



Сети ЭВМ и телекоммуникации

Лекция 6.

Проблемы стандартизации.
Модель OSI. Стек TCP/IP



Понятие открытой системы

- **Открытой системой** может быть названа любая система (компьютер, вычислительная сеть, ОС, программный пакет, другие аппаратные и программные продукты), которая построена в соответствии с открытыми спецификациями
- **Спецификация** - это формализованное описание аппаратных или программных компонентов, способов их функционирования, взаимодействия с другими компонентами, условий эксплуатации, ограничений и особых характеристик
- **Открытая спецификация** – опубликованная общедоступная спецификация, соответствующая стандартам и принятая в результате обсуждения всеми заинтересованными сторонами
- **Полная открытость невозможна.** Но взаимодействие оборудования с оборудованием конкурента – это большое преимущество, поэтому построение открытых систем выгодно



Модель взаимодействия открытых систем

- В 1984 году международные организации по стандартизации: International Standards Organization (ISO) и International Telecommunications Union (ITU) разработали стандартную **модель взаимодействия открытых систем (Open System Interconnection, OSI)**
- **Цель** – обобщенное представление средств сетевого взаимодействия сетей
- **Модель OSI определяет:**
 - Уровни взаимодействия систем в сетях
 - Стандартные названия уровней
 - Функции, которые должен выполнять каждый уровень
- **Модель OSI НЕ СОДЕРЖИТ** описаний реализаций конкретного набора протоколов

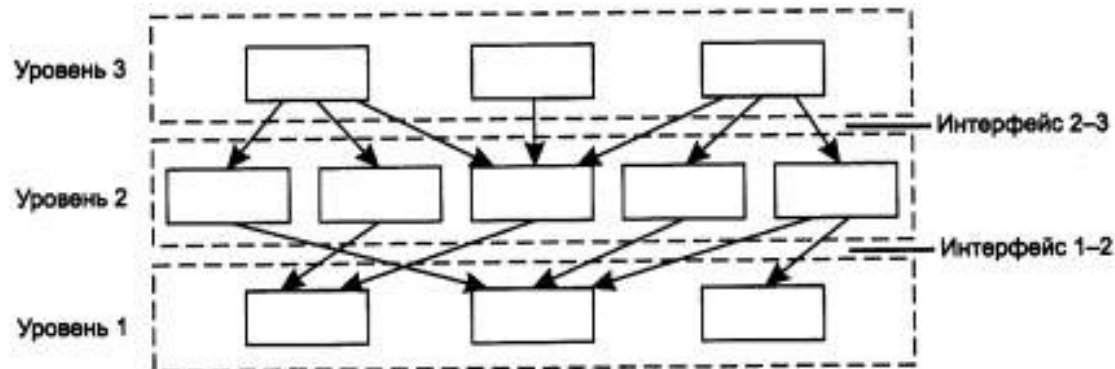


Преимущества открытых систем

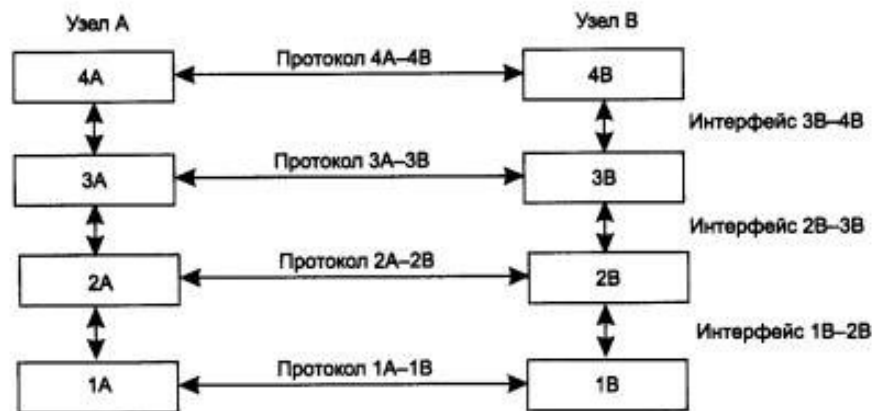
- Возможность построения сети из аппаратных и программных средств различных производителей, придерживающихся одного и того же стандарта
- Возможность безболезненной замены отдельных компонентов сети другими, более совершенными, что позволяет сети развиваться с минимальными затратами
- Возможность легкого сопряжения одной сети с другой
- Простота освоения и обслуживания сети

Многоуровневый подход

- Все множество модулей разбивают на уровни
- Уровни образуют иерархию, то есть имеются вышележащие и нижележащие уровни
- Множество модулей, составляющих каждый уровень, сформировано таким образом, что для выполнения своих задач они обращаются с запросами только к модулям непосредственно примыкающего нижележащего уровня
- Результаты работы всех модулей, принадлежащих некоторому уровню, могут быть переданы только модулям соседнего вышележащего уровня

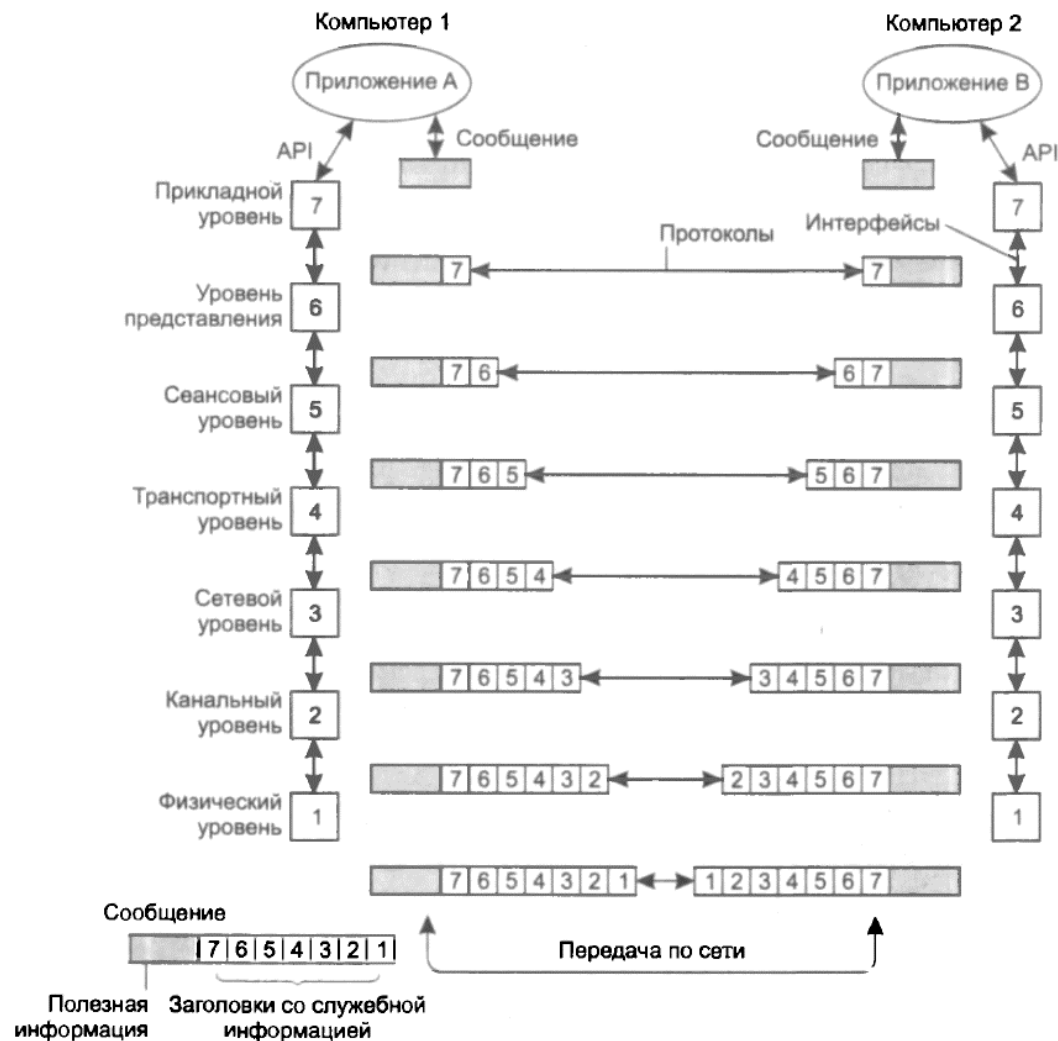


Многоуровневый подход



- Формализованные правила, определяющие последовательность и формат сообщений, которыми обмениваются сетевые компоненты, лежащие на одном уровне, но в разных узлах, называются **протоколом**
- Модули, реализующие протоколы соседних уровней и находящиеся в одном узле, также взаимодействуют друг с другом в соответствии с четко определенными правилами и с помощью стандартизованных форматов сообщений. Эти правила принято называть **интерфейсом**
- Иерархически организованный набор протоколов, достаточный для организации взаимодействия узлов в сети, называется **стеком коммуникационных протоколов**

Модель OSI

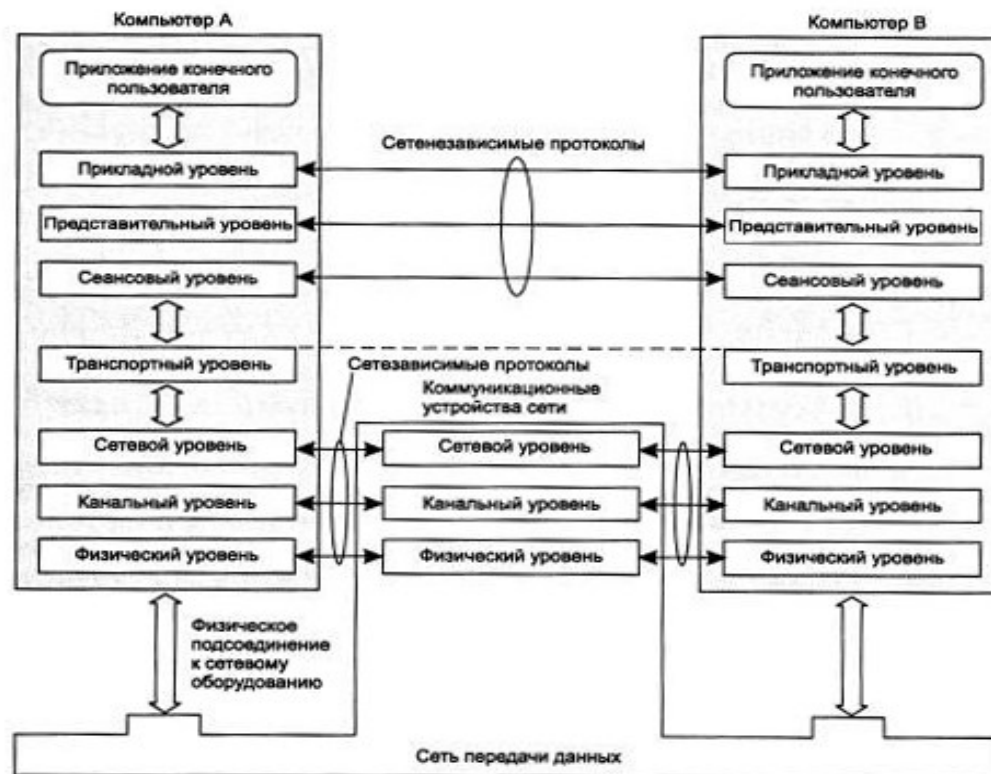


Уровни модели OSI:

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

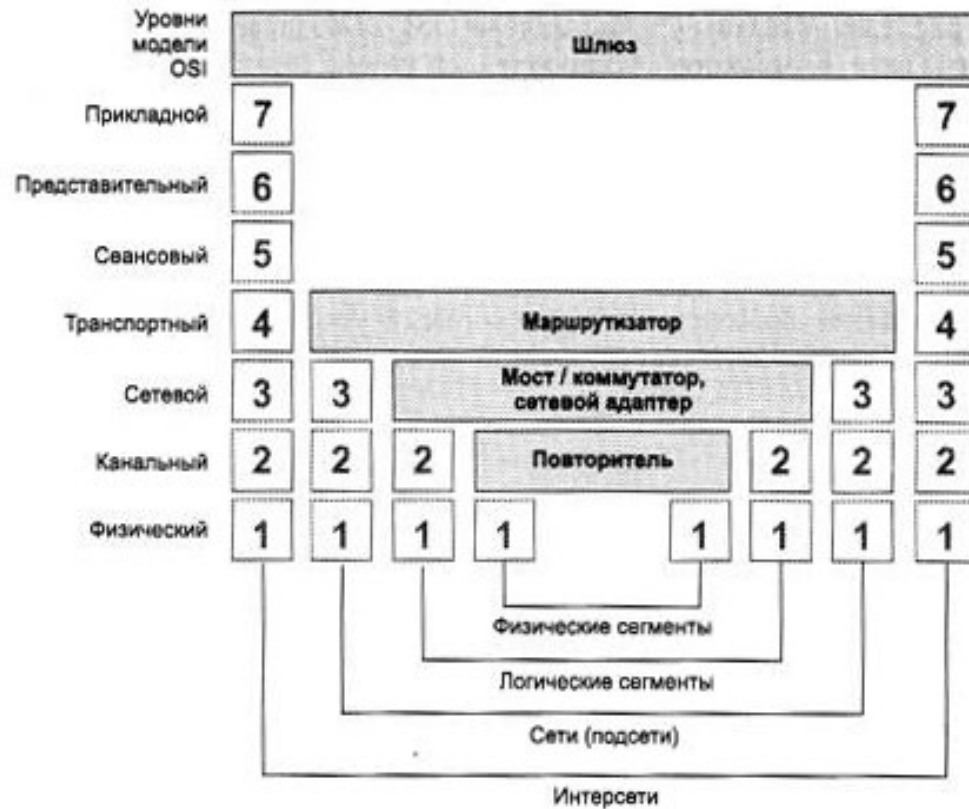
Модель OSI

- Протоколы верхних уровней являются **сетезависимыми**
- Протоколы Физического, Канального и Сетевого уровня **зависят** от используемой технологии построения сети



Модель OSI

Соответствие функций различных устройств сети уровням модели OSI



Физический уровень (Physical layer)

Функция: передача потока битов по физическим каналам связи (коаксиальный кабель, витая пара, оптоволокно)

- Функции физического уровня реализуются на всех сетевых устройствах
- На ПК функции физического уровня выполняет сетевой адаптер или последовательный порт
- **Пример** протокола физического уровня: спецификация 10Base-T технологии Ethernet:
 - Неэкранированная витая пара категории 3
 - Разъем RJ-45
 - Максимальная длина физического сегмента 100 м
 - Манчестерский код
- Физический уровень не вникает в смысл передаваемой информации – для него это **однородный поток битов**

Физический уровень (Physical layer)



Канальный уровень (Data link layer)

Функция: Обеспечивает прозрачность соединения для сетевого уровня:

- Установление логического соединения между взаимодействующими узлами
- Согласование в рамках соединения скоростей передатчика и приемника информации
- Обеспечение надежной передачи, обнаружение и коррекция ошибок – фиксация границ кадра, специальная последовательность битов в начале и конце + контрольная сумма; повторная передача в случае несовпадения контрольной суммы
- Спецификации IEEE 802.x делят канальный уровень на два подуровня: **управление логическим каналом** (LLC) и **управление доступом к среде** (MAC)
- Единицей передачи данных на канальном уровне является **кадр**
- Протоколы канального уровня реализуются на промежуточных и конечных узлах сети средствами сетевых адаптеров и их драйверов

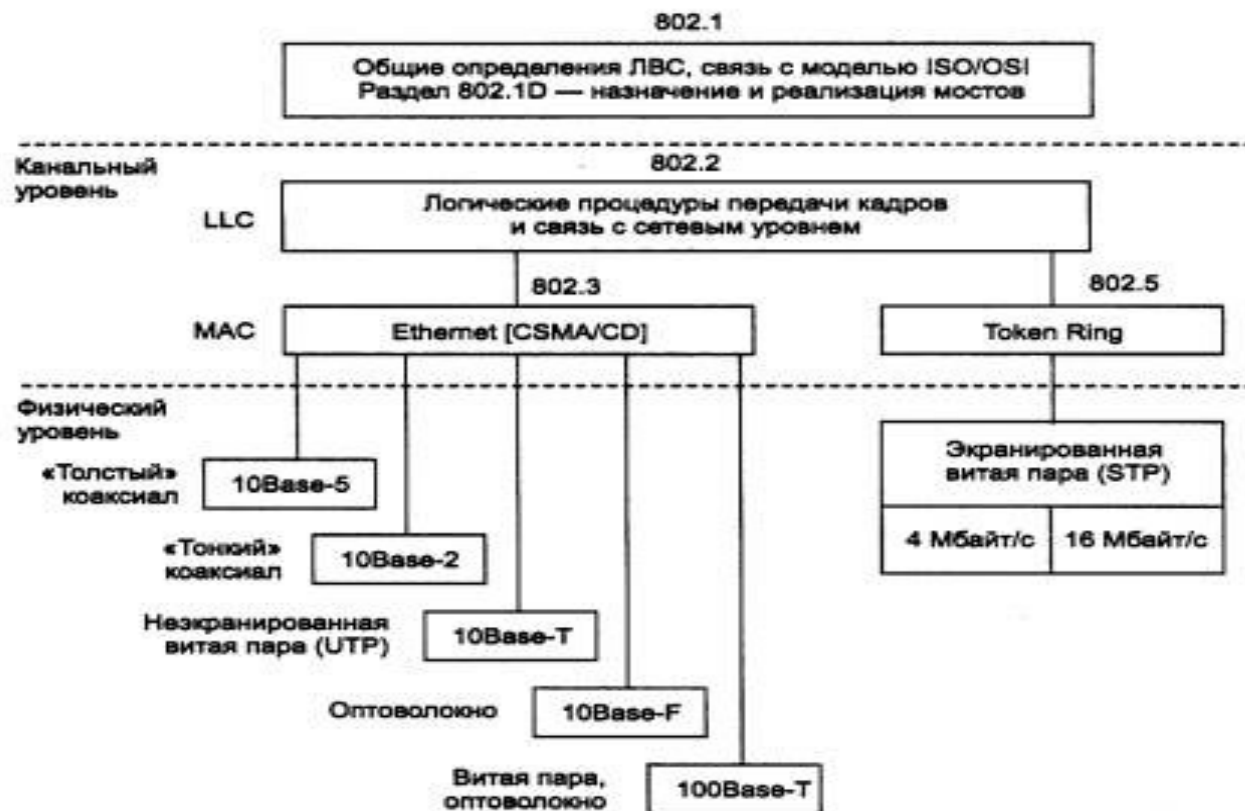


Канальный уровень (Data link layer)

- Протокол канального уровня работает **только в пределах сети**, являющейся составной частью более крупной составной сети, объединенной протоколами сетевого уровня. Адреса канального уровня используются для доставки кадров только в пределах этой сети. Для перемещения пакетов между сетями используются адреса сетевого уровня
- **Примеры** спецификаций канального уровня:
 - HDLC для последовательных соединений
 - Ethernet
 - Token ring
 - FDDI
 - X.25
 - Frame relay

Канальный уровень (Data link layer)

Распределение функций между физическим и канальным уровнями:





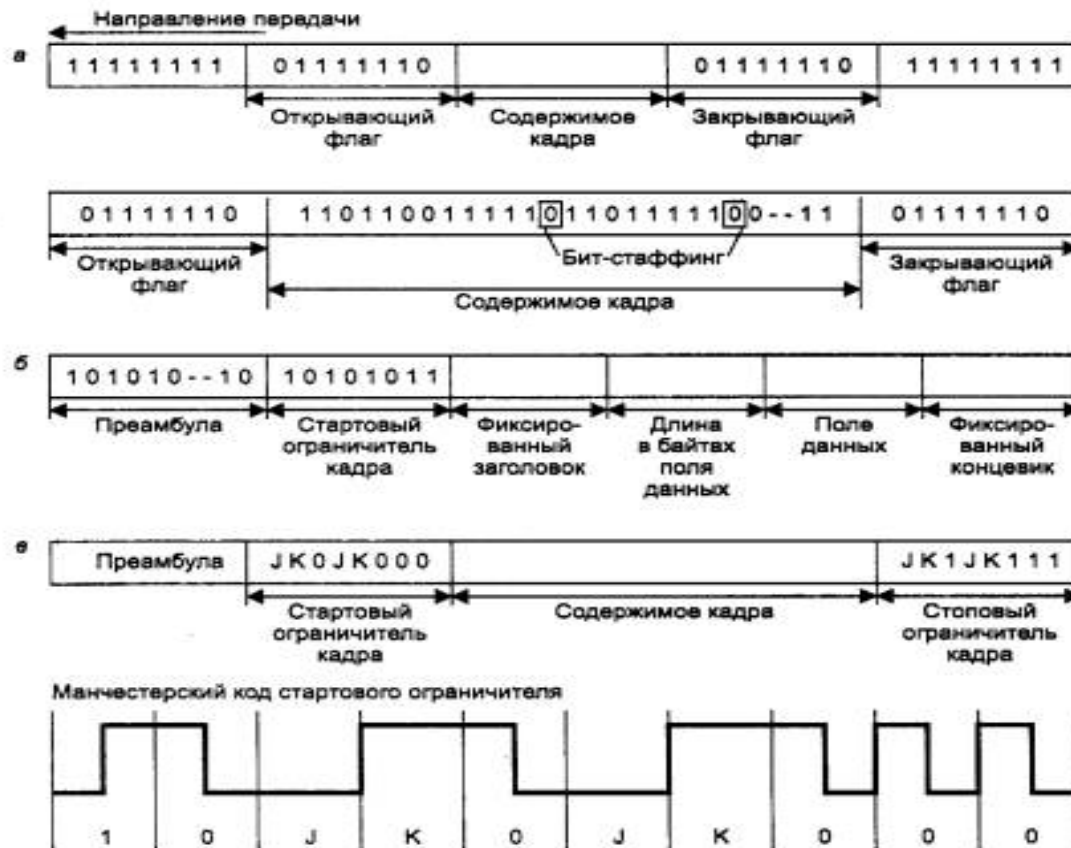
Канальный уровень (Data link layer)

Методы передачи данных:

- асинхронный/синхронный
- символьно-ориентированный/бит-ориентированный
- с предварительным установлением соединения/дейтаграммный
- с обнаружением искаженных данных/без обнаружения
- с обнаружением потерянных данных/без обнаружения
- с восстановлением искаженных и потерянных данных/без восстановления
- с поддержкой динамической компрессии данных/без поддержки

Канальный уровень (Data link layer)

Выделение начала и конца кадра при передаче:



Канальный уровень (Data link layer)

Цели использования процедуры установления соединения:

- Для взаимной аутентификации либо пользователей, либо оборудования
- Для согласования изменяемых параметров протокола: MTU, различных тайм-аутов
- Для обнаружения и коррекции ошибок
- В некоторых технологиях процедуру установления логического соединения используют при динамической настройке коммутаторов сети для маршрутизации всех последующих кадров

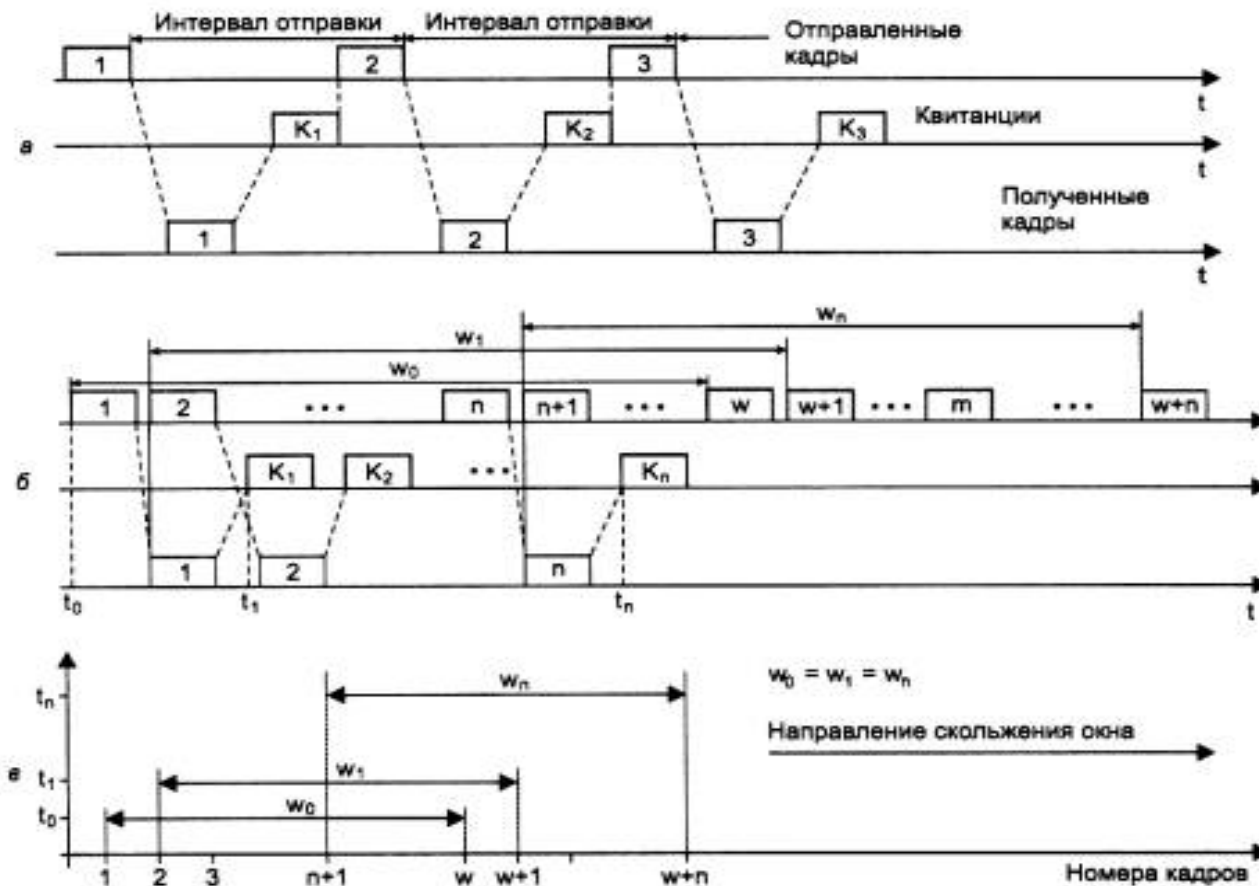


а – без установления соединения

б – с установлением соединения

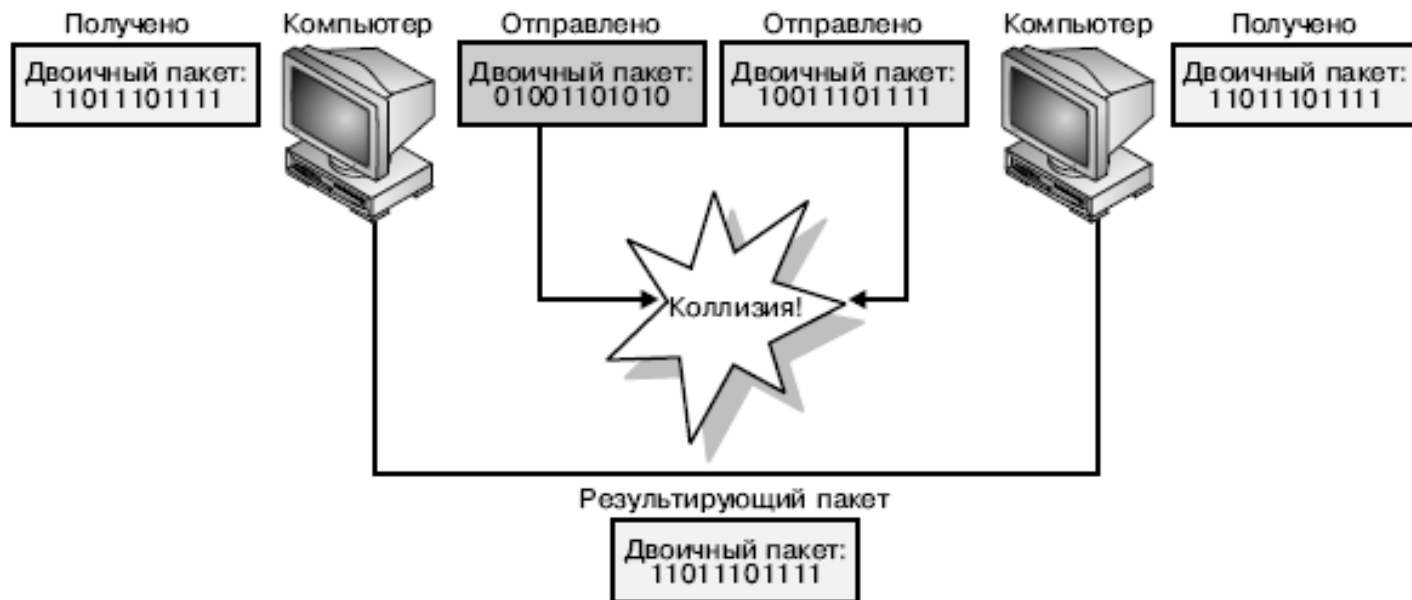
Канальный уровень (Data link layer)

Восстановление искаженных и потерянных кадров:



Канальный уровень (Data link layer)

Так же одной из самых важных задач канального уровня - это определение доступности среды:



Сетевой уровень (Network layer)

- Предназначен для образования единой транспортной системы из нескольких сетей – составной сети
- **Технология межсетевого взаимодействия** (internetworking) позволяет соединять в единую сеть множество сетей, построенных на основе разных технологий
- **Функции сетевого уровня** реализуются:
 - Группой протоколов:
 - IP (Internet Protocol) протокол Internet, сетевой протокол стека TCP/IP, который предоставляет адресную и маршрутную информацию
 - Протоколы маршрутизации (RIP, OSPF)
 - Маршрутизаторами (физически соединяют сети)
- **Пакет** – протокольная единица сетевого уровня (заголовок + поле данных).



Транспортный уровень (Transport layer)

- Обеспечивает приложениям или верхним уровням стека передачу данных с той степенью надежности, которая им требуется
- **Пять классов транспортного сервиса** отличаются качеством услуг:
 - Срочность
 - Возможность восстановления прерванной связи
 - Мультиплексированием соединений прикладного уровня через один общий транспортный протокол
 - Обнаружение и исправление ошибок передачи
- **Выбор качества сервиса** транспортного уровня зависит от:
 - Необходимой протоколам прикладного уровня надежности
 - Качеством каналов передачи данных (качество сетевого, канального и физического уровней)



Сеансовый уровень (Session layer)

- Управляет **взаимодействием сторон**:
 - Установление и завершение соединения между взаимодействующими сторонами
 - Какая сторона активна в данный момент
 - Средства синхронизации сеанса – контрольные точки в длинном сеансе. Если нарушен обмен, начинаем с последней точки, а не с начала
 - Извещение прикладных процессов об исключительных ситуациях
- Редко реализован в виде отдельного протокола. Функции данного уровня часто объединяются с функциями прикладного уровня в одном протоколе



Уровень представления (Presentation layer)

- Представление информации по сети без изменения ее содержимого. Передаваемая информация всегда будет понятна другой стороне:
 - **Проблемы кодировки данных** (изображений, MIME и т.п.)
 - **Шифрование/дешифрование** (secure socket layer – слой защищенных сокетов, SSL)
- **Уровень представления** отвечает за возможность диалога между приложениями на разных машинах. Этот уровень обеспечивает преобразование данных (кодирование, компрессия и т.п.) прикладного уровня в поток информации для транспортного уровня. Протоколы уровня представления обычно являются составной частью функций трех верхних уровней модели



Прикладной уровень (Application layer)

- **Прикладной уровень** отвечает за доступ приложений в сеть. Задачами этого уровня является перенос файлов, обмен почтовыми сообщениями и т.п.
- **Сообщение** – единица данных прикладного уровня
- **Наиболее распространённые протоколы:**
 - FTP - протокол переноса файлов
 - TFTP - упрощенный протокол переноса файлов
 - X.400 - электронная почта
 - Telnet
 - SMTP - простой протокол почтового обмена
 - SNMP - простой протокол управления сетью
 - NFS - сетевая файловая система
 - HTTP – протокол передачи гипертекста



Стек TCP/IP

- Стек **TCP/IP** был разработан по инициативе Министерства обороны США в 1970-х гг для связи экспериментальной сети ARPAnet с другими сетями как набор общих протоколов для разнородной вычислительной среды. Большой вклад в развитие стека TCP/IP, который получил свое название по популярным протоколам IP и TCP, внес университет Беркли, реализовав протоколы стека в своей версии ОС UNIX
- В стеке TCP/IP **4 уровня**:
 - Прикладной
 - Транспортный
 - Сетевой (межсетевой)
 - Уровень сетевого доступа

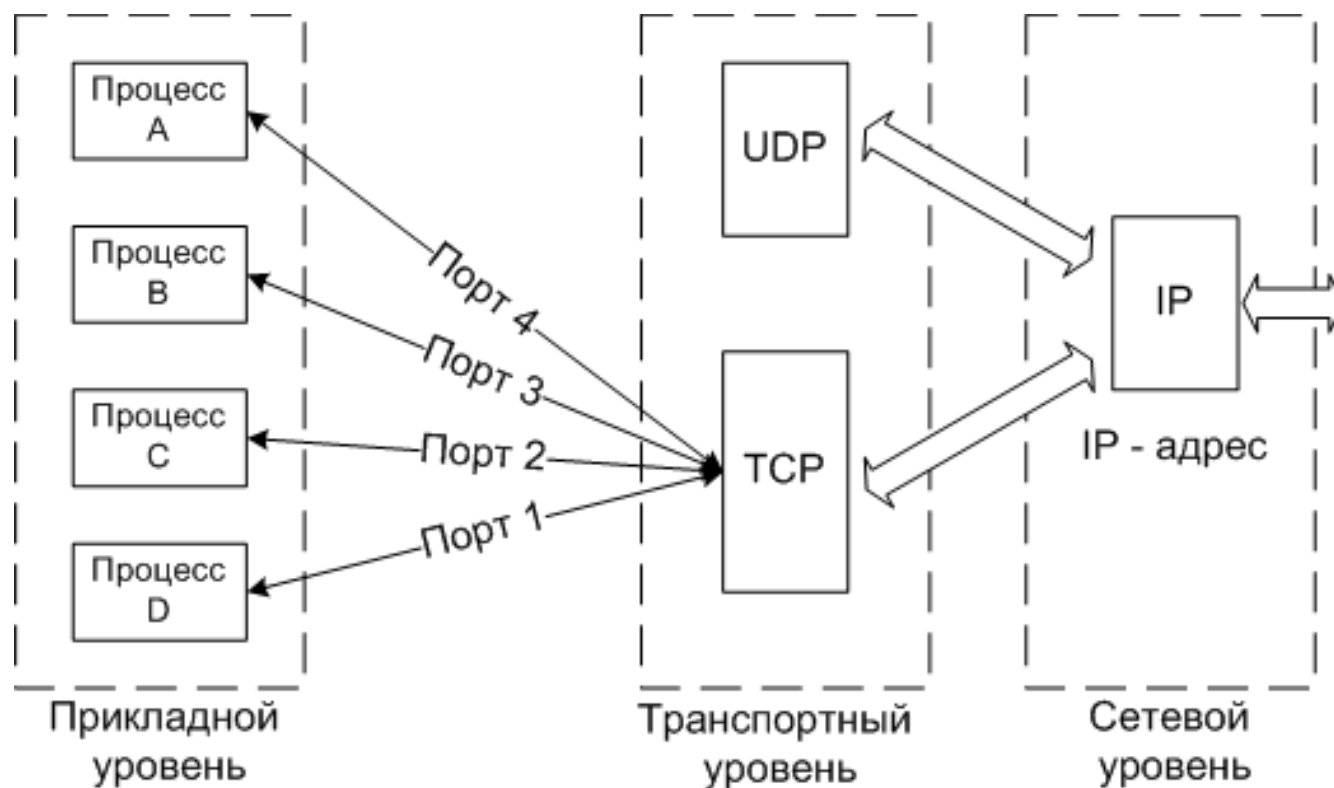
Соответствие между TCP/IP и OSI

7	www, Gopher, WAIS	SNMP	FTP	telnet	SMTP	TFTP	I
6							
5	TCP					UDP	II
4							
3	IP	ICMP	RIP	OSPF	ARP		III
2	Не регламентируется Ethernet, Toking Ring, FDDI, X.25, SLIP, PPP						IV
1							

Уровни
модели OSI

Уровни
стека
TCP/IP

Схема мультиплексирования информационных потоков





Основные функции протокола IP

IP — это протокол межсетевого взаимодействия (Internet Protocol)

Основные задачи IP:

- Передача пакетов между сетями в режиме дейтаграмм
- Динамическая фрагментация пакетов

Протокол IP относится к протоколам **без установления соединений**. Перед IP не ставится задача надежной доставки сообщений от отправителя к получателю. Протокол IP обрабатывает каждый IP-пакет как **независимую единицу**, не имеющую связи ни с какими другими IP-пакетами

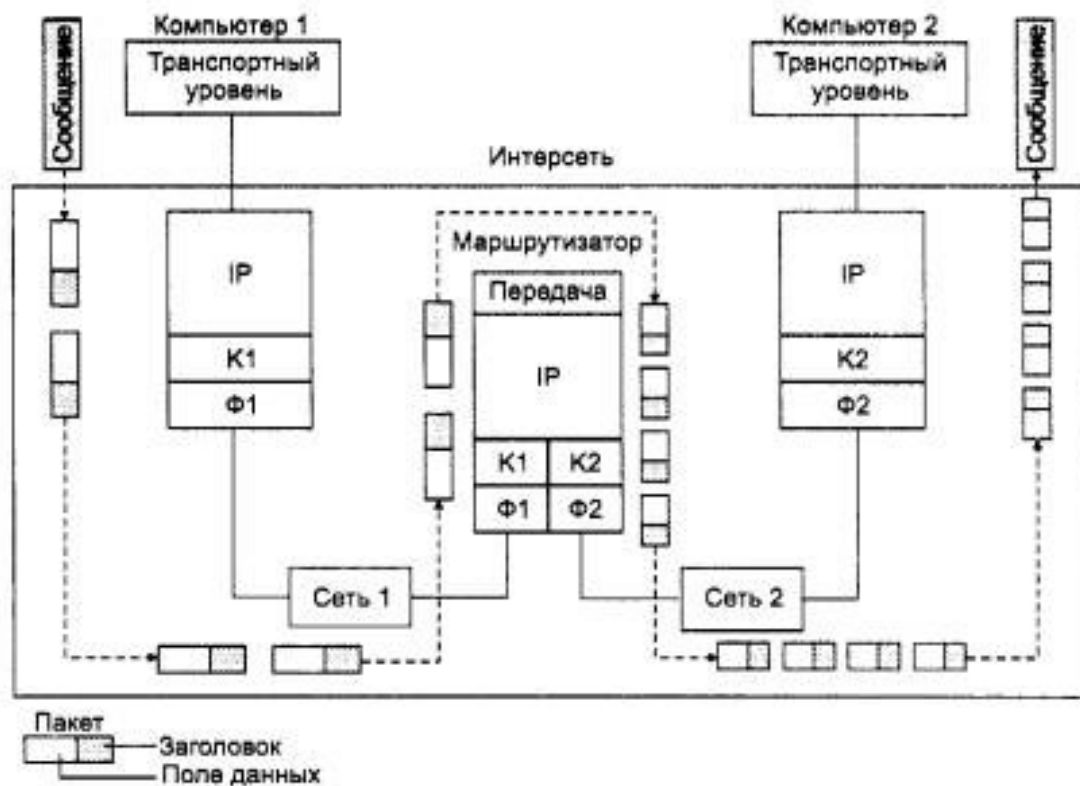
Структура IP-пакета

4 бита Номер версии	4 бита Длина заголовка	8 бит Тип сервиса					16 бит Общая длина									
		PR	D	T	R											
16 бит Идентификатор пакета								3 бита Флаги	13 бит Смещение фрагмента							
																D
8 бит Время жизни		8 бит Протокол верхнего уровня					16 бит Контрольная сумма									
32 бита IP-адрес источника																
32 бита IP-адрес назначения																
Опции и выравнивание																

Тип сервиса (Type of Service):

- **Поле приоритета** пакета (Precedence): может иметь значения от самого низкого - 0 (нормальный пакет) до самого высокого - 7 (пакет управляющей информации)
- Бит **D** говорит о том, что маршрут должен выбираться для минимизации задержки доставки данного пакета
- Бит **T** - для максимизации пропускной способности
- Бит **R** - для максимизации надежности доставки

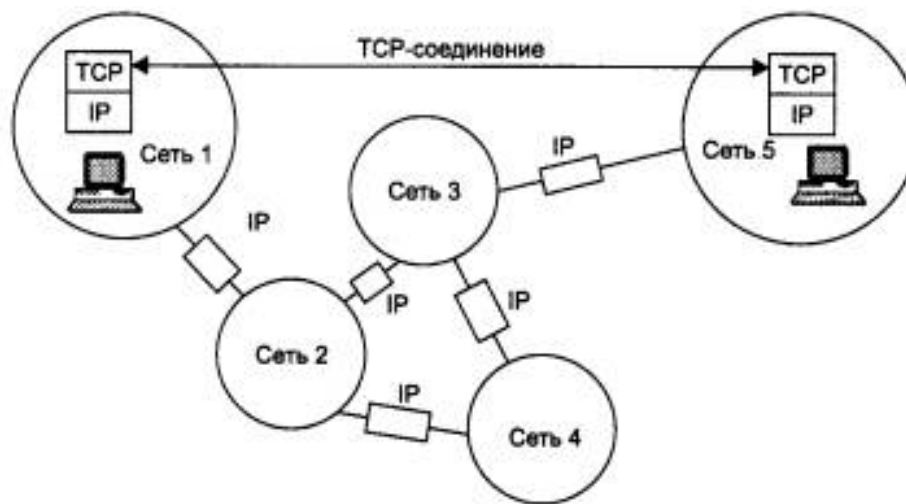
Фрагментация IP-пакетов



Протокол TCP

TCP (Transmission Control Protocol) — это транспортный механизм, предоставляющий поток данных, с предварительной установкой соединения, за счёт этого дающий уверенность в достоверности получаемых данных, осуществляет повторный запрос данных в случае потери данных и устраняет дублирование при получении двух копий одного пакета.

В отличие от **UDP** гарантирует целостность передаваемых данных и уведомление отправителя о результатах передачи.



Описание сетевого соединения

Сокет отправителя =

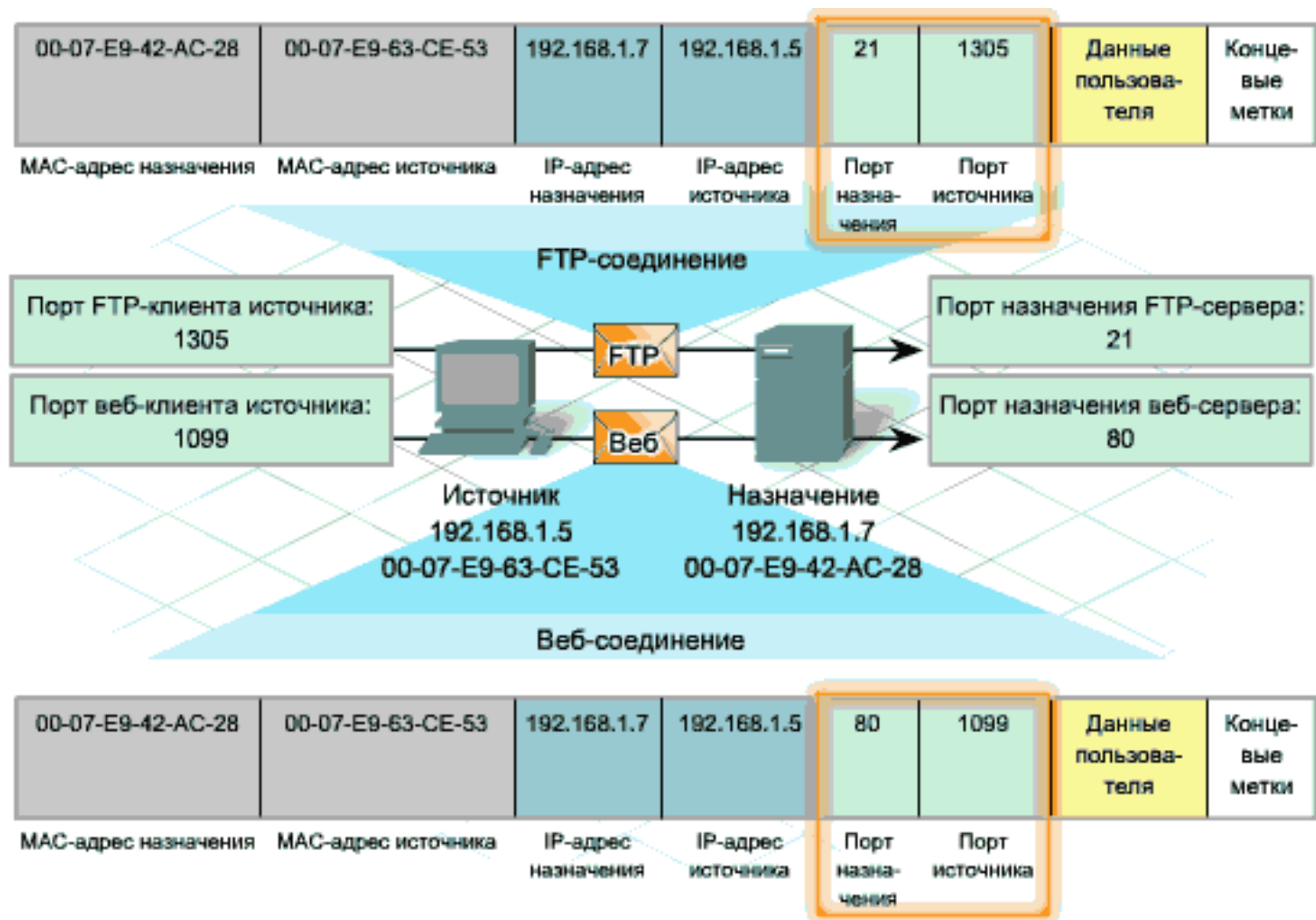
IP-адрес отправителя (IP_S) + номер порта отправителя (P_S)

Сокет адресата =

IP-адрес адресата (IP_D) + номер порта адресата (P_D)

Ансамбль **$IP_S P_S + IP_D P_D$** уникально описывает сокет

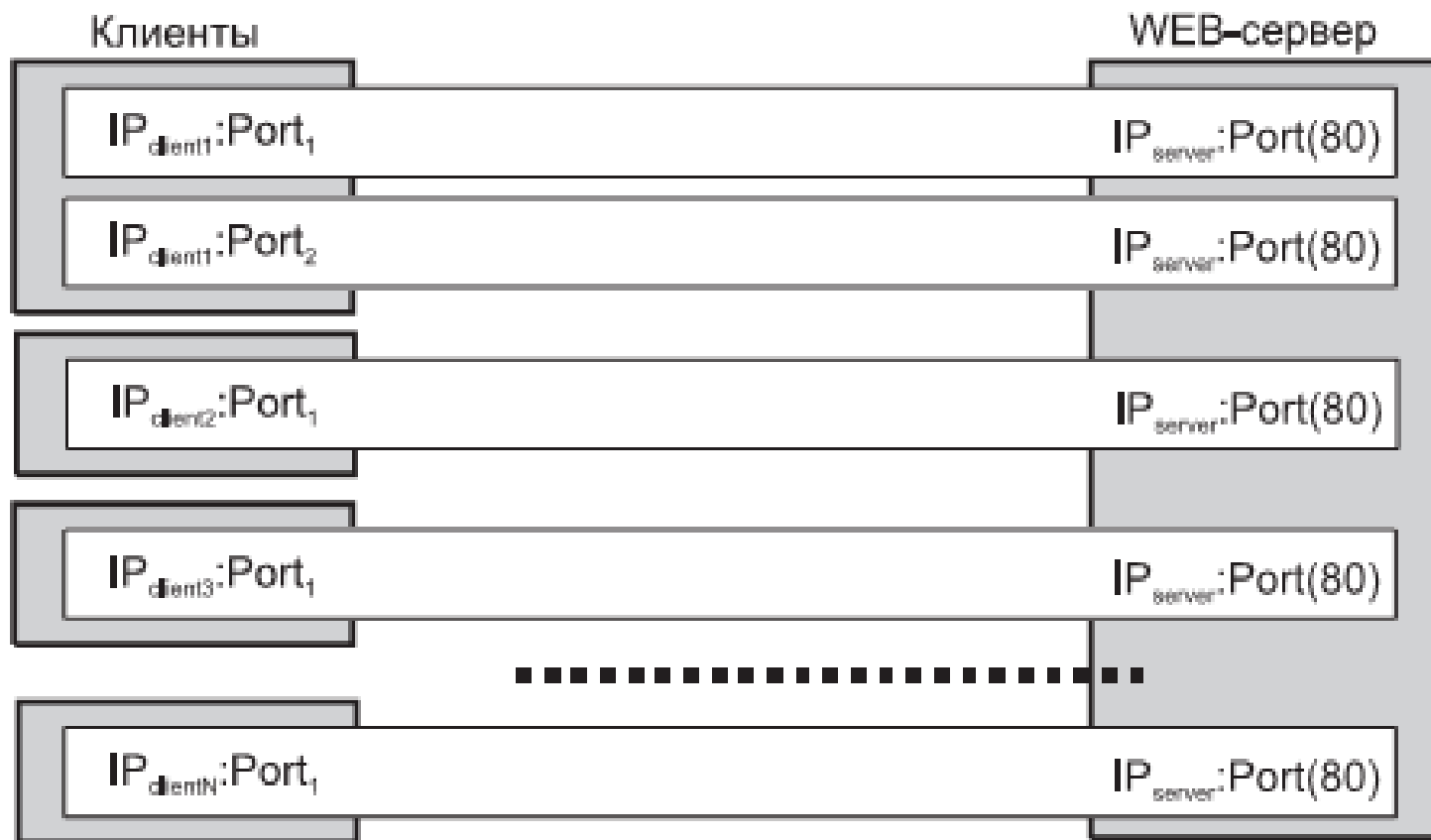
Инкапсуляция в стеке TCP/IP



Стандартные порты TCP

ftp-data	20/tcp	File Transfer [Default Data]
ftp	21/tcp	File Transfer [Control]
ssh	22/tcp	SSH Remote Login Protocol
telnet	23/tcp	Telnet
smtp	25/tcp	Simple Mail Transfer Protocol
domain	53/udp	Domain Name Server
finger	79/tcp	
http	80/tcp	World Wide Web HTTP(8000,8080)
pop3	110/tcp	Post Office Protocol(Ver 3)
auth	113/tcp	ident tap #Authentication Service
nntp	119/tcp	#Network News Transfer Protocol
netbios-ns	137/udp	NETBIOS Name Service
netbios-dgm	138/udp	NETBIOS Datagram Service
netbios-ssn	139/udp	NETBIOS Session Service
imap4	143/tcp	Interim Mail Access Pr v4
snmp	161/udp	SNMP

Пример TCP соединений



Клиент-серверное приложение

