

Элементы сетевого программирования

IPC (Межпроцессное взаимодействие / Inter-Process Communication) - обмен данными между потоками одного или разных процессов.

Канальные IPC: каналы и «почтовые ящики»

Каналы — это объекты, которые работают аналогично файлам и обеспечивают межпроцессное взаимодействие.

Типы каналов:

Неименованный канал (pipe):

- Используется только между родственными процессами.
- Не может быть открыт сторонними процессами.

Именованный канал (FIFO):

- Полноценный объект файловой системы.
- Доступен всем процессам в системе.

Сокеты - механизм взаимодействия, близкий к каналам, используется в сетевом и локальном программировании.

Физический канал - это реальный канал связи, который представляет собой физическую инфраструктуру (например, провода, оптоволокно, беспроводные средства связи), через который передаются данные между устройствами.

Виртуальный канал - это абстрактный канал передачи данных, который создается на основе физического канала. Виртуальный канал обладает своими свойствами и характеристиками, которые могут отличаться от физического канала. Виртуальный канал передачи данных имитирует файл или поток ввода-вывода для прикладных программ.

Коммутация – распределение имеющихся физических каналов для создания виртуальных.

Основные методы:

- **Коммутация каналов:** Соединение абонентов физическим каналом на время сеанса. Можно передавать поток данных или сообщения.
- **Коммутация сообщений:** Канал формируется заново для каждого блока данных. Размер сообщения большой и неизвестен заранее.
- **Коммутация пакетов:** Данные передаются множествами пакетов ограниченного размера, доставляемыми независимо. Фрагментация и сборка автоматически, прозрачно для абонентов.

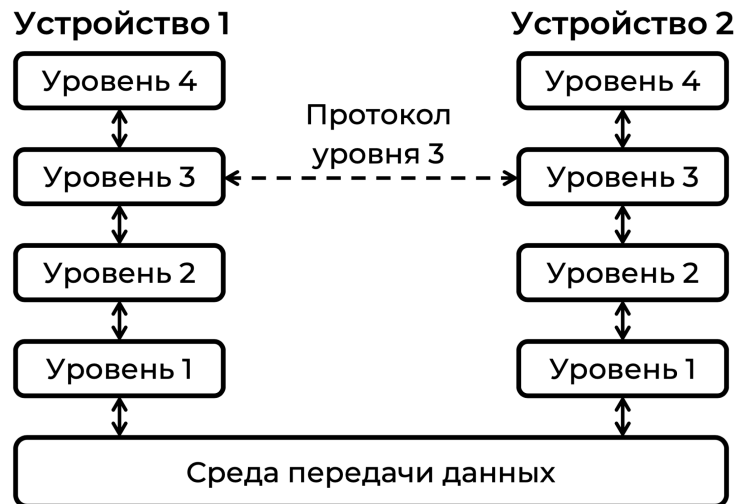
Сети сегодня используют коммутацию пакетов как более надежный и экономичный метод. Apanet (предшественник Internet) исследовала коммутацию пакетов для построения надежной сети передачи данных.

Интернет - компьютерная сеть, объединяющая миллионы компьютеров в единую информационную систему.

Сервис определяет, что именно делает уровень.

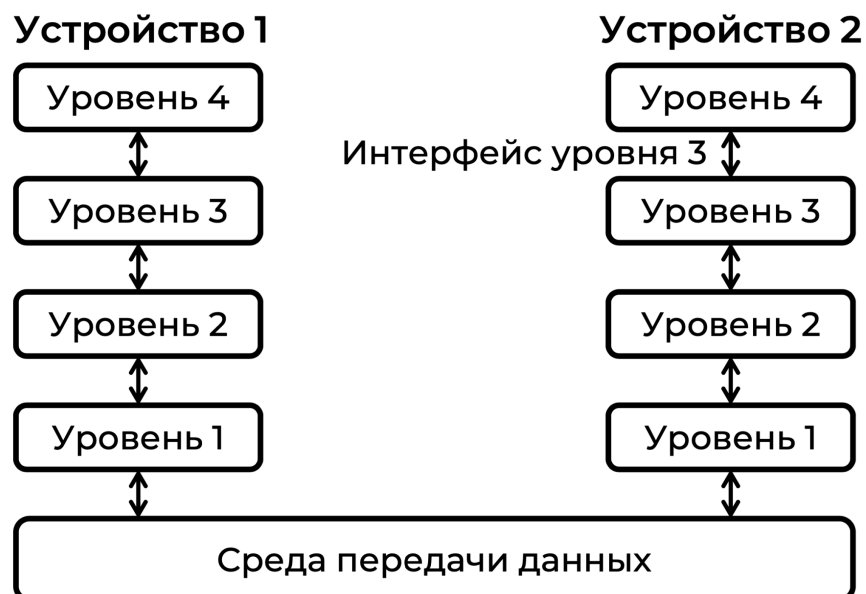
Протокол уровня N — правила и соглашения, используемые для связи уровня N одного устройства с уровнем N другого устройства. (язык общения)

Протокол



Интерфейс — набор примитивных операций, предоставляемых нижним уровнем верхнему.

Интерфейс



Интерфейс и протокол

- Интерфейс
 - Реальное взаимодействие внутри устройства
 - Уровень N вызывает функции уровня N-1
 - Программист создает сокет и записывает в него данные
- Протокол
 - Виртуальное взаимодействие между устройствами
 - Реально соединяются только уровни, работающие с физической средой

Топологией (структурой) компьютерной сети называют физическое расположение компьютеров сети друг относительно друга и способ соединения их линиями связи.

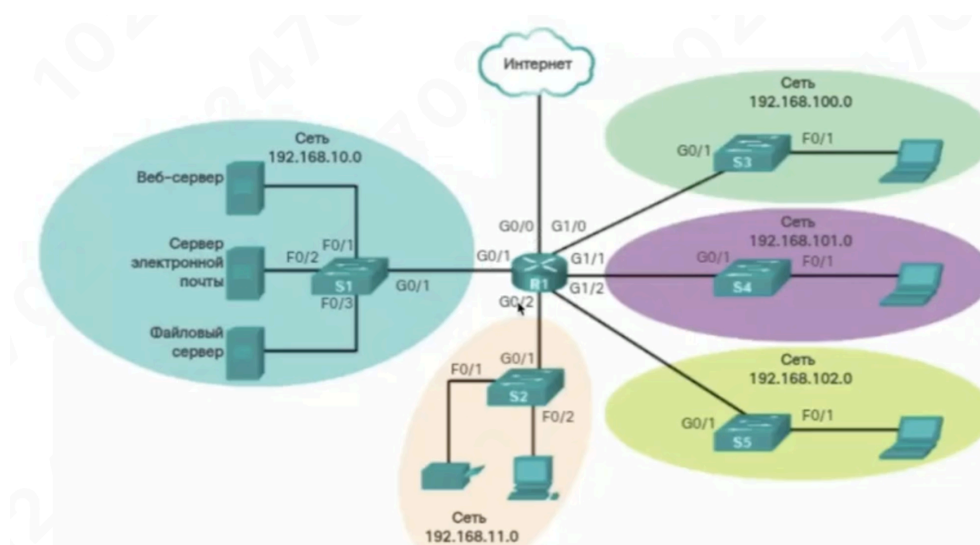
Шлюз (Gateway) – это сетевое устройство, предназначенное для объединения двух сетей (передачи между ними пользовательского трафика), которые обладают различными характеристиками, используют различные протоколы или технологии. Gateway может работать на любом из 7 уровней OSI.

Адрес шлюза = адрес подсети + 1;

Адрес первого устройства = адрес подсети + 2;

n в количестве устройств = 32 - колво единиц в маске;

Маршрутизатор (роутер) — специализированное устройство, которое пересылает данных между компьютерами в сети на основе правил и таблиц маршрутизации. Он работает на сетевом уровне модели OSI.



Обозначения:

круглый по центру - маршрутизатор

параллелограмм - коммутатор

Модель OSI (данные на каждом слое обрастают метаданной):

физический уровень (через канал или радиоволны, wifi, bluetooth)

канальный уровень (хост подключается по проводу к порту коммутатора, коммутация, известно только то, что устройства характеризуются mac-адресом)

сетевой уровень (появляется логическая адресация)

транспортный уровень

сеансовый уровень

уровень представления (шифрование *спорно)

прикладной уровень

ethernet пакет (2 мак адреса, payload максимум 1500 байт, тк это максимальный размер ethernet пакета; checksum (контрольная сумма) показывает, побились ли данные во время передачи)

ipv6 - 10% от маршрутизации трафика (не так много), есть встроенное шифрование, 128 битов, кидает пакеты напрямую (без прокси)

ip пакет (пакет имеет время жизни (может истечь ttl (time to live)); хранит размер пакета и сами данные; 2 ip адреса)

MTU (maximum transmission unit) - максимальный размер полезного блока данных одного пакета.

QoS (quality of service) - механизм, который помогает управлять приоритетами для различных типов интернет-трафика.

Модель DOD (Модель TCP/IP) (англ. Department of Defense — Министерство обороны США)

Описание	Windows	Linux
Чтобы узнать IP/MAC-адрес вашего устройства, посмотреть доступные сетевые интерфейсы на устройстве	ipconfig	ifconfig (ip a)
Утилита для отслеживания маршрута сетевых пакетов (отслеживать маршрут пакетов от вашего компьютера до указанного узла в сети)	tracert	traceroute

TCP handshake - отправка пакетов со специальными флагами:

1. SYN - синхронизация
2. SYN + ACK
3. ACK (acknowledge) - осознание

Состояния клиента и сервера (listen, established, syn-sent, syn-received, fin (ожидание termination запроса))

TCP пакет (2 порта (src, dest), флаги, данные, порядковый номер (для нумерации пакетов для восстановления с принимающей стороны))

UDP - быстрее пакеты передаются, тк не тратится время на установку соединения; пакеты сами занимают меньше места и не имеют такого набора метаданных; пакеты приходят в рандомном порядке (некоторые могут не прийти).

Сервисы использующие UDP: стриминг (видео или игры), dhcp, dns, vpn.
В UDP само приложение разбирается в очереди пакетов.

DNS записи могут кэшироваться + за DNS записи отвечают определенные namespaces, которым делегирована определенная зона.

Используя NAT клиент не знает про роутер.

<https://skillbox.ru/media/code/cto-takoe-ipadres-i-maski-podseti-i-zachem-oni-nuzhny/>
<https://ipcalc.co/>

Сеть - сложная система, состоящая из базовой сети передачи данных (СПД) и конечных узлов. Передача данных может быть с установлением соединения или без него.

При передаче **без установления соединения** данные передаются в виде **датаграмм** по произвольному маршруту без гарантии порядка, уникальности или доставки.

Передача **с установлением соединения** обеспечивает **целостность** и упорядоченность данных, гарантируя доставку в правильном порядке и обнаружение прерываний.

Датаграмма — блок информации, передаваемый протоколом через сеть связи без предварительного установления соединения и создания виртуального канала.

1	Физический	Взаимодействие со средой передачи данных на уровне сигналов (модуляция, демодуляция, детектирование и т.д.)
2	Канальный	Формирование и распознавание низкоуровневых сообщений (кадров), контроль и, возможно, исправление ошибок, низкоуровневая идентификация абонентов (MAC адреса)
3	Сетевой	Логическая идентификация (адресация) абонентов, перенаправление адресованных данных (маршрутизация)
4	Транспортный	Передача данных между абонентами с требуемым качеством и дополнительным сервисом. Первые 4 уровня принято объединять в транспортную систему . Точки доступа к ней принято называть портами .
5	Сеансовый	Установление соединений (сеансов) между абонентами и управление ими
6	Уровень представления	Преобразования форматов данных в соответствии с правилами программного обеспечения следующего 7-го уровня в конкретной системе
7	Прикладной	Конечные приложения, прикладные задачи

Стек протоколов в сети – набор протоколов, обслуживающих различные уровни взаимодействия.



Инкапсуляция блоков данных

Сокет (socket) - программный объект, обеспечивающий доступ к транспортной системе. Он обеспечивает удобный интерфейс для программирования. Сокеты могут быть ассоциированы с портами и транспортными протоколами, обеспечивая унифицированный способ взаимодействия.

На Windows используется подсистема Winsocket (Winsock, Winsock2), в то время как в Unix сокеты часто реализованы в ядре. Для компиляции на Windows с использованием Visual Studio требуются определенные заголовочные файлы, библиотеки и инициализация подсистемы.

Основные характеристики сокета включают в себя:

- Тип протоколов - выбор стека протоколов, например, TCP/IP.
- Тип адресации - способ идентификации участников взаимодействия.
- Адрес сокета - идентификатор точки взаимодействия.
- Тип сокета - выбранный протокол взаимодействия.

Сокеты обычно совместимы с файловыми дескрипторами и могут использоваться в функциях файлового ввода-вывода.

Структуры адресации:

struct sockaddr: базовая структура адреса

struct sockaddr_in: структура для адреса IPv4

struct in_addr: структура для представления IP-адреса

Функции для работы с адресами:

Для заполнения адреса: inet_aton(), inet_pton(), inet_addr()

Для обратного преобразования: inet_ntoa(), inet_ntop()

Для разрешения сетевого имени в адрес: gethostbyname(), getnameinfo()

Операции с сокетами:

Создание сокета: socket()

Связывание сокета с локальным и удаленным адресом: bind(), connect()

Установка сокета в режим прослушивания: listen()

Прием/передача данных: recv(), send(), accept()

- send(): Отправка данных через установленное соединение.
- recv(): Прием данных через установленное соединение.
- sendto(): Отправка данных с указанием адреса назначения.
- recvfrom(): Прием данных и получение информации об отправителе.

Закрытие сокета: closesocket(), shutdown()

Управление параметрами сокета:

Установка и получение параметров: setsockopt(), getsockopt()

Низкоуровневое управление: ioctlsocket()

Прочие функции:

Для согласования формата чисел: htons(), htonl(), ntohs(), ntohl()

Windows Sockets API (Winsock):

Поддерживает сокеты в Windows-системах

Инициализация и деинициализация: WSASStartup(), WSACleanup()

Winsock 2 добавляет дополнительные возможности, включая асинхронную работу и поддержку различных протоколов.

Клиент-Серверное Взаимодействие:

Сервер: Программа, обрабатывающая запросы.

Клиент: Программа, отправляющая запросы и получающая результаты.

Взаимодействие без Установления Соединения:

Симметричное взаимодействие с различием ролей лишь на уровне логики.

Взаимодействие с Установлением Соединения:

Несимметричное взаимодействие с использованием сокетов и отдельных точек для соединения и обмена данными.

Многопользовательские Серверы:

Распараллеливание работы для обслуживания нескольких клиентов.

Последовательный Сервер:

Последовательная обработка запросов без параллельного выполнения.

Многозадачные Многопользовательские Серверы:

Параллельное обслуживание соединений с использованием многозадачности или многопоточности.

Сервер с Мультиплексированием Обработки Запросов:

Проверка и обработка данных сразу из всех открытых сокетов для эффективного обслуживания.