоимости путей после получения полной информации о сети

8.8.3. Этапы работы протокола OSPF.

1. Изучение топологии сети:
- Маршрутизаторы изучают подключенные сети и ближайших соседей
- Информация о топологии распространяется по всей сети с помощью лавинной рассылки (flooding)

2. Расчет стоимости маршрутов в сети:
- Выполняется после того, как будет известна полная конфигурация сети
- Каждый маршрутизатор выполняет расчет самостоятельно

3. Обновление информации о конфигурации сети:
- Маршрутизаторы проверяют доступность соседей
- Рассылка информации об изменении конфигурации сети

8.9. Иерархическая маршрутизация.

8.9.1. Иерархическая маршрутизация:

- Разделение сети на отдельные области
- Внутри областей и между ними протоколы маршрутизация работает по разному

Примеры:
- Иерархическая маршрутизация в OSPF
- Маршрутизация в Интернет

8.9.2. Политики маршрутизации:

- Сервис «Transit»
- Сервис «Peer»

8.9.3. Протоколы маршрутизации:

- Внутренние (RIP, OSPF)
- Внешние (BGP)

Автономная системы (autonomous system) - это одна или несколько сетей с единым администрированием:

- Имеет свой номер
- Включает адреса IP-сетей (префиксы)
- Часто принадлежат одному провайдеру

Протокол OSPF - https://youtu.be/rDPF5LZiyOw
Протокол RIP - https://youtu.be/LhvQoTsndRs

8.10. Протокол BGP.

BGP (Border Gateway Protocol) – протокол граничного маршрутизатора (шлюза)
- BGPv4 – RFC 4271, 2006
- Дистанционно-векторный алгоритм (вектор пути)
- Протокол TCP, порт 179
- Вычисление маршрутов на уровне автономных систем

Информация о маршруте в BGP содержит следующие компоненты:
- Автономная система одна или несколько IP-сетей)
- Вектор пути (список промежуточных автономных систем)
- Следующий маршрутизатор

Политики маршрутизации:
- Можно выбирать, какие автономные системы анонсировать.