

# **IPC**

## **(inter-process communication)**

Межпроцессное взаимодействие

Обмен данными между потоками одного или разных процессов.

Реализуется посредством механизмов, предоставляемых ядром ОС или процессом.

Может осуществляться как на одном компьютере, так и между несколькими компьютерами сети.

# Методы межпроцессного взаимодействия

- **Фаил (File)** - взаимодействие процессов через общие файлы.
- **Канал (Pipe)** - представляет собой средство связи стандартного вывода одного процесса со стандартным вводом другого. Каналы старейший из инструментов IPC, существующий приблизительно со времени появления самых ранних версий операционной системы UNIX.
- **Сигнал (Signal)** - являются программными прерываниями, которые посылаются процессу, когда случается некоторое событие.
- **Очереди сообщений (Message Queues)** - представляют собой связный список в адресном пространстве ядра. Сообщения могут посылаться в очередь по порядку и доставаться из очереди несколькими разными путями.
- **Семафор (Semaphore)** - это целая переменная, значение которой можно опрашивать и менять только при помощи неделимых (атомарных) операций. Двоичный семафор может принимать только значения 0 или 1.
- **Разделяемая память (Shared memory)** - может быть наилучшим образом описана как отображение участка (сегмента) памяти, которая будет разделена между более чем одним процессом.

# Сокет (Socket)

**Сокет** – программный интерфейс для обеспечения обмена данными между процессами. Процессы с таким обменом могут выполняться как на одном компьютере, так и на разных компьютерах, соединенных сетью.

**Сокет** - это абстрактный объект, представляющий конечную точку соединения.

Сокет делятся на два вида: **клиентский** и **серверный**

Клиентское приложение (например, браузер) использует только **клиентские сокеты**, а серверное (например, веб-сервер, которому браузер посылает запросы) — как **клиентские**, так и **серверные сокеты**.

Для связи между процессами, использующими стек протоколов TCP/IP, вам нужен **адрес и порт**.  
Эта пара определяет СОКЕТ!

# Основные типы сокетов

**Поточный** – обеспечивает двусторонний, последовательный, надежный, и недублированный поток данных без определенных границ. Тип сокета - **SOCK\_STREAM**, в домене Интернета он использует протокол **TCP**.

**Датаграммный** – поддерживает двухсторонний поток сообщений. Приложение, использующее такие сокеты, может получать сообщения в порядке, отличном от последовательности, в которой эти сообщения посылались. Тип сокета - **SOCK\_DGRAM**, в домене Интернета он использует протокол **UDP**.

**Сокет последовательных пакетов** – обеспечивает двусторонний, последовательный, надежный обмен датаграммами фиксированной максимальной длины. Тип сокета - **SOCK\_SEQPACKET**. Для этого типа сокета не существует специального протокола.

**Простой сокет** – обеспечивает доступ к основным протоколам связи.

# Протокол связи

Это система правил, которая позволяет двум или более объектам системы передавать информацию посредством любого изменения физической величины.

Протокол определяет правила, синтаксис, семантику и синхронизацию связи и возможные методы восстановления после ошибок.

Протоколы могут быть реализованы аппаратными средствами, программным обеспечением или их комбинацией.

# Сетевая модель

теоретическое описание принципов работы набора сетевых протоколов, взаимодействующих друг с другом.

Модель обычно делится на уровни, так, чтобы данные протокола вышестоящего уровня передавались бы с помощью нижележащих протоколов — этот процесс называют инкапсуляцией, процесс извлечения данных вышестоящего уровня из данных нижестоящего — декапсуляцией.

Модели бывают как практические (использующиеся в сетях), так и теоретические (показывающие принципы реализации сетевых моделей).

# Наиболее известные сетевые модели

**Модель OSI**, она же Модель ВОО, Взаимосвязь открытых систем. Эталонная модель. — теоретическая модель, описанная в международных стандартах и ГОСТах.

**Модель DOD (Модель TCP/IP)** — практически используемая модель, принятая для работы в Интернете.

**Модель SPX/IPX** — модель стека SPX/IPX (семейство протоколов для ЛВС).

**Модель AppleTalk** — модель для сетей AppleTalk (протоколы для работы сетей с оборудованием Apple).

**Модель Fibre Channel** — модель для высокоскоростных сетей Fibre Channel.

# Модель OSI

Open Systems Interconnection Basic Reference Model — Базовая Эталонная Модель Взаимодействия Открытых Систем (ЭМВОС)) — сетевая модель стека сетевых протоколов OSI/ISO (ГОСТ Р ИСО/МЭК 7498-1-99).

Посредством данной модели различные сетевые устройства могут взаимодействовать друг с другом. Модель определяет различные уровни взаимодействия систем. Каждый уровень выполняет определенные функции при таком взаимодействии.



# Модель TCP/IP

концептуальная модель и набор коммуникационных протоколов, используемых в Интернете и подобных компьютерных сетях.

Базовые протоколы в пакете - это протокол управления передачей (TCP) и интернет-протокол (IP).

Его иногда называют моделью Министерства обороны (МО), поскольку разработка сетевого метода финансировалась Министерством обороны Соединенных Штатов через DARPA.

Набор интернет-протоколов обеспечивает сквозную передачу данных, определяющую, как данные должны пакетироваться, обрабатываться, передаваться, маршрутизироваться и приниматься.

# История TCP/IP

Стек протоколов TCP/IP был создан на основе NCP (Network Control Protocol) группой разработчиков под руководством Винтона Серфа в 1972 году. В июле 1976 года Винт Серф и Боб Кан впервые продемонстрировали передачу данных с использованием TCP по трём различным сетям.

Пакет прошел по следующему маршруту: Сан-Франциско — Лондон — Университет Южной Калифорнии.

К концу своего путешествия пакет проделал 150 тысяч км, не потеряв ни одного бита. В 1978 году Серф, Джон Постел и Дэнни Кохэн решили выделить в TCP две отдельные функции: TCP и IP.

TCP был ответственен за разбивку сообщения на датаграммы (англ. datagram) и соединение их в конечном пункте отправки. IP отвечал за передачу (с контролем получения) отдельных датаграмм. Вот так родился современный протокол Интернета. А 1 января 1983 года ARPANET перешла на новый протокол. Этот день принято считать официальной датой рождения Интернета.

# Уровни стека

## TCP/IP

Прикладной уровень  
**(Application Layer)**  
HTTP, FTP, SMTP, Any Custom Protocol

Транспортный уровень  
**(Transport Layer)**  
TCP, UDP

Сетевой (межсетевой) уровень  
**(Network Layer)**  
IP, ICMP, IGMP

Канальный уровень  
**(Link Layer)**  
Ethernet, WLAN, DSL, ADSL

## OSI

Прикладной уровень  
**(Application Layer)**  
HTTP, FTP, SMTP, Any Custom Protocol

Уровень представления  
**(Presentation Layer)**  
FTP, Telnet, SMTP

Сеансовый уровень  
**(Session Layer)**  
RPC, NetBIOS, ASP

Транспортный уровень  
**(Transport Layer)**  
TCP, UDP

Сетевой (межсетевой) уровень  
**(Network Layer)**  
IP, ICMP, IGMP

Канальный уровень  
**(Data Link Layer)**  
Ethernet, WLAN, DSL, ADSL

Физический уровень  
**(Physical Layer)**  
RS-232, RS-422, RS-423,

# TCP/IP

## Прикладной уровень (APPLICATION LAYER)

На прикладном уровне работает большинство сетевых приложений.

Эти программы имеют свои собственные протоколы обмена информацией, например:

- интернет браузер для протокола HTTP
- ftp-клиент для протокола FTP (передача файлов)
- почтовая программа для протокола SMTP (электронная почта)
- SSH (безопасное соединение с удалённой машиной)
- DNS (преобразование символьных имён в IP-адреса)

и многие другие.

# TCP/IP

## Транспортный уровень (TRANSPORT LAYER)

Протоколы транспортного уровня могут решать проблему негарантированной доставки сообщений («дошло ли сообщение до адресата?»), а также гарантировать правильную последовательность прихода данных.

В стеке TCP/IP транспортные протоколы определяют, для какого именно приложения предназначены эти данные.

# TCP (Transmission Control Protocol)

## UDP (User Datagram Protocol)

**TCP** — «гарантированный» транспортный механизм с предварительным установлением соединения, предоставляющий приложению надёжный поток данных, дающий уверенность в безошибочности получаемых данных, перезапрашивающий данные в случае потери и устраняющий дублирование данных. TCP позволяет регулировать нагрузку на сеть, а также уменьшать время ожидания данных при передаче на большие расстояния.

**UDP** — протокол передачи датаграмм без установления соединения. Также его называют протоколом «ненадёжной» передачи, в смысле невозможности удостовериться в доставке сообщения адресату, а также возможного перемешивания пакетов. Обычно используется в таких приложениях, как потоковое видео и компьютерные игры, где допускается потеря пакетов, а повторный запрос затруднён или не оправдан, либо в приложениях вида запрос-ответ (например, запросы к DNS), где создание соединения занимает больше ресурсов, чем повторная отправка.

# Структура пакета TCP

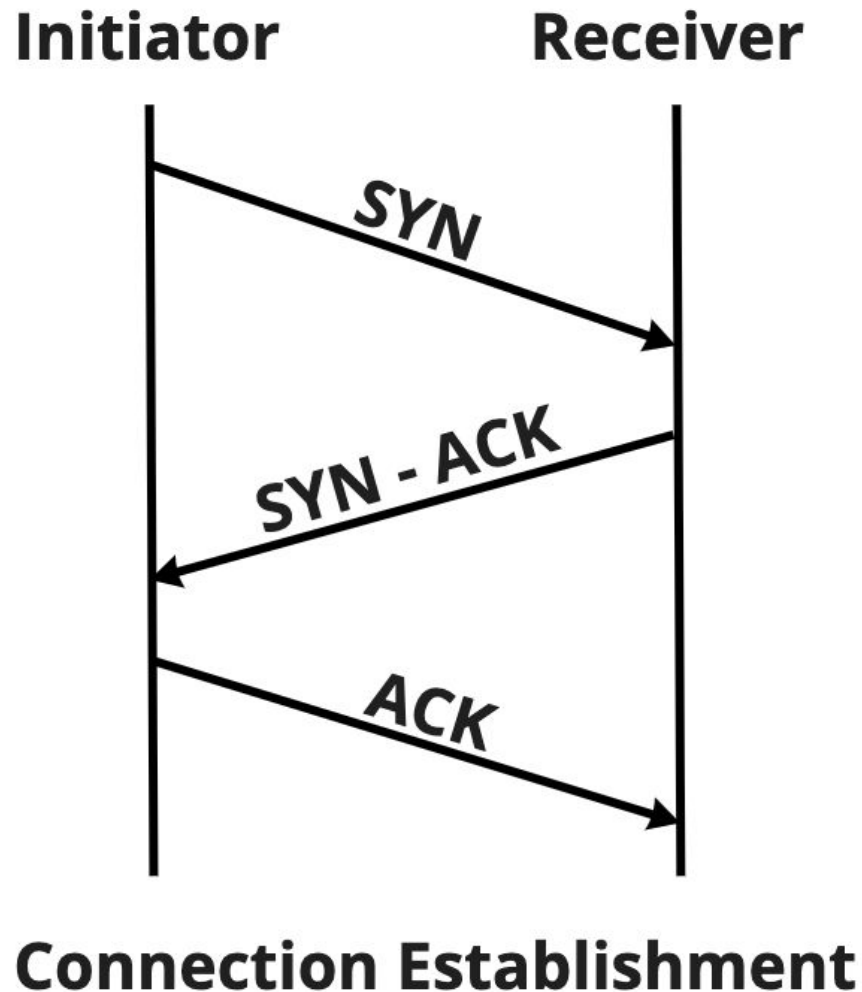
Бит	0-3	4-9	10-15	16-31
0	Порт источника, <b>Source Port</b>			Порт назначения, <b>Destination Port</b>
32	Порядковый номер, <b>Sequence Number (SN)</b>			
64	Номер подтверждения, <b>Acknowledgment Number (ACK SN)</b>			
96	Длина заголовка	Зарезервировано	Флаги	Размер Окна
128	Контрольная сумма			Указатель важности
160	Опции (необязательное, но используется практически всегда)			
160/192+	Данные			

# Структура пакета UDP

Бит	0-15	16-31
0-31	Порт отправителя ( <b>Source port</b> )	Порт получателя ( <b>Destination port</b> )
32-63	Длина датаграммы ( <b>Length</b> )	Контрольная сумма ( <b>Checksum</b> )
64+	Данные ( <b>Data</b> )	



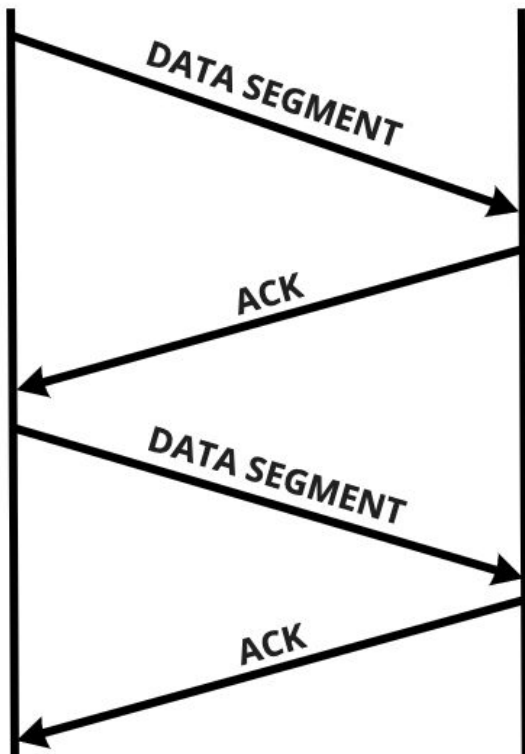
# Установка соединения ТСР Connection (multi-step handshake process)



# Передача данных TCP Data Transfer

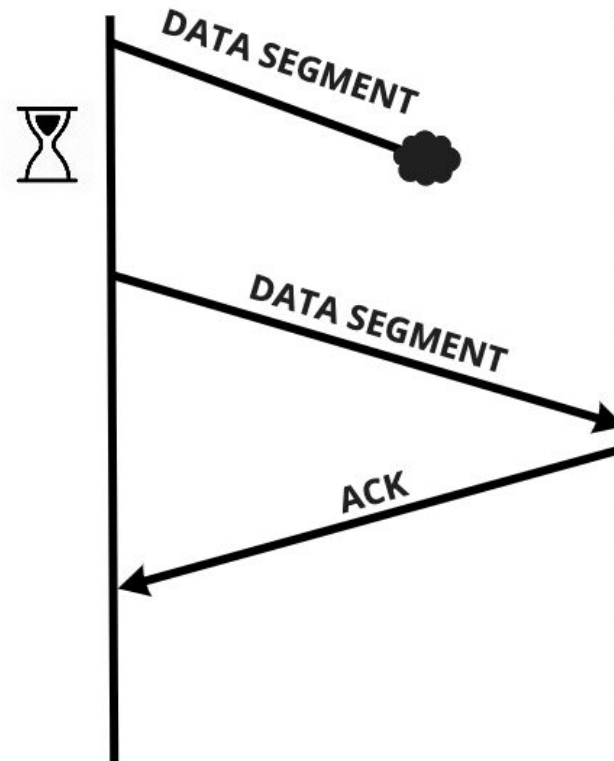
Initiator

Receiver

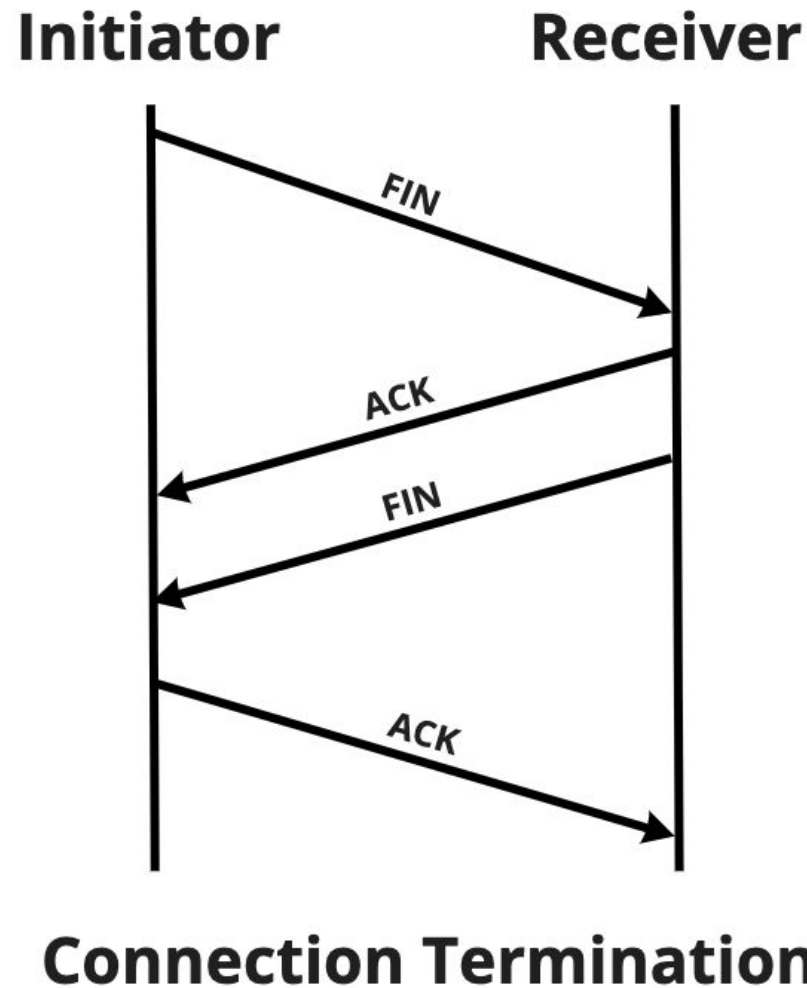


Initiator

Receiver



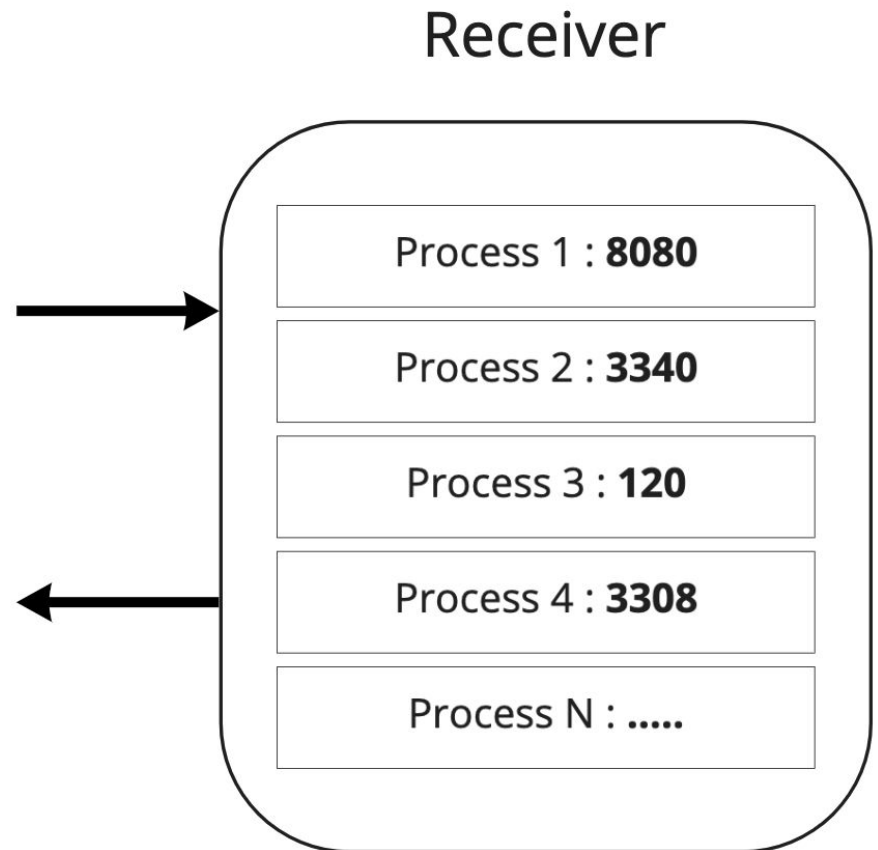
# Завершение соединения TCP Termination



# TCP Port

натуральное число,  
записываемое в заголовках  
протоколов транспортного  
уровня модели OSI (TCP, UDP,  
SCTP, DCCP).

Используется для определения  
процесса-получателя пакета в  
пределах одного устройства.



# TCP/IP

## Сетевой уровень (NETWORK LAYER)

Обеспечивает маршрутизацию, и управление загрузкой канала передачи, предоставляет необработанный маршрут передачи, состоящий лишь из конечных точек. Отвечает за деление пользователей на группы. На этом уровне происходит маршрутизация пакетов на основе преобразования MAC-адресов в сетевые адреса. Сетевой уровень обеспечивает также прозрачную передачу пакетов на транспортный уровень.

**Internet Protocol** (IP «межсетевой протокол») — маршрутизируемый протокол сетевого уровня стека TCP/IP. Именно IP стал тем протоколом, который объединил отдельные компьютерные сети во всемирную сеть Интернет. Неотъемлемой частью протокола является адресация сети

# Internet Protocol Address

является обязательным уникальным логическим адресом, который должен иметь каждое устройство в Интернете.

**IPv4** использует 32-битные (четырёхбайтные) адреса, ограничивающие адресное пространство 4 294 967 296 ( $2^{32}$ ) возможными уникальными адресами.

Традиционной формой записи IPv4 адреса является запись в виде четырёх десятичных чисел (от 0 до 255), разделенных точками. Через дробь указывается длина маски подсети. **Пример: 123.12.34.7**

**IPv6** — новая версия интернет протокола (IP), призванная решить проблемы, с которыми столкнулась предыдущая версия (IPv4) при её использовании в Интернете, за счёт использования длины адреса 128 бит вместо 32.

# Назначения подсетей

**Loopback** адреса – по соглашению, адрес **127.0.0.1 (localhost)** назначается интерфейсу обратной связи.

Все, что отправлено на этот IP-адрес, зацикливается и становится IP-входом, никогда не покидая машину.

Этот идентификатор адреса часто используется при тестировании клиента и сервера на одном хосте. (Этот адрес известен как `INADDR_LOOPBACK`).

**Unspecified** адрес – состоящий из 32 нулевых битов. Может посылаться в сеть только в качестве адреса источника, если хосту еще не назначен IP адрес.

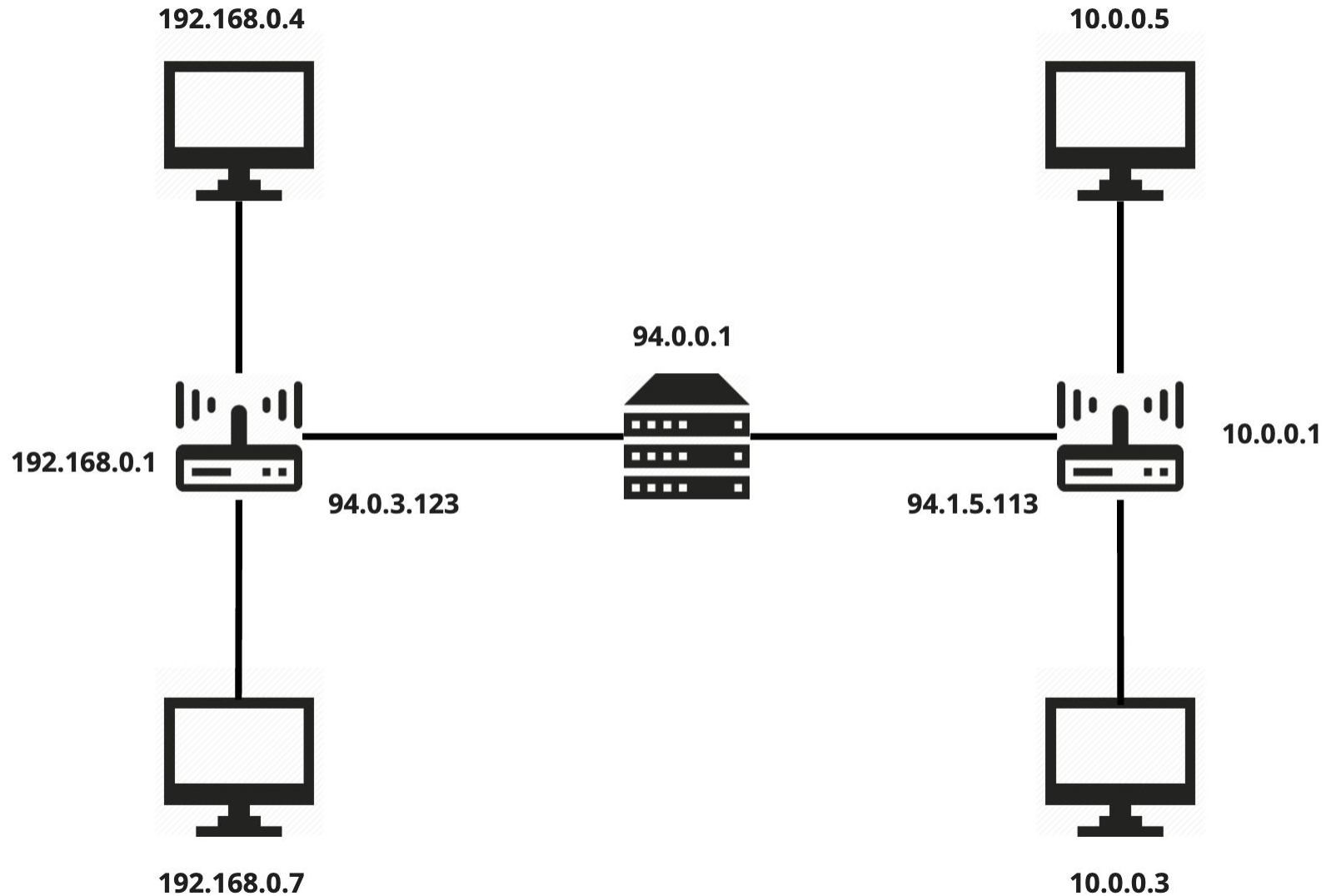
**Private** адреса – для использования в частных сетях.

**10.0.0.0 ... 10.255.255.255**

**172.16.0.0 ... 172.31.255.255**

**192.168.0.0 ... 192.168.255.255**

# Сеть Network





# TCP/IP

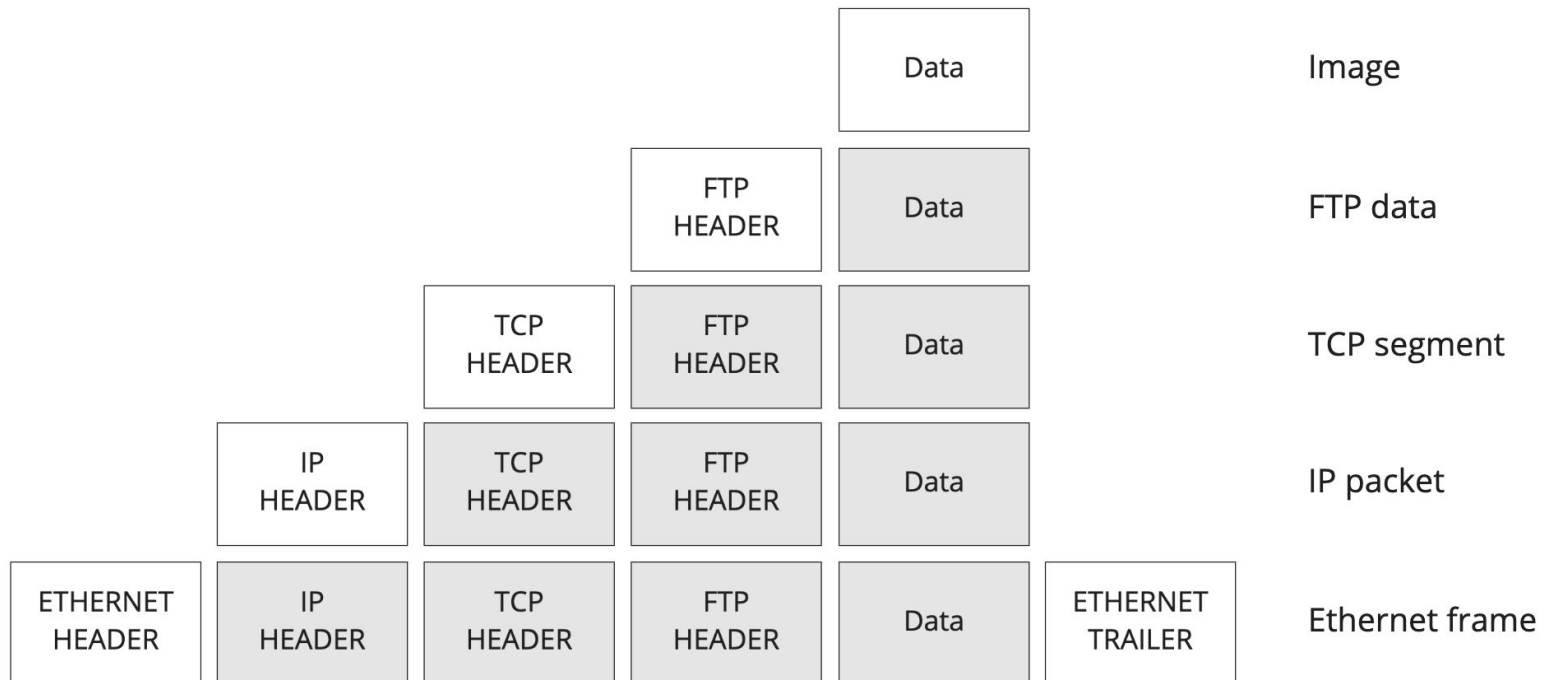
## Канальный уровень (LINK LAYER)

описывает способ кодирования данных для передачи пакета данных на физическом уровне (то есть специальные последовательности бит, определяющих начало и конец пакета данных, а также обеспечивающие помехоустойчивость).

Ethernet, например, в полях заголовка пакета содержит указание того, какой машине или машинам в сети предназначен этот пакет.

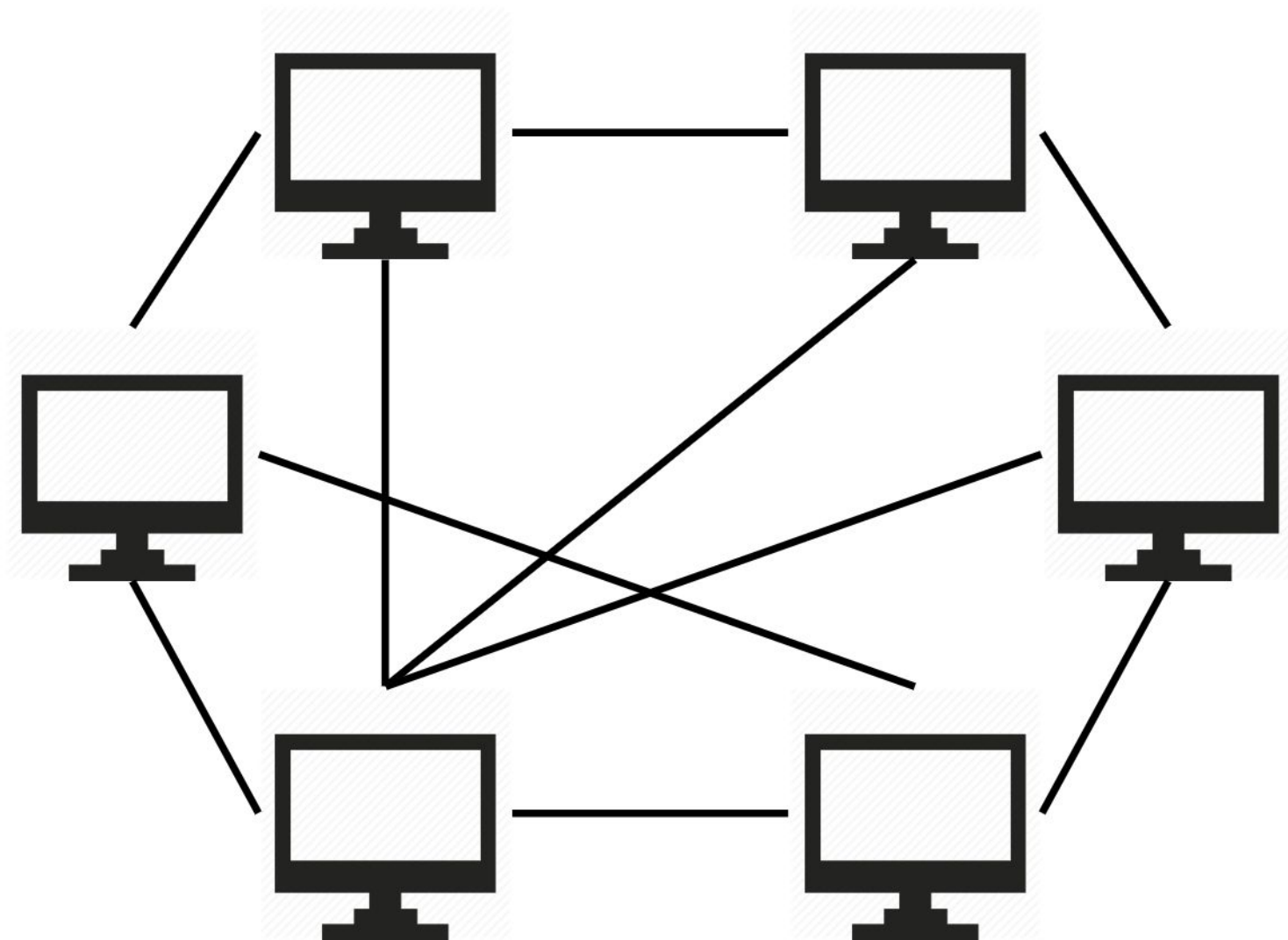
# Инкапсуляция данных в TCP/IP (Data Encapsulation)

Инкапсуляция - это сокрытие данных объекта от остального мира.  
Для семейства протоколов это означает:  
добавление управляющей информации к данным при переходе на  
один уровень вниз.

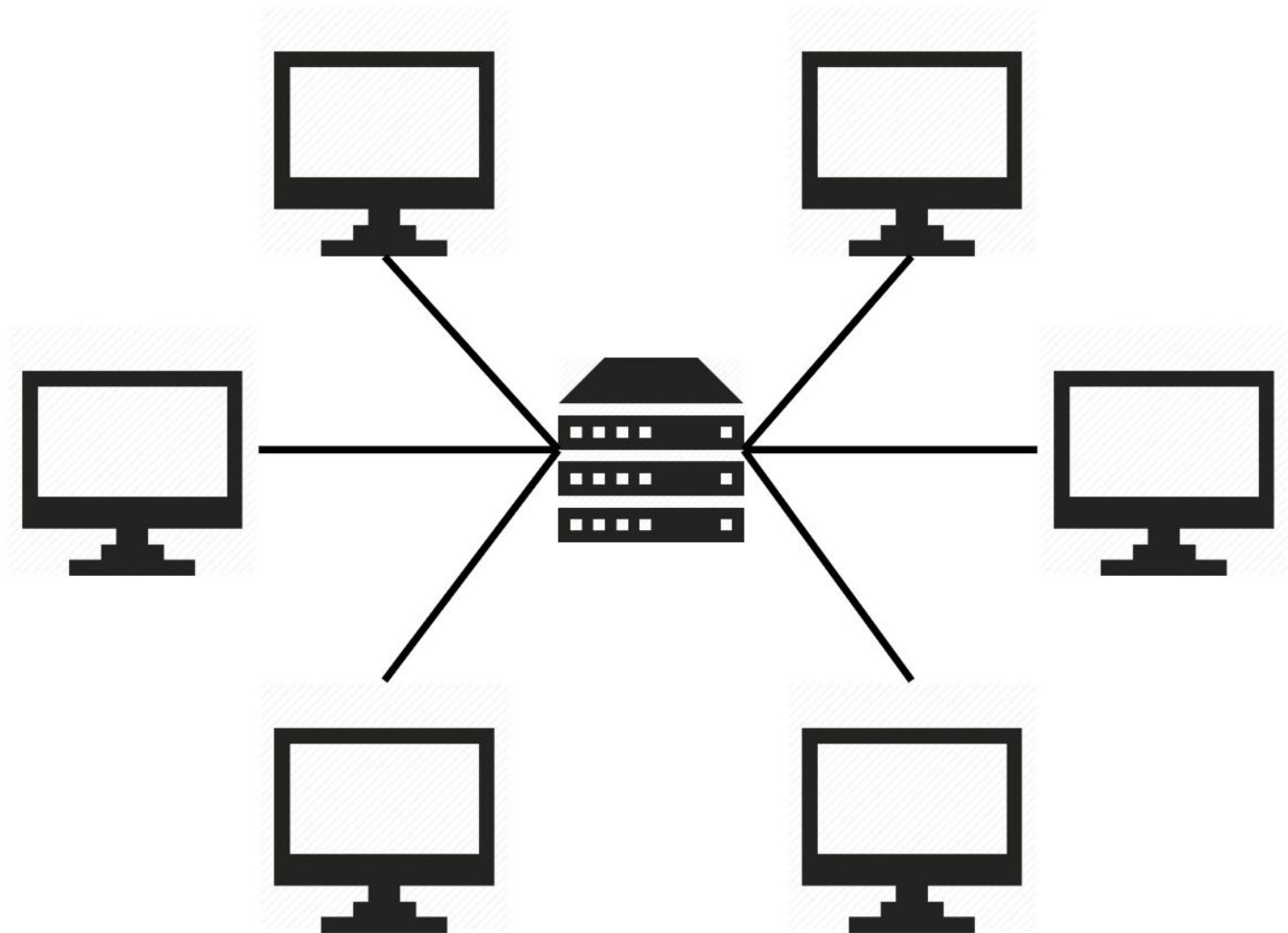


# Архитектура точка-точка

## NETWORK P2P (Peer-to-Peer) Architecture

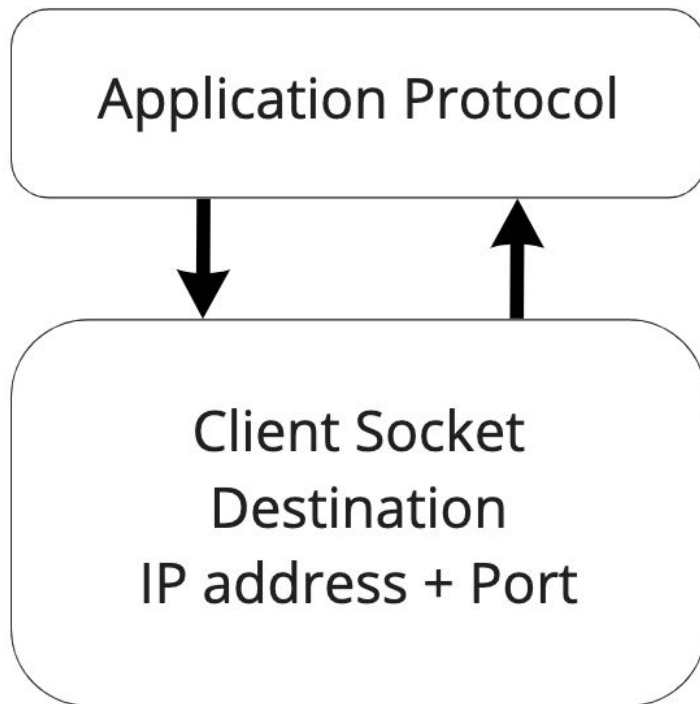


# Архитектура клиент-сервер NETWORK Client-Server Architecture



# IPC Java

## INITIATOR (CLIENT)



## RECEIVER (SERVER)

