ТЕМА 5. Сетевые протоколы

5.1 Многоуровневые модели.

Эталонная модель OSI — это описательная схема сети; ее стандарты гарантируют высокую совместимость и способность к взаимодействию различных типов сетевых технологий:

- иллюстрирует процесс перемещения информации по сетям;
- определяет сетевые функции, реализуемые на каждом ее уровне.

Модель OSI описывает, каким образом информация проделывает путь через сетевую среду (например, провода) от одной прикладной программы (например, программы обработки таблиц) к другой прикладной программе, находящейся в другом подключенном к сети компьютере.

Модель взаимодействия открытых систем.

Сложность сетевых структур и разнообразие телекоммуникационных устройств, выпускаемых различными фирмами, привели к необходимости стандартизации как устройств, так и процедур обмена данными между пользователями.

Международная организация стандартов (International Standards Organization – ISO) создала эталонную модель взаимодействия открытых систем (Open System Interconnection reference model – OSI), которая определяет концепцию и методологию создания сетей передачи данных.

Модель описывает стандартные правила функционирования устройств и программных средств при обмене данными между узлами (компьютерами) в открытой системе.

Открытая система состоит из программно-аппаратных средств, способных взаимодействовать между собой при использовании стандартных правил и устройств сопряжения (интерфейсов).

Модель ISO/OSI включает семь уровней.

Совокупность правил, по которым происходит обмен данными между программно-аппаратными средствами, находящимися на одном уровне, называется протоколом.

Набор протоколов называется стеком протоколов и задается определенным стандартом.

Взаимодействие между уровнями определяется стандартными интерфейсами.

Рисунок 5.1. Модель взаимодействия двух устройств: узла источника (source) и узла назначения (destination).

Стеки протоколов.

Стек протоколов Интернета. Стек протоколов сети Интернет был разработан до модели OSI. Поэтому уровни в стеке протоколов Интернета не соответствуют аналогичным уровням в модели OSI.

Стек протоколов Интернета состоит из пяти уровней:

1) физического,
2) звена передачи данных,
3) сети,
4) транспортного,
5) прикладного.

Первые четыре уровня обеспечивают физические стандарты, сетевой интерфейс, межсетевое взаимодействие и транспортные функции, которые соответствуют первым четырем уровням модели OSI.

Три самых верхних уровня в модели OSI представлены в стеке протоколов Интернета единственным уровнем, называемым прикладным уровнем

Стек базовых протоколов Интернета — иерархический, составленный из диалоговых модулей, каждый из которых обеспечивает заданные функциональные возможности; но эти модули не обязательно взаимозависимые.

В отличие от модели OSI, где определяется строго, какие функции принадлежат каждому из ее уровней, уровни набора протокола TCP/IP содержат относительно независимые протоколы, которые могут быть смешаны и согласованы в зависимости от потребностей системы.

Термин иерархический означает, что каждый верхний протокол уровня поддерживается соответственно одним или более протоколами нижнего уровня.

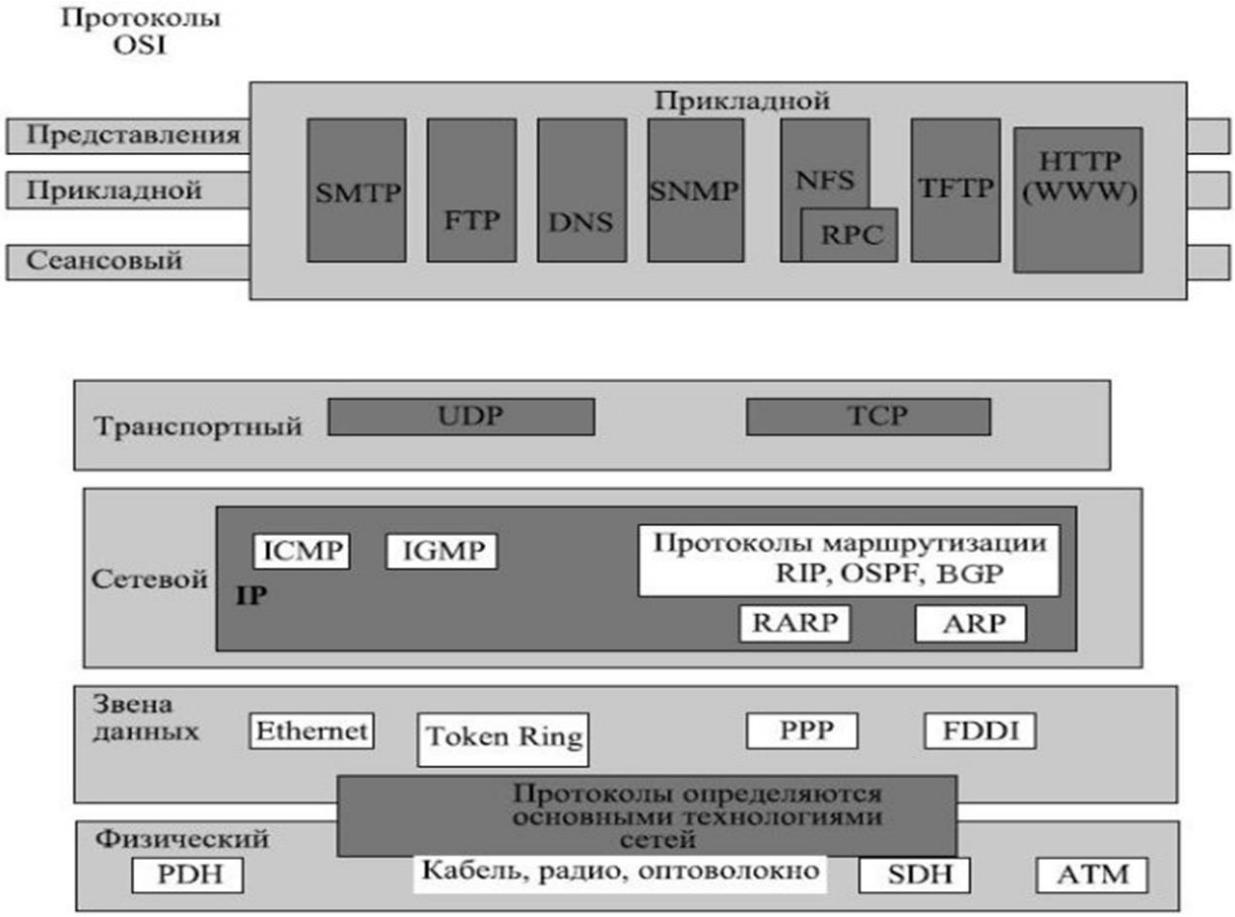


Рисунок 5.2. Стек протоколов Интернета по сравнению с OSI

Таблица 5.1. Протоколы Интернет

ARP	Address Resolution Protocol	Протокол нахождения адреса
ATM	Asynchronous Transfer Mode	Режим асинхронной передачи
BGP	Border Gateway Protocol	Протокол пограничной маршрутизации
DNS	Domain Name System	Система доменных имен
Ethernet	Ethernet Network	Сеть Ethernet
FDDI	Fiber Distributed Data Interface	Волоконно-оптический распределенный интерфейс данных
HTTP	Hyper Text Transfer Protocol	Протокол передачи гипертекста
FTP	File transfer Protocol	Протокол передачи файлов
ICMP	Internet Control Message Protocol	Протокол управляющих сообщений
IGMP	Internet Group Management Protocol	Протокол управления группами (пользователей) в Интернете
IP	Internet Protocol	Межсетевой протокол
NFS	Network File System	Протокол сетевого доступа к файловым системам
OSPF	Open Shortest Path First	Открытый протокол предпочтения кратчайшего канала
PDH	Plesiochronous Digital Hierarchy	Плезиохронная цифровая иерархия
PPP	Point-to- Point Protocol	Протокол связи "точка-точка"

RARP	Reverse Address Resolution Protocol	Протокол обратной конвертации адресов
RIP	Routing Information Protocol	Протокол обмена маршрутной информацией
RPC	Remote Procedure Call	Дистанционный вызов процедур
SMTP	Simple Mail Transfer Protocol	Простой протокол передачи почты
SDH	Synchronous Digital Hierarchy	Синхронная цифровая иерархия
SNMP	Simple Network Management Protocol	Простой протокол управления сетью
TCP	Transmission Control Protocol	Протокол управления передачей
TFTP	Trivial File Transfer Protocol	Простейший протокол передачи данных
TR	Token Ring	Маркерное кольцо
UDP	User Datagram Protocol	Дейтаграммный протокол пользователя
WWW	World Wide Web	Мировая паутина

Преимущества многоуровневых моделей:

Разбиение работы сети на подзадачи, что облегчает организацию и функционирование сети

Обеспечение совместимости сетевых продуктов разных производителей

Создание отдельных модулей, каждый из которых исполняет некоторый комплекс операций

Внесение изменений в отдельные модули, не затрагивая при этом другие, что ускоряет модернизацию отдельных частей сети.

Модель OSI делит задачу перемещения информации между компьютерами через сетевую среду на семь более легко разрешимых подзадач.

Прикладная и транспортная подсистемы модели OSI

Протоколы четырех нижних уровней обобщенно называют сетевым транспортом или транспортной подсистемой, так как они полностью решают задачу транспортировки сообщений с заданным уровнем качества в составных сетях с произвольной топологией и различными технологиями.

Остальные три верхних уровня решают задачи предоставления прикладных сервисов на основании имеющейся транспортной подсистемы.

5.2. Уровень 7 (прикладной уровень)

Самый близкий к пользователю уровень OSI.

Не обеспечивает услуг ни одному из других уровней OSI;

Обеспечивает услугами прикладные процессы, лежащие за пределами масштаба модели OSI.

Примерами таких прикладных процессов могут служить программы обработки крупномасштабных таблиц, программы обработки слов, программы банковских терминалов и т.д.

Прикладной уровень. ФУНКЦИИ:

Идентифицирует и устанавливает наличие предполагаемых партнеров для связи;

Синхронизирует совместно работающие прикладные программы;

Устанавливает соглашение по процедурам устранения ошибок и управления целостностью информации;

Определяет, имеется ли в наличии достаточно ресурсов для предполагаемой связи.

5.3 Уровень 6 (Представительный уровень)

Отвечает за то, чтобы информация, посылаемая из прикладного уровня одной системы, была читаемой для прикладного уровня другой системы.

Данный уровень имеет дело с формой представления передаваемой по сети информации, не меняя при этом ее содержания.

С помощью средств данного уровня протоколы прикладных уровней могут преодолеть синтаксические различия в представлении данных или же различия в кодах символов, например в кодах ASCII и EBCDIC.

На этом уровне может выполняться шифрование и дешифрование данных, благодаря которому секретность обмена данными обеспечивается сразу для всех прикладных служб.

ФУНКЦИИ

Уровень представления — согласовывает представление (синтаксис) данных при взаимодействии двух прикладных процессов:

преобразование данных из внешнего формата во внутренний;

шифрование и расшифровка данных.

Примером такого протокола является протокол

Secure Socket Layer (SSL), который обеспечивает секретный обмен сообщениями для протоколов прикладного уровня стека TCP/IP.

SSL (англ. *secure sockets layer* — уровень защищённых сокетов) — криптографический протокол, который подразумевает более безопасную связь.

Он использует асимметричную криптографию для аутентификации ключей обмена, симметричное шифрование для сохранения конфиденциальности, коды аутентификации сообщений для целостности сообщений.

Протокол широко использовался для обмена мгновенными сообщениями и передачи голоса через IP (англ. *Voice over IP* — VoIP) в таких приложениях, как электронная почта, Интернет-факс и др.

5.4 Уровень 5 (Сеансовый уровень)

Предназначен для организации и синхронизации диалога и управления обменом данными. Предоставляет услуги по установлению сеансового соединения между двумя представительными объектами и поддержанию упорядоченного взаимодействия при обмене данными между ними.

Для осуществления передачи данных между представительными объектами сеанс использует транспортное соединение. Сеанс может быть расторгнут сеансовыми или представительными объектами. Выполняется управление диалогом между взаимодействующими процессами, т.е. регулируется, какая из сторон осуществляет передачу, когда, как долго и т.д.

Функции сеансового уровня:

- установление и расторжение сеансового соединения
- обмен нормальными и срочными данными
- управление взаимодействием
- синхронизация сеанса
- восстановление сеанса

Синхронизацию между пользовательскими задачами сеансовый уровень обеспечивает посредством расстановки в потоке данных контрольных точек (chekpoints). (Для возобновления передачи данных в случае сетевой ошибки).

5.5 Протоколы верхнего уровня

На 3 верхних уровнях работают следующие протоколы:
HTTP
SMTP
Telnet
FTP
TFTP
SNMP и др.

HTTP

Протокол HTTP (Hypertext Transfer Protocol - Протокол передачи гипертекста) является протоколом уровня приложений.

HTTP был разработан для эффективной передачи по Интернету Web-страниц.

Протокол HTTP является основой системы World Wide Web.

SMTP

Протокол SMTP (Simple Mail Transfer Protocol — Простой протокол пересылки почты) обеспечивает механизм передачи электронной почты.

Главной целью протокола SMTP является надежная и эффективная доставка электронных почтовых сообщений.

SMTP - это довольно независимая субсистема, требующая только надежного канала связи.

Telnet

Telnet — стандартный протокол эмуляции.

Протокол Telnet используется для организации соединений с удаленного терминала и позволяет пользователям входить в удаленную систему и

использовать ее ресурсы так, словно они подключены к локальной системе.

FTP

Протокол FTP (File Transfer Protocol – Протокол передачи файлов) – обеспечивает способ перемещения файлов между компьютерными системами.

FTP реализует удаленный доступ к файлу.

Для того чтобы обеспечить надежную передачу, FTP использует в качестве транспорта протокол с установлением соединений — TCP.

Кроме пересылки файлов протокол, FTP предлагает и другие услуги.

Так пользователю предоставляется возможность интерактивной работы с удаленной машиной, например, он может распечатать содержимое ее каталогов, FTP позволяет пользователю указывать тип и формат запоминаемых данных.

В стеке TCP/IP протокол FTP предлагает наиболее широкий набор услуг для работы с файлами, однако он является и самым сложным для программирования.

FTP

TFTP (Trivial File Transfer Protocol)

Приложения, которым не требуются все возможности FTP, могут использовать другой, более экономичный протокол — простейший протокол пересылки файлов TFTP (Trivial File Transfer Protocol).

Этот протокол реализует только передачу файлов, причем в качестве транспорта используется более простой, чем TCP, протокол без установления соединения — UDP.

5.6 Уровень 4 (Транспортный уровень)

Транспортный уровень обеспечивает приложениям или верхним уровням стека — прикладному и сеансовому — передачу данных с той степенью надежности, которая им требуется.

Модель OSI определяет пять классов сервиса, предоставляемых транспортным уровнем, отличающихся качеством предоставляемых услуг:

срочностью, возможностью восстановления прерванной связи;

наличием средств мультиплексирования нескольких соединений между различными прикладными протоколами через общий транспортный протокол;

способностью к обнаружению и исправлению ошибок передачи, таких как искажение, потеря и дублирование пакетов.

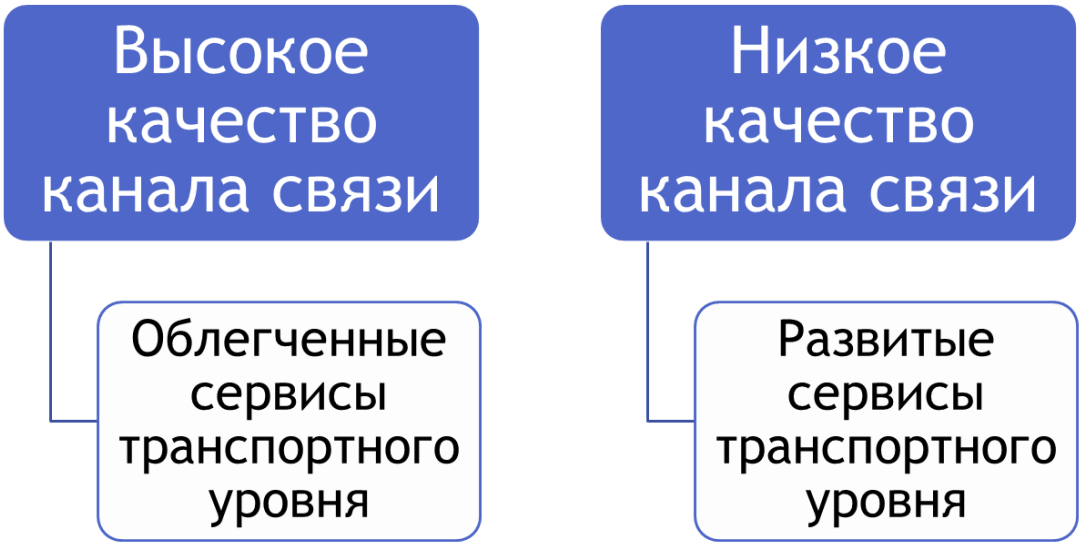


Рисунок 5.3. Критерии выбора класса сервиса транспортного уровня

Транспортный уровень. ФУНКЦИИ.

Транспортный уровень — обеспечение доставки информации с требуемым качеством между любыми узлами сети:

разбиение сообщения сеансового уровня на сегменты, их нумерация

буферизация принимаемых сегментов

упорядочивание прибывающих сегментов

адресация прикладных процессов (с помощью присвоения номера порта, по которому идентифицируется приложение, работающее на 7 уровне)

управление потоком

На данном уровне работают протоколы UDP, TCP.

Протоколы транспортного уровня: UDP, TCP.

TCP - один из основных сетевых протоколов Интернета, предназначенный для управления передачей данных в сетях и подсетях TCP/IP.

UDP (User Datagram Protocol — протокол пользовательских датаграмм) — это транспортный протокол для передачи данных в сетях IP без установления соединения. Он является одним из самых простых протоколов транспортного уровня модели OSI

Протокол TCP.

Протокол TCP (Transmission Control Protocol - протокол управления передачей) — надежный протокол с установлением соединения.

Он отвечает за разбиение сообщений на сегменты, их сборку на станции в пункте назначения, повторную отсылку всего, что оказалось не полученным, и сборку сообщений из сегментов.

Протокол TCP обеспечивает виртуальный канал между приложениями конечных пользователей.

Протокол TCP может также поддерживать многочисленные одновременные диалоги высших уровней.

Рисунок 5.4. Структура заголовка TCP

Каждый TCP сегмент содержит номер порта источника и назначения, с помощью которых идентифицируются отправляющие и принимающие приложения.

Примеры портов: HTTP – 80, FTP – 20, 21, Telnet – 23, SMTP – 25.

Протокол UDP

Протокол UDP (User Datagram Protocol – протокол пользовательских дейтаграмм) является "ненадежным", неориентированным на установление соединения протоколом, он не предназначен для проверки доставки сегментов.

Протокол UDP намного проще, чем TCP; он полезен в ситуациях, когда мощные механизмы обеспечения надежности протокола TCP не обязательны.

Протокол UDP

Недостаточная надёжность протокола может выражаться как в потере отдельных пакетов, так и в их дублировании.

UDP используется при передаче *потокового видео, игр реального времени, в IP-телефонии,* а также некоторых других типов данных, чувствительных к задержкам.

Каждый UDP сегмент содержит *номер порта источника* и назначения, с помощью которых идентифицируются отправляющие и принимающие приложения.

Примеры портов: TFTP – 69, SNMP – 25, DHCP – 67, 68, DNS - 53.

5.7. Уровень 3 (Сетевой уровень)

Служит для образования единой транспортной системы, объединяющей несколько сетей, причем эти сети могут использовать различные принципы передачи сообщений между конечными узлами и обладать произвольной структурой связей. На этом уровне работают маршрутизаторы. При этом используются схемы логической адресации, которыми может управлять сетевой администратор. Этот уровень использует схему адресации протокола IP (наиболее распространенный), а также схемы адресации AppleTalk, DECNet, Vines и IPX.

Проблема выбора наилучшего пути называется маршрутизацией, и ее решение является одной из главных задач сетевого уровня.

Сообщения сетевого уровня принято называть пакетами (packet).

На сетевом уровне определяется два вида протоколов.

Сетевые протоколы (routed protocols) — реализуют продвижение пакетов через сеть. Пример данного вида протоколов: IP, AppleTalk, DECNet, Vines и IPX.

Протоколы обмена маршрутной информацией или просто протоколы маршрутизации (routing protocols).

С помощью этих протоколов маршрутизаторы собирают информацию о топологии межсетевых соединений.

Примеры этих протоколов: RIP, OSPF, BGP, IS-IS.

Протоколы сетевого уровня реализуются программными модулями операционной системы, а также программными и аппаратными средствами маршрутизаторов.

Также на данном уровне работают протоколы ICMP, ARP, RARP.

IP

Протокол IP (Internet Protocol — Интернет-протокол) обеспечивает маршрутизацию пакетов с негарантированной доставкой (best-effort delivery) без установки логического соединения (connectionless).

Рисунок 5.5. Структура IP пакета

Этот протокол не интересуется содержанием пакетов; он лишь ищет наилучший способ направить пакет к месту его назначения.

В каждом заголовке IP-пакета должен быть достоверный IP-адрес источника и адресата.

Без достоверной информации об адресе отправленные пакеты не попадут к узлу назначения. К источнику пакеты не вернутся.

Кроме того в заголовке находится контрольная информация с описанием пакета, предназначенная для сетевых устройств, например, маршрутизаторов. Эта информация помогает контролировать поведение пакета в сети.

В Интернете используются только уникальные IP-адреса.

Существуют организации, которые контролируют распределение IP-адресов и не допускают дублирования.

Интернет-провайдеры получают блоки IP-адресов от локального, национального или регионального Интернет-регистратора (RIR).

Интернет-провайдеры распоряжаются этими адресами и предоставляют их конечным пользователям.

AppleTalk

AppleTalk — стек протоколов, разработанных Apple Computer для компьютерной сети.

Он был изначально включён в Macintosh (1984), но потом компания отказалась от него в пользу TCP/IP.

IPX

Протокол IPX предназначен для передачи дейтограмм в системах, неориентированных на соединение (также как и IP или NETBIOS,

разработанный IBM и эмулируемый в Novell), он обеспечивает связь между NetWare серверами и конечными станциями.

Максимальный размер IPX-дейтограммы составляет 576 байт, из них 30 байта занимает заголовок. Предполагается, что сеть, через которую транспортируются эти дейтограммы, способна пересылать пакеты соответствующей длины.

BGP

BGP (англ. Border Gateway Protocol, протокол граничного шлюза) — динамический протокол маршрутизации внешнего шлюза (англ. EGP — External Gateway Protocol).

На текущий момент является основным протоколом динамической маршрутизации в сети Интернет.

Протокол BGP предназначен для обмена информацией о достижимости подсетей между автономными системами (АС, англ. AS — autonomous system

BGP, наряду с DNS, является одним из главных механизмов, обеспечивающих функционирование Интернета.

BGP является протоколом прикладного уровня и функционирует поверх протокола транспортного уровня TCP (порт 179).

5.8 Уровень 2 (Канальный уровень)

Задачи канального уровня (Data Link layer):

- ✓ проверка доступности среды передачи;
- ✓ реализация механизмов обнаружения и коррекции ошибок.

Для этого на канальном уровне биты группируются в наборы, называемые кадрами (frames).

Канальный уровень обеспечивает корректность передачи каждого кадра, помещая специальную последовательность бит в начало и конец каждого кадра, для его выделения, а также вычисляет контрольную сумму. Когда кадр приходит по сети, получатель снова вычисляет контрольную сумму полученных данных и сравнивает результат с контрольной суммой из кадра. В целом канальный уровень представляет собой весьма мощный набор функций по пересылке сообщений между узлами сети.

Канальный уровень может не только обнаруживать ошибки, но и исправлять их за счет повторной передачи поврежденных кадров. Функция исправления ошибок для канального уровня не является обязательной, поэтому в некоторых протоколах этого уровня она отсутствует, например в Ethernet и frame relay.

В локальных сетях протоколы канального уровня используются компьютерами, мостами, коммутаторами и маршрутизаторами. В компьютерах функции канального уровня реализуются совместными усилиями сетевых адаптеров и их драйверов.

В некоторых случаях протоколы канального уровня оказываются

самодостаточными транспортными средствами, и тогда поверх них могут работать непосредственно протоколы прикладного уровня или приложения, без привлечения средств сетевого и транспортного уровней. Однако, для обеспечения качественной транспортировки сообщений в сетях любых топологий и технологий функций канального уровня оказывается недостаточно, поэтому в модели OSI решение этой задачи возлагается на два следующих уровня — сетевой и транспортный.

Канальный уровень реализуется программно-аппаратно. Примерами протоколов канального уровня являются протоколы Ethernet, Token Ring, FDDI, 100VG-AnyLAN, PPP, Frame Relay, ATM.

Канальный уровень использует физическую адресацию, а именно MAC – адреса в технологии Ethernet,

DLCI (идентификатор подключения к соединению) в технологии Frame Relay,

VPI/VCI (идентификатор виртуального пути (номер канала)/ идентификатор виртуального канала (номер соединения)) в технологии ATM.

Ethernet

Ethernet — это самый распространенный на сегодняшний день стандарт локальных сетей, а также с недавнего времени широко внедряется на магистральных сетях.

Общее количество сетей, работающих по протоколу Ethernet в настоящее время, оценивается в несколько миллионов.

Ethernet работает на 2 нижних уровнях модели OSI: канальном и физическом.

В стандарте первых версий в качестве передающей среды используется коаксиальный кабель, в дальнейшем появилась возможность использовать витую пару и оптический кабель.

Ethernet в качестве физической адресации использует MAC-адреса.

В зависимости от скорости передачи данных и передающей среды существует несколько вариантов технологии:

10 Мбит/с Ethernet,

Быстрый Ethernet (Fast Ethernet, 100 Мбит/с),

Гигабитный Ethernet (Gigabit Ethernet, 1 Гбит/с),

10-гигабитный Ethernet,

40-гигабитный,

100-гигабитный Ethernet.

Технологические опции

Ethernet

Fast Ethernet

Gigabit Ethernet

10 Gig Ethernet

WLAN

Доступ к среде.

Ethernet и Wi-Fi - оба технологии “мультидоступа”

Используют широковещательный носитель, совместно использованный многими узлами.

Одновременные передачи приведут к коллизиям следовательно требуется протокол управления доступом к среде (MAC).

Правила о том, как совместно использовать носитель:

Канальный уровень разделен на два

Подуровень управления доступом к среде MAC Части

Подуровень LLC (Логическое Управление Ссылкой)

802.3 Ethernet

Множественный доступ с обнаружением несущей с обнаружением коллизий (CSMA/CD).

CS = обнаружение несущей;

MA = множественный доступ;

CD = обнаружение коллизий.

Основной стандарт Ethernet составляет 10 Мбит/с.

100 Мбит/с, 1 Гбит/с, стандарты на 10 Гбит/с прибыли позже

Ethernet CSMA/CD

CSMA/CD (множественный доступ с контролем несущей и обнаружением коллизий)

Используется протокол доступа к среде.

Данные переданы в форме пакетов.

Канал контролирует факт передачи пакета.

Пакет передается, только если канал неактивен; иначе – передача задерживается, пока канал не станет неактивным.

После того, как пакетная передача запущена, узел контролирует свою собственную передачу, чтобы видеть, испытал ли пакет коллизию.

Если пакет подвергается коллизии, передача прерывается и пакет ретранслируется после случайного интервала времени, с использованием Двоичного алгоритма Экспоненциальной задержки.

Ethernet Address

Конечные узлы идентифицированы их Ethernet-адресами (MAC-адрес или Аппаратный адрес), который является уникальным 6-байтовым адресом.

MAC-адрес представлен в Десятичном формате Неха, например, 00:05:5D:FE:10:0A

Первые 3 байта идентифицируют поставщика (также названный префиксом) и последние 3 байта уникальны для каждого узла или устройства

Структура кадра Ethernet

Преамбула: 7 байтов образца 10101010 и один байт образца 10101011.

Используется, чтобы синхронизировать тактовые частоты получателя с отправителем

Адреса: 6 байт, кадр принимается всеми адаптерами на LAN и отбрасывается если не соответствует

Длина: 2 байта, длина Поля данных

CRC: 4 байта - проверка ошибки на стороне получателя. Если ошибка обнаружена, фрейм просто отброшен

Полезная нагрузка данных: Максимальные 1500 байтов, минимальные 46 байт. Если данные составляют меньше чем 46 байт – дополняются нулями до 46 байт.

Типы кабеля/Топологии Ethernet:

10Base5 (Сеть с толстым коаксиальный кабель) (Шинная топология)

10Base2 (Сеть с тонким коаксиальный кабель) (Шинная топология)

10BaseT (UTP) CAT 3/5 (Звезда/Древовидная топология)

10BaseFL (Многорежимное / Одномодовое оптоволокно) (Звезда/Древовидная топология)

Максимальная длина сегмента Ethernet

10 Base5 - 500 м с самое большее 4 повторителями (Используют Мост, чтобы расширить сеть),

10 Base2 - 185 м с самое большее 4 повторителями (Используют Мост, чтобы расширить сеть),

10 BaseT - 100 м с самое большее 4 концентраторами (Используют Switch, чтобы расширить сеть),

Fast Ethernet

Пропускная способность до 100 Мбит/с

Используется тот же протокол доступа к среде CSMA/CD и пакет форматирует как в Ethernet.

Стандарты 100 BaseTX (UTP) и 100BaseFX (Волокно)

Физические среды:

100 BaseTX - CAT 5e UTP

100 BaseFX - Многорежимный / Одномодовое оптоволокно

Режимы: Полный дуплекс / Полудуплекс.

Fast Ethernet

Максимальная длина сегмента

100 BASE TX - 100 м

100 BASE FX - 2 км (многомодовое оптоволокно)

100 BASE FX - 20 км (одномодовое оптоволокно)

Gigabit Ethernet

Пропускная способность до 1 Гбит/с.

Используется тот же протокол доступа к среде CSMA/CD как в Ethernet и обратно совместим (10/100 модули доступны).

1000BaseT (UTP), 1000BaseSX (Многомодовое оптоволокно) и 1000BaseLX (Многорежимное / Одномодовое оптоволокно) стандарты.

Максимальная длина сегмента

1000BaseT - 100 м (CAT 5e/6)

1000BaseSX - 275 м (Многомодовое оптоволокно)

1000BaseLX - 512 м (Многомодовое оптоволокно)

1000BaseLX - 20 км (одномодовое оптоволокно)

1000BaseLH - 80 км (одномодовое оптоволокно)

10 Gig Ethernet

Пропускная способность до 10 Гбит/с.

Используется тот же протокол доступа к среде CSMA/CD как в Ethernet.

Предложен для метро Ethernet (магистральных)

Максимальная длина сегмента

1000Base-T - Не доступный

10GBase-LR - 10 км (одномодовое оптоволокно)

10GBase-ER - 40 км (одномодовое оптоволокно)

Token Ring

Token Ring (маркерное кольцо) - архитектура сетей с кольцевой логической топологией и детерминированным методом доступа, основанная на передаче маркера.

Кольцевая топология означает упорядоченную передачу информации от одной станции к другой в одном направлении, строго по порядку включения.

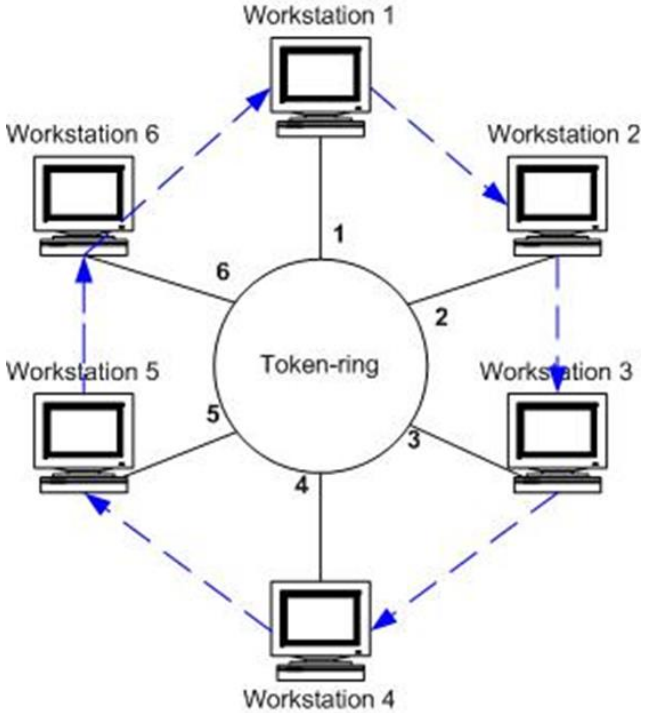


Рисунок 5.6. Token Ring

PPP

Протокол PPP (Point-to-point Protocol - Протокол "точка-точка") используется для установления прямой связи между двумя узлами сети, причем он может обеспечить аутентификацию соединения, шифрование и сжатие данных.

Используется на многих типах физических сетей: телефонная линия, сотовая связь и т. д.

Часто встречаются подвиды протокола PPP:

Point-to-Point Protocol over Ethernet (PPPoE), используемый для подключения по Ethernet, и иногда через DSL;

Point-to-Point Protocol over ATM (PPPoA), который используется для подключения по ATM, который является основной альтернативой PPPoE для DSL.

5.9. Уровень 1 (Физический уровень)

Физический уровень обеспечивает передачу потока бит в физическую среду передачи информации.

При этом используются такие физические передающие среды, как витые пары, коаксиальные и оптоволоконные кабели, радиоволны.

На этом уровне определяется:

физическая среда передачи – тип кабеля для соединения устройств;

механические параметры – количество контактов (пинов) (тип разъема);

электрические параметры (напряжение, длительность единичного импульса сигнала);

функциональные параметры (для чего используется каждый пин сетевого разъема, как устанавливается начальное физическое соединение и как оно разрывается).

Примерами реализации стандартов физического уровня являются RS-232, RS-449, RS-530 и множество спецификаций МСЭ-Т серии V и X (например, V.35, V.24, X.21).

Функции физического уровня:

1) передача битов по физическим каналам;
2) формирование электрических сигналов;
3) кодирование информации;
4) синхронизация;
5) модуляция.

Данный уровень реализуется аппаратно. Функции физического уровня реализуются во всех устройствах, подключенных к сети. Со стороны компьютера функции физического уровня выполняются сетевым адаптером или последовательным портом.

Процесс инкапсуляции данных

Эталонная модель OSI описывает процесс прохождения информации от прикладной программы (такой, например, как электронная почта) через передающую среду к другой прикладной программе, работающей на другом компьютере. Каждый уровень для осуществления обмена данными с соответствующим уровнем другой системы использует собственный протокол. При этом информация передается в виде модулей данных протокола (protocol data units, PDU).

Модели OSI не допускают непосредственной коммуникации между соответствующими уровнями разных хостов.

Рисунок 5.6. Обмен информацией между хостами OSI

Обмен информацией между хостами

На хосте А находится информация, которую нужно передать на хост В.

Рисунок 5.7. Обмен информацией между хостами

ПРИМЕР

Рассмотрим поэтапный процесс передачи электронного сообщения от хоста А к хосту Б.

В процессе инкапсуляции данных, позволяющей передать это сообщение по электронной почте, выполняются пять этапов преобразования.

Когда пользователь посылает электронное сообщение, буквенно-цифровые символы последовательно преобразуются в данные для передачи на 7, 6 и 5-м уровнях и после этого передаются в сеть.

Используя сегменты своего формата, транспортный уровень упаковывает данные для транспортировки их по сети и обеспечивает надежную связь между двумя хостами, участвующими в передаче и приеме электронного сообщения.

На 3-м уровне данные упаковываются в пакет (дейтаграмму), содержащий сетевой заголовок и логические адреса отправителя и получателя (IP-адреса). После этого сетевые устройства пересылают пакеты по сети, используя выбранный маршрутизатором путь.

На 2-м уровне каждое сетевое устройство должно вставить пакет в кадр (фрейм). Фрейм позволяет осуществить соединение со следующим сетевым устройством. Каждое устройство на выбранном сетевом пути требует создания фрейма для соединения со следующим устройством.

На 1-м уровне фрейм должен быть преобразован в последовательность нулей и единиц для прохождения по передающей среде. Механизм синхронизации позволяет различать между собой эти биты по мере того как они проходят через передающую среду. На различных участках сетевого пути тип

передающей среды может меняться.

5.10. Стеки протоколов

Важнейшим направлением стандартизации в области вычислительных сетей является стандартизация коммуникационных протоколов.

Протокол (коммуникационный) - это набор правил и процедур взаимодействия модулей одного уровня в разных узлах.

Интерфейс - это набор правил и процедур взаимодействия модулей соседних уровней в одном узле.

Стек протоколов – это иерархически организованный набор протоколов, достаточный для организации взаимодействия узлов в сети.

Стандартные стеки коммуникационных протоколов

В настоящее время в сетях используется большое количество стеков коммуникационных протоколов.

Наиболее популярны следующие стеки:

TCP/IP ;

IPX/SPX ;

NetBIOS/SMB ;

DECnet ;

SNA ;

OSI.

Практически все эти стеки на нижних уровнях — физическом и канальном, — используют одни и те же хорошо стандартизованные протоколы Ethernet, Token Ring, FDDI и ряд других, которые позволяют задействовать во всех сетях одну и ту же аппаратуру. Зато на верхних уровнях все стеки работают по своим протоколам.

Стандартные стеки коммуникационных протоколов

Эти протоколы часто не соответствуют рекомендуемому моделью OSI разбиению на уровни. В частности, функции сеансового и представительного уровня, как правило, объединены с прикладным уровнем. Такое несоответствие связано с тем, что модель OSI появилась как результат обобщения уже существующих и реально используемых стеков, а не наоборот.

Стек TCP/IP

Стек TCP/IP, является одним из наиболее популярных стеков коммуникационных протоколов. На этом стеке работает всемирная информационная сеть Internet. Так как стек TCP/IP был разработан до появления модели взаимодействия открытых систем OSI, то, хотя он также имеет многоуровневую структуру, соответствие уровней стека TCP/IP уровням модели OSI достаточно условно.

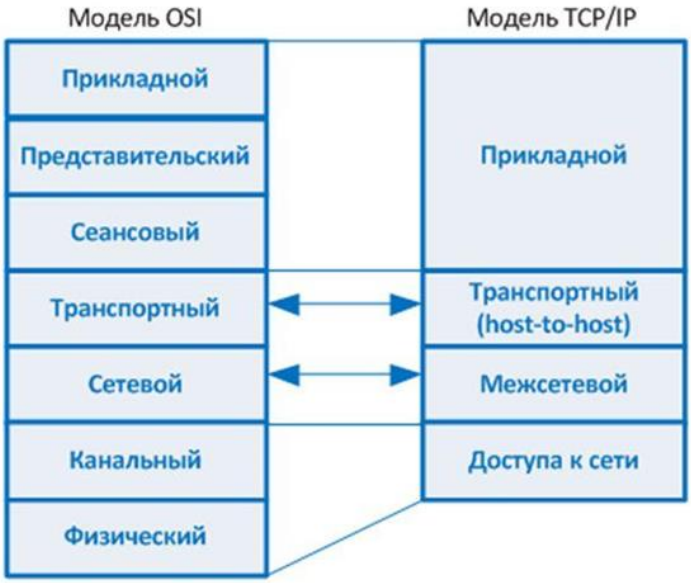


Рисунок 5.8. Условное соответствие уровней стеков OSI и TCP/IP

Протоколы TCP/IP делятся на 4 уровня

Уровень IV — уровень сетевых интерфейсов — соответствует физическому и канальному уровням модели OSI. Этот уровень в протоколах TCP/IP не регламентируется, но поддерживает все популярные стандарты физического и канального уровня.

Уровень III — это уровень межсетевого взаимодействия, который полностью соответствует сетевому уровню модели OSI.

Уровень II называется транспортным, который полностью соответствует транспортному уровню модели OSI.

Верхний уровень (уровень I) называется прикладным. Соответствует трем верхним уровням модели OSI.

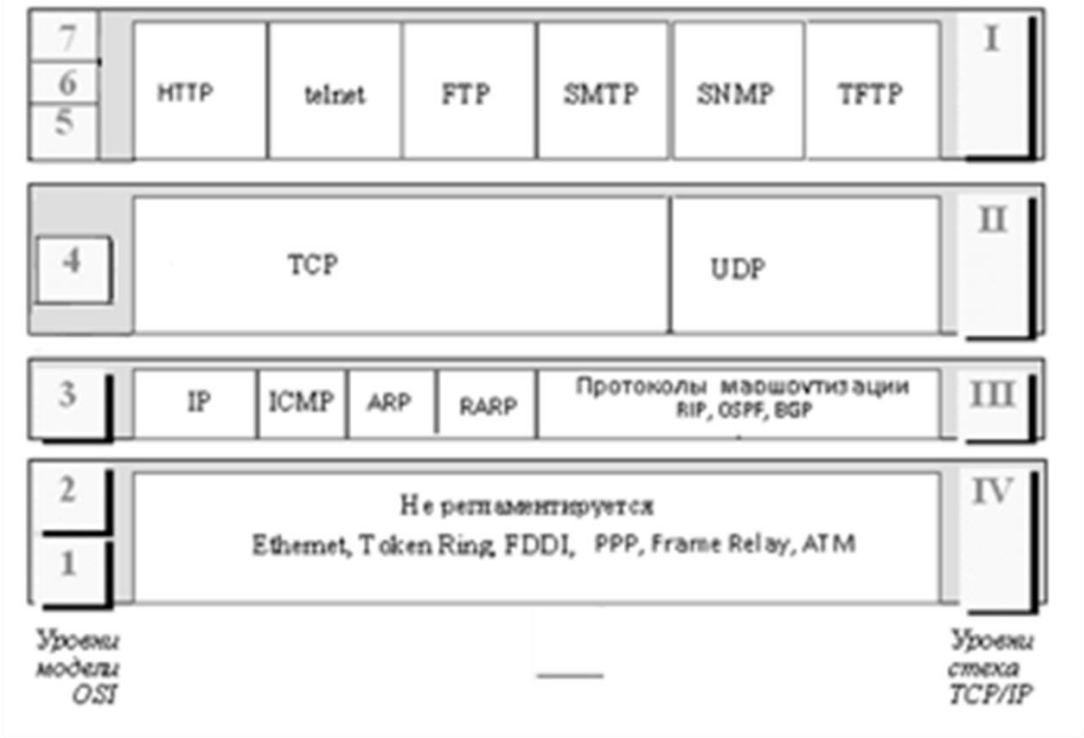


Рисунок 5.9. Стек TCP/IP

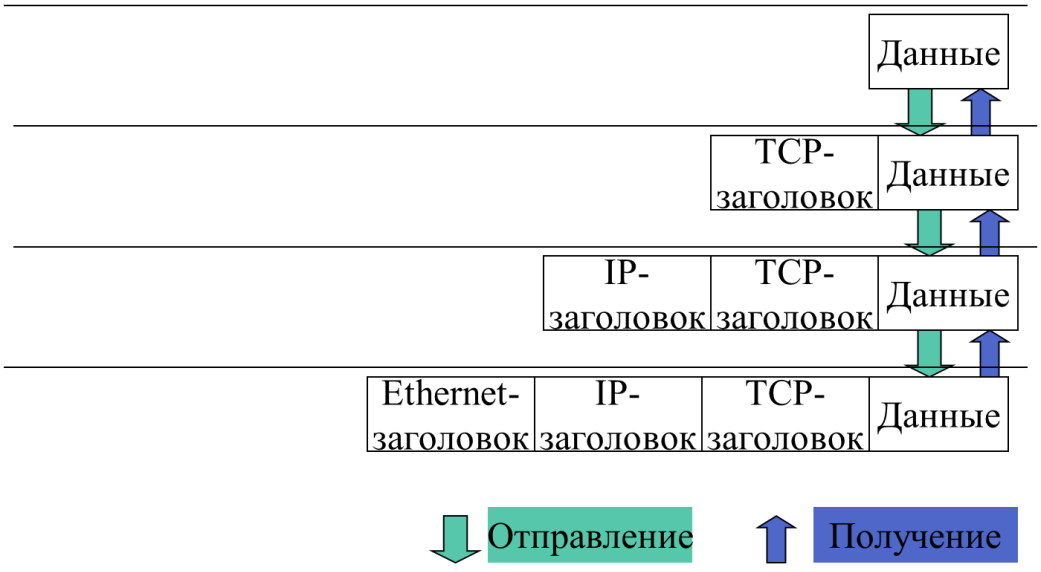


Рисунок 5.10. Схема инкапсуляции данных в стеке протоколов TCP/IP

Достоинства:

– наиболее популярный на сегодняшний день стек сетевых протоколов;

– этот стек используется для связи компьютеров Internet, а также в огромном числе корпоративных сетей;

– основными протоколами стека, давшими ему название, являются протоколы IP и TCP. IP обеспечивает продвижение пакета по составной сети, а TCP гарантирует надежность его доставки.

– стек TCP/IP вобрал в себя большое количество протоколов прикладного уровня: FTP, telnet, SMTP, HTTP и многие другие.

– стек TCP/IP изначально создавался для глобальной сети Internet, в связи с этим:

– способен фрагментировать пакеты;

– имеет гибкую систему адресации;

– экономно использует возможности широковещательных рассылок.

Недостатки:

– мощные функциональные возможности протоколов стека TCP/IP требуют для своей реализации высоких вычислительных затрат;

– гибкая система адресации и отказ от широковещательных рассылок приводят к наличию в IP-сети различных централизованных служб типа DNS, DHCP и т. п.

Стек SNA

SNA (Systems *Network Architecture*) является патентованной архитектурой компании IBM, созданной в 1974 году в соответствии с иерархической моделью построения сетей, которой в то время придерживалась IBM.

В эту иерархию включились мэйнфреймы (хост), коммуникационные контроллеры, кластерные (групповые) контроллеры и терминалы.

Уровневая архитектура и основные протоколы стека *SNA* в сопоставлении с архитектурой ISO/OSI представлены на рис.

Как видно, модель *SNA* напоминает модель ISO/OSI. Но функции в ней сгруппированы по-другому.

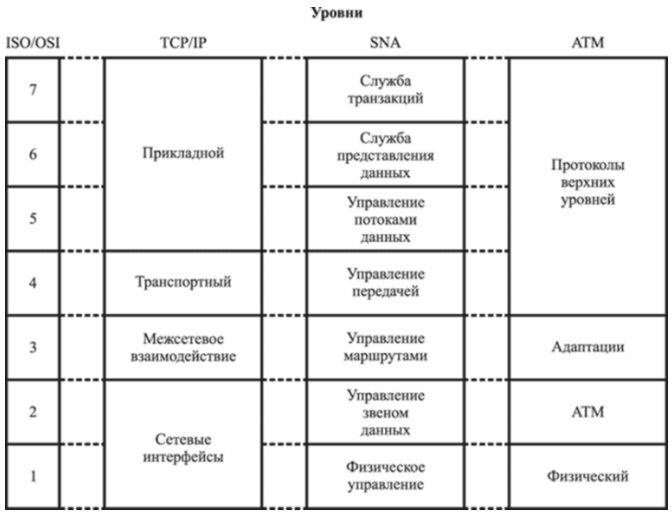


Рисунок 5.11. Стек SNA

Верхний уровень службы транзакций (Transaction Services) обеспечивает средства приложений для распределенной обработки и управления сетью. К прикладным протоколам относятся:

DIA (Document Interchange Architecture) - определяет стандарты обмена документами между разнородными вычислительными системами; координирует передачу файлов, поиск документов и их хранение;

SNADS (*SNA* Distributed Service) - управляет распространением документов и сообщений (инфраструктура для распространения электронной почты);

DDM (Distributed *Data Management*) - обеспечивает прозрачный удаленный доступ к файлам за счет механизма перенаправления запросов.

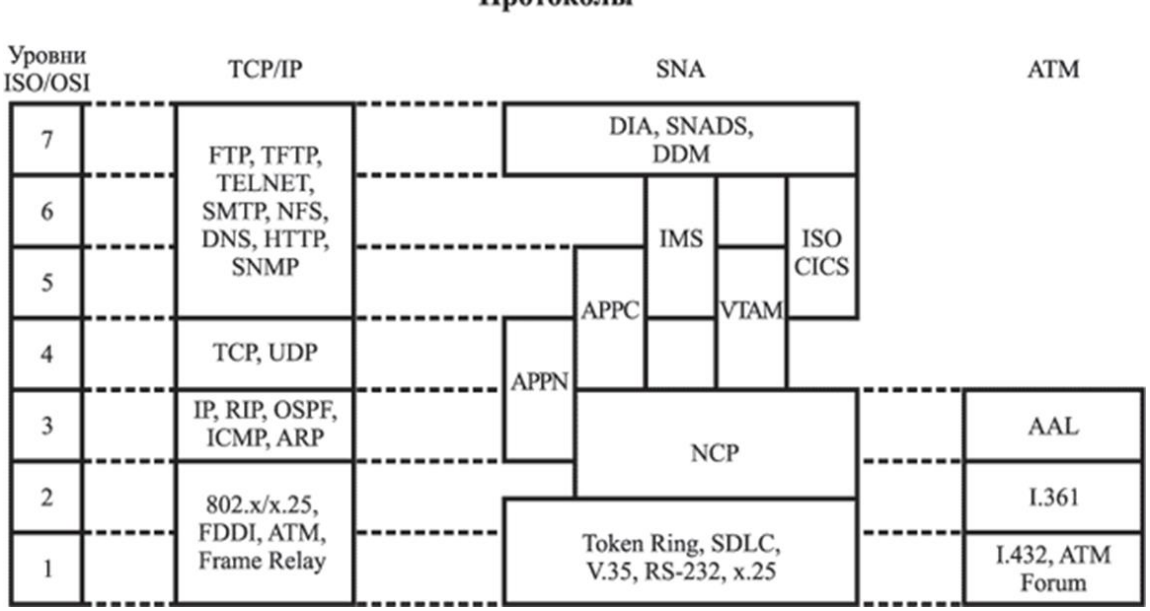


Рисунок 5.12. Стек SNA. Протоколы.

Уровень службы представления данных (Presentation Services) выполняет часть функций шестого уровня модели ISO/OSI (трансляция данных) и частично седьмого по административному управлению совместного использования ресурсов и синхронизации операций.

Уровень управления потоком данных (Data Flow Control) по своим функциям в основном соответствует сеансовому уровню модели ISO/OSI. Он управляет диалогами, обработкой запросов и ответов, групповых сообщений и прерыванием потока данных по запросу.

Уровень управления передачей (Transmission Control) выполняет функции транспортного уровня ISO/OSI по управлению передачей данных в пределах установленных сессий и некоторые функции (шифрование/дешифрование и др.) шестого уровня.

Уровень управления маршрутом (Path Control) определяет функции, в основном входящие в сетевой уровень модели ISO/OSI, а также включает в себя управление потоками данных (в модели ISO/OSI это функция канального уровня).

Уровень звена данных (Data Link) почти аналогичен второму уровню эталонной модели и совместим с ним по используемому протоколу, так как протоколы 802.2 и *SDLC* входят в семейство оригинального протокола *HDLC*.

На средних уровнях располагаются протоколы:

APPC (Advanced Program-to-Program Communication) - выполняет функции сеансового и транспортного уровней ISO/OSI;

на сеансовом уровне обеспечивает администрирование сеанса и трансляцию синтаксиса файлов,

на транспортном - организацию последовательностей сегментов и сквозное управление потоком данных.

CICS (Customer Information *Control System*) - инструментальное средство для построения приложений обработки транзакций, организует:

- ✓ доступ к распределенной файловой системе;
- ✓ защиту информации;
- ✓ многозадачность и пр.

IMS (Information Management System) - еще одна среда обработки транзакций, подобная *CICS*, позволяющая нескольким приложениям совместно использовать базы данных и планировать приоритеты транзакций.

TSO (*Time Sharing* Operation) – обеспечивает:

интерактивный пользовательский терминальный интерфейс, реализуя одновременную поддержку множества независимых параллельных пользовательских сеансов;

каждый пользователь TSO при помощи специальных команд получает возможность выполнять операции над наборами данных, запускать задания и контролировать ход их выполнения, использовать устройства, связываться с

другими пользователями и т.п.

Коммуникационные протоколы:

APPN (Advanced Peer-to-Peer Networking) - работает на сетевом и транспортном уровнях и обеспечивает:

одноранговое сетевое взаимодействие между несколькими физическими устройствами (миникомпьютерами, кластерными контроллерами, шлюзами, рабочими станциями и пр.);

предусматривает управление окном передач и службу каталогов.

VTAM (Virtual *Telecommunication Access Method*) –

обеспечивает: управление, обмен данными и управление потоками данных в сетях *SNA*;

на сеансовом уровне *VTAM* управляет диалогом и реализует управление сеансом;

на транспортном уровне обеспечивает сквозное управление потоками данных.

NCP (*Network Control* Program) - протокол управления ресурсами, подключенными к коммуникационным контроллерам;

частично выполняет функции сетевого уровня (маршрутизация, шлюзование);

частично - канального уровня (управление доступом к каналу, физическая и логическая адресация, управление потоком данных).

Первый уровень - физический (Physical) подобно модели ISO/OSI определяет характеристики сопряжения со средой передачи данных.

Решения этого уровня основаны преимущественно на тех же стандартах и рекомендациях, что и модель ISO/OSI.

Стек OSI

Следует различать стек протоколов OSI и модель OSI.

Стек OSI – это набор вполне конкретных спецификаций протоколов, образующих согласованный стек протоколов. Этот стек протоколов поддерживает правительство США в своей программе GOSIP.

Стек OSI в отличие от других стандартных стеков полностью соответствует модели взаимодействия OSI и включает спецификации для всех семи уровней модели взаимодействия открытых систем.

Рисунок 5.13. Стек OSI

На физическом и канальном уровнях стек OSI поддерживает спецификации Ethernet, Token Ring, FDDI, а также протоколы LLC, X.25 и ISDN.

На сетевом уровне реализованы протоколы, как без установления соединений, так и с установлением соединений.

Транспортный протокол стека OSI скрывает различия между сетевыми сервисами с установлением соединения и без установления соединения, так что пользователи получают нужное качество обслуживания независимо от нижележащего сетевого уровня. Чтобы обеспечить это, транспортный уровень требует, чтобы пользователь задал нужное качество обслуживания.

Определены 5 классов транспортного сервиса, от низшего класса 0 до высшего класса 4, которые отличаются степенью устойчивости к ошибкам и требованиями к восстановлению данных после ошибок.

Сервисы прикладного уровня включают передачу файлов, эмуляцию терминала, службу каталогов и почту. Из них наиболее перспективными являются: служба каталогов (стандарт X.500), электронная почта (X.400), протокол виртуального терминала (VT), протокол передачи, доступа и управления файлами (FTAM), протокол пересылки и управления работами (JTM).

В последнее время ISO сконцентрировала свои усилия именно на сервисах верхнего уровня.

Достоинства:

– стек OSI - международный, независимый от производителей стандарт;

– стек OSI полностью соответствует модели OSI;

– протоколы сетевого, транспортного и сеансового уровней стека OSI специфицированы и реализованы различными производителями, но распространены мало;

– довольно удачны и популярны прикладные протоколы: протокол

передачи файлов FTAM, протокол эмуляции терминала VTP, протоколы справочной службы X.500, электронной почты X.400 и ряд других;

Недостатки:

– протоколы стека OSI отличает большая сложность и неоднозначность спецификаций, т.к. разработчики стремились создать универсальное средство на все случаи жизни;

– из-за своей сложности протоколы OSI требуют больших затрат вычислительной мощности.

Стек DECnet

Digital Equipment Corporation (Digital) разработала семейство протоколов DECnet с целью обеспечения своих компьютеров рациональным способом сообщения друг с другом.

Первая версия DECnet обеспечивала возможность сообщения двух напрямую подключенных миникомпьютеров PDP-11 (1975г.).

В настоящее время выпущена пятая версия основного изделия DECnet (которую иногда называют Phase V, а в литературе компании Digital - DECnet/OSI).

DECnet Phase V представляет собой надлежащим образом расширенный набор комплекта протоколов OSI, поддерживающий все протоколы OSI, а также несколько других патентованных и стандартных протоколов, которые поддерживались предыдущими версиями DECnet.

DECnet не является архитектурой сети, а представляет собой ряд изделий, соответствующих Архитектуре Цифровой сети (*Digital Network Architecture* - DNA) компании Digital.

Рис. 5.14. иллюстрирует неполную картину DNA и связь некоторых ее компонентов с эталонной моделью OSI.

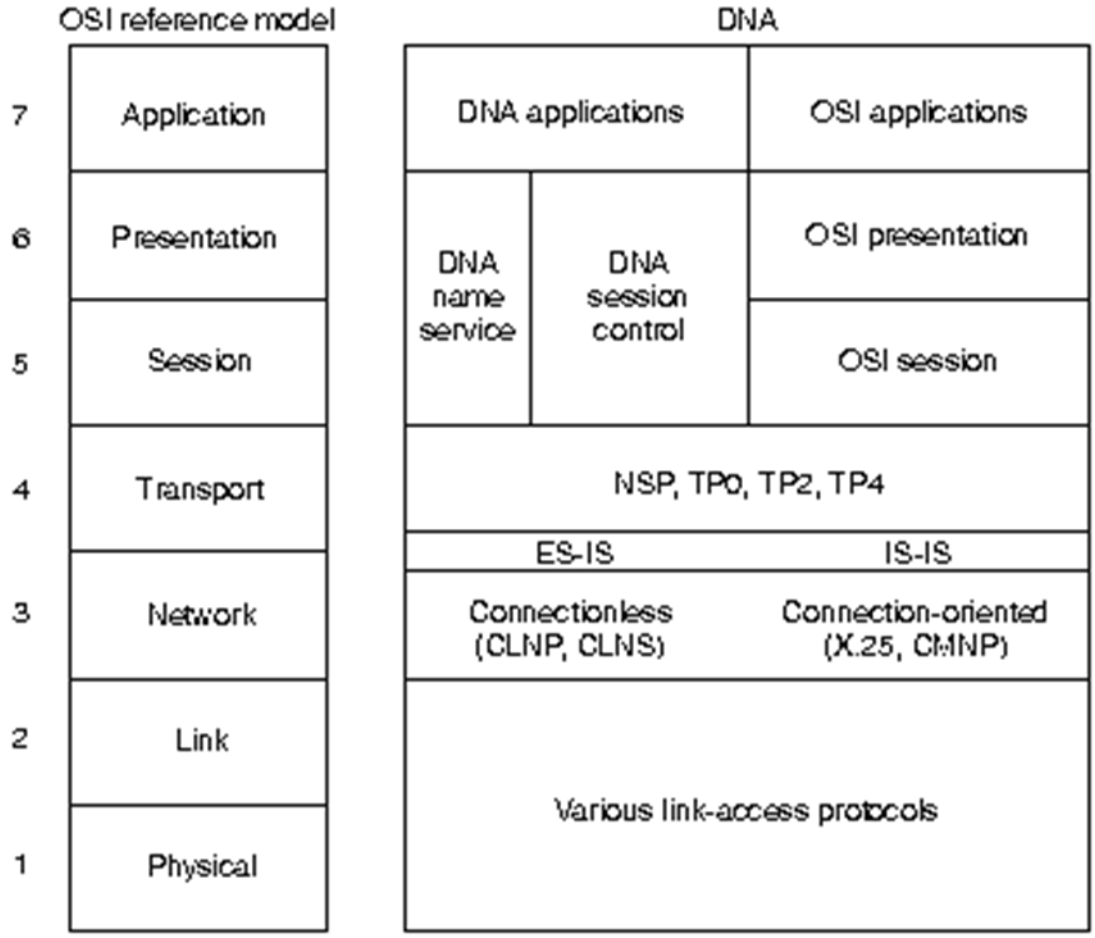


Рисунок 5.14. Стек DECnet

DNA поддерживает различные реализации физического и канального уровней. Среди них такие известные стандарты, как: Ethernet, Token Ring, Fiber Distributed Data Interface (FDDI), IEEE 802..2 и X.25.

DNA также предлагает протокол канального уровня для традиционного двухточечного соединения, который называется *Digital Data Communications Message Protocol* (DDCMP) (Протокол сообщений цифровой связи) и шину с пропускной способностью 70 Mb/sek, используемую для группы абонентов VAX, которая называется *Computer-room Interconnect bus* (CI bus) (шина межсоединений машинного зала).

Стек DECnet. Сетевой уровень.

DECnet поддерживает сетевые уровни как без установления соединения, так и с установлением соединения.

Оба сетевых уровня реализуются протоколами OSI.

Реализации без установления соединения используют:

Connectionless Network Protocol (CLNP) (Протокол сети без установления соединения)

Connectionless Network Service (CLNS) (Услуги сети без установления соединения).

Сетевой уровень с установлением соединения использует:

X.25 Packet-Level Protocol (PLP) (Протокол пакетного уровня), который также известен как X.25 level 3 (Уровень 3 X.25),

Connection-Mode Network Protocol (CMNP) (Протокол сети с установлением соединения).

Маршрутизация DNA Phase V включает в себя маршрутизацию OSI (ES-IS и IS-IS) и постоянную поддержку протокола маршрутизации DECnet Phase IV.

Формат блока данных маршрутизации DECnet Phase IV

Протокол маршрутизации DECnet Phase IV имеет несколько отличий от IS-IS. Одно из них-это разница в заголовках протоколов.

Заголовок слоя маршрутизации DNA Phase IV приведен на рис. 5.15.

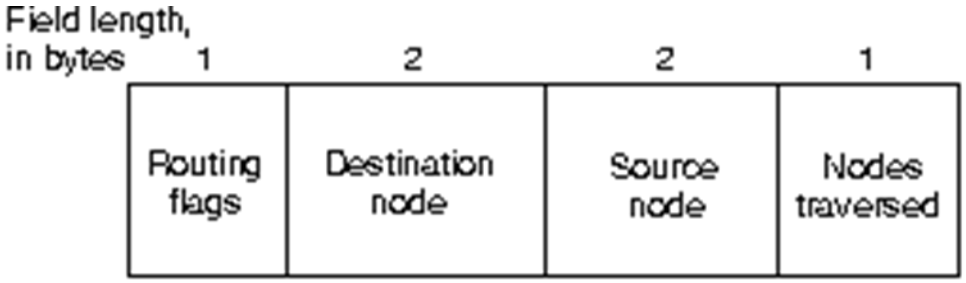


Рисунок 5.15. Заголовок слоя маршрутизации DNA Phase IV

Первое поле в заголовке маршрутизации DNA Phase IV-это поле флагов маршрутизации (routing flags), которое состоит из:

return-to-sender - бит возврата получателю, если он задан, то указывает, что данный пакет возвращается в источник.

return-to-sender request - бит запроса о возврате получателю, если он задан, то указывает на то, что запрашиваемые пакеты должны быть возвращены в источник, если они не могут быть доставлены в пункт назначения.

intraLAN - устанавливается по умолчанию. Если роутер обнаружит, что две сообщающиеся конечные системы не принадлежат одной и той же подсети, он исключает этот бит.

другие биты, которые обозначают формат заголовка, указывают, применялась ли набивка, и выполняют другие функции.

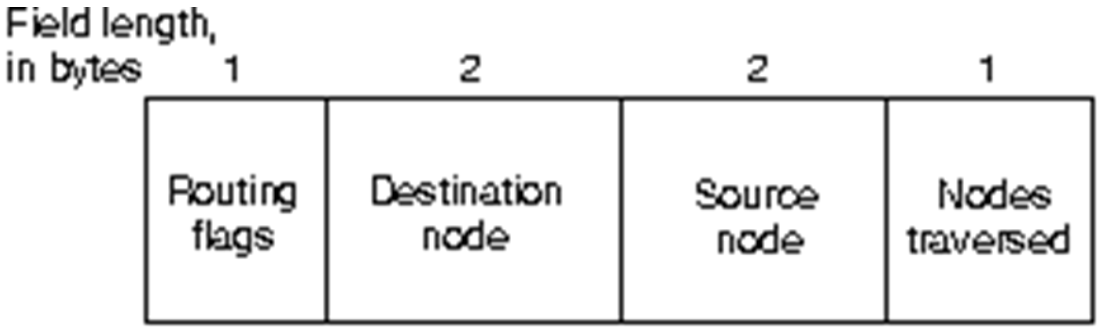


Рисунок 5.16. Формат блока данных маршрутизации DECnet Phase IV

Транспортный уровень DNA

Транспортный уровень DNA реализуется различными протоколами транспортного уровня, как патентованными, так и стандартными. Поддерживаются следующие протоколы транспортного уровня OSI: TP0, TP2 и TP4.

Принадлежащий Digital Протокол услуг сети (Network services protocol - NSP):

Обеспечивает ориентированное на соединение, с контролируемым потоком обслуживание, с фрагментацией и повторной сборкой сообщений.

Обеспечиваются два подканала – один для нормальных данных, второй для срочных данных и информации управления потоком.

Обеспечивается два типа управления потоком – простой механизм старт/стоп, при котором получатель сообщает отправителю, когда следует завершать и возобновлять передачу данных, и более сложная техника управления потоком, при которой получатель сообщает отправителю, сколько сообщений он может принять.

Может также реагировать на уведомления о перегрузке, поступающие из сетевого уровня, путем уменьшения числа невыполненных сообщений, которое он может допустить.

Стек IPX/SPX

Разработан фирмой Novell для сетевой операционной системы NetWare. Не требует большой вычислительной мощности. Ориентирован на работу в локальных сетях небольших размеров. В результате протоколы стека IPX/SPX до недавнего времени хорошо работали в локальных сетях и не очень — в больших корпоративных сетях, так как они слишком перегружали глобальные связи широковещательными пакетами, которые интенсивно используются несколькими протоколами этого стека (например, для установления связи между клиентами и серверами). С момента выпуска версии NetWare 4.0 специалисты Novell внесли в протоколы серьезные изменения, направленные на их адаптацию для работы в корпоративных сетях. Сейчас стек IPX/SPX реализован не только в NetWare, но и в нескольких других популярных сетевых ОС, например SCO UNIX, Sun Solaris, Microsoft Windows.

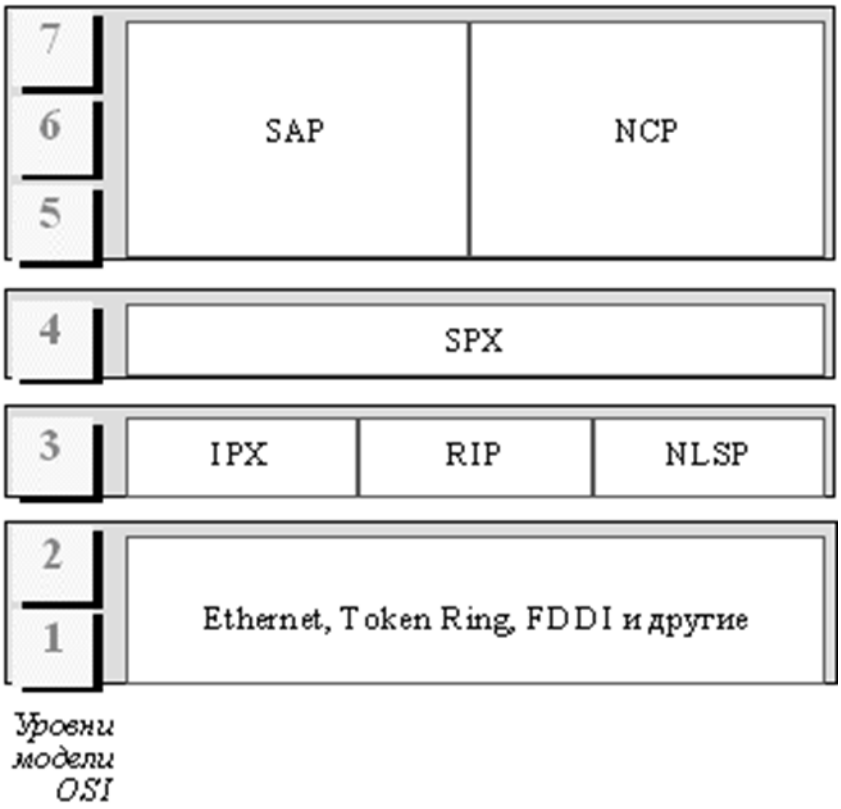


Рисунок 5.17. Стек IPX/SPX

Протокол NCP (NetWare Core Protocol):

– реализует архитектуру клиент-сервер.

– с помощью функций этого протокола рабочая станция производит:

– подключение к серверу,

– отображает каталоги сервера на локальные буквы дисководов,

– просматривает файловую систему сервера,

– копирует удаленные файлы,

– изменяет их атрибуты и т.п.,

– осуществляет разделение сетевого принтера между рабочими станциями.

SAP (Service Advertising Protocol):

позволяет маршрутизаторам обмениваться маршрутной информацией

дает возможность сетевым устройствам (серверам и маршрутизаторам) обмениваться информацией об имеющихся сетевых сервисах и сервисных услугах

позволяет сетевым устройствам постоянно корректировать данные о том, какие сервисные услуги имеются сейчас в сети.

используется серверами при старте для оповещения оставшейся части сети о своих услугах.

используется серверами при завершении работы, чтобы известить сеть о

прекращении действия своих услуг.

Транспортному уровню модели OSI в стеке Novell соответствует протокол SPX (Sequenced Packet Exchange), который осуществляет передачу сообщений с установлением соединений.

На сетевом уровне в стеке Novell работает протокол IPX (Internetwork Packet Exchange), который предназначен для передачи дейтограмм в системах, неориентированных на соединение и обеспечивает связь между NetWare серверами и конечными станциями

IPX обеспечивает выполнение трех функций: задание адреса, установление маршрута и рассылку дейтаграмм, за счет чего экономно потребляет вычислительные ресурсы

Также используются протоколы обмена маршрутной информацией RIP (Routing Internet Protocol) и NLSP (NetWare Link State Protocol) (аналог протокола OSPF стека TCP/IP).

На физическом и канальном уровнях в сетях Novell используются все популярные протоколы этих уровней (Ethernet, Token Ring, FDDI и другие).

Стек NetBIOS/SMB

Широко используется в продуктах компаний IBM и Microsoft. Протокол NetBIOS (Network Basic Input/Output System) появился в 1984 году как сетевое расширение стандартных функций базовой системы ввода/вывода (BIOS) IBM PC для сетевой программы PC Network фирмы IBM. После этот протокол был заменен так называемым протоколом расширенного пользовательского интерфейса NetBEUI — NetBIOS Extended User Interface. Для обеспечения совместимости приложений в качестве интерфейса к протоколу NetBEUI был сохранен интерфейс NetBIOS. Протокол SMB (Server Message Block) выполняет функции сеансового, представительного и прикладного уровней.

На основе SMB реализуется файловая служба, а также службы печати и передачи сообщений между приложениями.

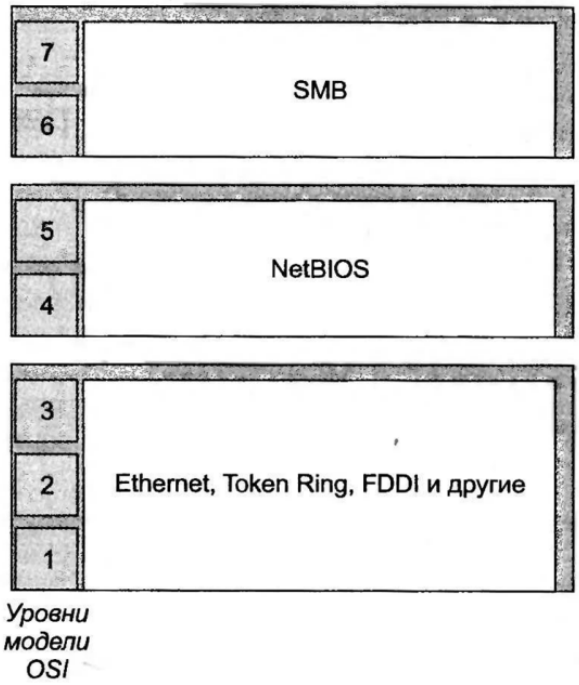


Рисунок 5.18. Стек NetBIOS/SMB

NetBEUI (NetBIOS Extended User Interface) позволяет организовать очень эффективный обмен информацией в сетях, состоящих не более чем из 200 компьютеров. Каждый из компьютеров должен обладать логическим именем. Логические имена присваиваются компьютерам динамически при их подключении к сети. Таблица имен распространяется на каждый компьютер сети. Поддерживается также работа с групповыми именами, что позволяет передавать данные сразу нескольким адресатам.

Протокол SMB (Server Message Block) несколько раз был усовершенствован (вышло три его версии), что позволило применять его даже в таких современных операционных системах, как Microsoft Vista и Windows 7.:

С его помощью организуется работа сети на трех самых высоких уровнях сеансовом, уровне представления и прикладном уровне. При его использовании становится возможным доступ к файлам, принтерам и другим ресурсам сети.

Протокол SMB универсален и может работать в паре практически с любым транспортным протоколом, например TCP/IP и SPX.

Достоинства NetBEUI:

– высокая скорость работы и очень малые требования к ресурсам

– наиболее эффективен для быстрого обмена данными в небольшой сети, состоящей из одного сегмента

– для доставки сообщений установленное соединение не является обязательным требованием: в случае отсутствия соединения протокол использует датаграммный метод, когда сообщение снабжается адресом получателя и отправителя и "пускается в путь", переходя от одного компьютера

к другому

Недостаток NetBEUI: полностью лишен понятия о маршрутизации пакетов, поэтому его использование в сложных составных сетях не имеет смысла.

5.11. Объединения сетей и глобальные сети.

Телекоммуникационные сети представляют собой комплекс аппаратных и программных средств, обеспечивающих передачу информационных сообщений между абонентами с заданными параметрами качества. Сообщение отображается изменением какого-либо параметра информационного сигнала (электромагнитные сигналы в сетях).

При создании сетей телекоммуникаций невозможно соединить всех абонентов между собой отдельными (выделенными) линиями связи. Это нецелесообразно экономически и невыполнимо практически, поэтому соединение многочисленных абонентов (А), находящихся на большом расстоянии, обычно производится через транзитные (телекоммуникационные) узлы (ТУ) связи.

Коммутация и маршрутизация

В некоторых сетях все возможные маршруты уже созданы и необходимо только выбрать наиболее оптимальный. Процесс выбора оптимального маршрута получил название маршрутизация, а устройство, ее реализующее, – маршрутизатор. Выбор оптимального маршрута узлы производят на основе таблиц маршрутизации (или коммутации) с использованием определенного критерия – метрики.

Различают сети с коммутацией каналов, когда телекоммуникационные узлы выполняют функции коммутаторов, и с коммутацией пакетов (сообщений), когда телекоммуникационные узлы выполняют функции маршрутизаторов. В сетях с коммутацией каналов канал создается до передачи сообщения. Эти два вида сетей используются для передачи двух различных видов трафика.

Сети с коммутацией каналов обычно передают равномерный (потоковый) трафик – например, телефонные сети. В сетях передачи данных с пульсирующим трафиком применяется коммутация пакетов (сообщений), например, в компьютерных сетях.

Различие коммутации пакетов (сообщений) состоит в том, что сообщение может быть очень большим. Поэтому если в нем обнаруживается ошибка, то повторно нужно передавать все сообщения большого объема. В сетях с коммутацией пакетов большое сообщение предварительно разбивается на сравнительно небольшие пакеты (сегменты). Поэтому при потере или искажении части сообщения повторно передается только потерянный пакет (сегмент).

Принципы межсетевого взаимодействия.

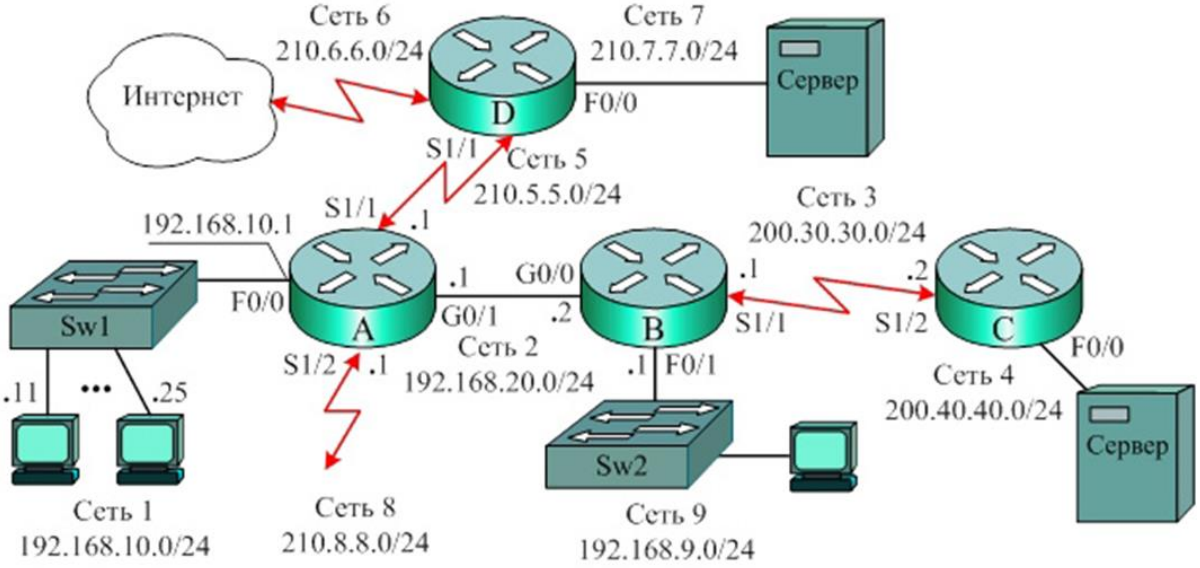
Приведем основные устройства и методы межсетевого взаимодействия,

принципы маршрутизации (статической и динамической), функционирование таблиц маршрутизации в сетях IPv4 и IPv6.

Рассмотрим процесс передачи данных по сети.

Соединение локальных сетей LAN различных технологий (FastEthernet, GigabitEthernet, Token Ring и др.) в глобальную (распределенную) WAN-сеть происходит с помощью устройств (маршрутизаторов) и протоколов сетевого уровня (3) семиуровневой эталонной модели OSI или уровня межсетевого взаимодействия модели TCP/IP.

Поэтому маршрутизаторы имеют как LAN, так и WAN интерфейсы



5.19. Пример распределенной сети

LAN-интерфейсы (G0/0, G0/1, F0/0, F0/1) используются для связи с узлами (компьютерами, серверами), напрямую или через коммутаторы;

WAN-интерфейсы (S1/1, S1/2) необходимы, чтобы связываться с другими маршрутизаторами и всемирной сетью Интернет.

Интерфейсы могут подключаться к разным видам передающей среды, в которых могут использоваться различные технологии канального и физического уровней.

Когда адресат назначения находится в другой сети, то конечный узел пересылает пакет на шлюз по умолчанию, роль которого выполняет интерфейс маршрутизатора, через который все пакеты из локальной сети пересылаются в удаленные сети.

Например, для сети 192.168.10.0/24 (рис.5.19) шлюзом по умолчанию является интерфейс F0/0 маршрутизатора A с адресом 192.168.10.1, а интерфейс F0/1 маршрутизатора B выполняет роль шлюза по умолчанию для сети 192.168.9.0/24.

Через шлюз по умолчанию пакеты из удаленных сетей поступают в локальную сеть назначения.

Функции маршрутизатора

При пересылке пакетов адресату назначения маршрутизатор реализует две основные функции:

выбирает наилучший (оптимальный) путь к адресату назначения, анализируя логический адрес назначения передаваемого пакета данных.

Маршрут — это последовательность маршрутизаторов, которые должен пройти пакет от отправителя до пункта назначения.

производит коммутацию принятого пакета с входного интерфейса на выходной для пересылки адресату.

Процесс выбора наилучшего пути получил название маршрутизация.

Маршрутизаторы принимают решения, базируясь на сетевых логических адресах (IP-адресах), находящихся в заголовке пакета.

Для определения наилучшего пути передачи данных через связываемые сети, маршрутизаторы *строят таблицы маршрутизации* и *обмениваются сетевой маршрутной информацией* с другими сетевыми устройствами.

Протокол IP. Транспортный уровень.

Протокол IP (Internet Protocol — Интернет-протокол) обеспечивает маршрутизацию пакетов с негарантированной доставкой (best-effort delivery) без установки логического соединения (connectionless).

Каждый компьютер, включенный в Internet (а точнее каждый сетевой интерфейс) получает уникальный в рамках всего Internet адрес (адресами ведают национальные комитеты Internet).

IP-адрес - это 4-байтовая последовательность, каждый байт записывается в виде десятичного числа.

Например, 195.19.19.19 - адрес одной из машин

Адресация и виды информации в Internet

IP-адрес состоит из двух частей:

адреса сети

номер хоста. Под хостом следует понимать не только компьютер в сети, но и вообще любое устройство, которое имеет свой сетевой интерфейс.

Существует несколько классов IP-адресов.

Адреса класса A предназначены для использования в больших сетях общего пользования.

Адреса класса B предназначены для использования в сетях среднего размера.

Адреса класса C предназначены для использования в сетях с небольшим числом компьютеров.

Таблица 5.2. Классы IP-сетей

Класс	Диапазон значений первого байта	Возможное число сетей	Возможное число узлов
A	1-126	126	16777214
B	128- 191	16382	65534
C	192-223	2097150	254

Классы IP-адресов
Маска подсети
Пример - маска подсети
Адрес в сети:
123.45.6.0
01111011.00101101.00000110.00000000
255.255.255.255
11111111.11111111.11111111.11111111
/16 – префиксная запись
255.255.0.0
/24 - префиксная запись
255.255.255.0

Сетевая маска	Инверсия	Префикс	Используется	Размер
0.0.0.0	255.255.255.255	/0	4,294,967,294	весь интернет
128.0.0.0	127.255.255.255	/1	2,147,483,646	128 классов 'a'
192.0.0.0	63.255.255.255	/2	1,073,741,822	64 класса 'a'
224.0.0.0	31.255.255.255	/3	536,870,910	32 класса 'a'
240.0.0.0	15.255.255.255	/4	268,435,454	16 классов 'a'
248.0.0.0	7.255.255.255	/5	134,217,726	8 классов 'a'
252.0.0.0	3.255.255.255	/6	67,108,862	4 класса 'a'
254.0.0.0	1.255.255.255	/7	33,554,430	2 класса 'a'
255.0.0.0	0.255.255.255	/8	16,777,214	1 класс 'a'
255.128.0.0	0.127.255.255	/9	8,388,606	128 классов 'b'
255.192.0.0	0.63.255.255	/10	4,194,302	64 класса 'b'
255.224.0.0	0.31.255.255	/11	2,097,150	32 класса 'b'
255.240.0.0	0.15.255.255	/12	1,048,574	16 классов 'b'
255.248.0.0	0.7.255.255	/13	524,286	8 классов 'b'
255.252.0.0	0.3.255.255	/14	262,142	4 класса 'b'
255.254.0.0	0.1.255.255	/15	131,070	2 класса 'b'
255.255.0.0	0.0.255.255	/16	65,534	1 класс 'b'
255.255.128.0	0.0.127.255	/17	32,766	128 классов 'c'
255.255.192.0	0.0.63.255	/18	16,382	64 класса 'c'
255.255.224.0	0.0.31.255	/19	8,190	32 класса 'c'
255.255.240.0	0.0.15.255	/20	4,094	16 классов 'c'
255.255.248.0	0.0.7.255	/21	2,046	8 классов 'c'
255.255.252.0	0.0.3.255	/22	1,022	4 класса 'c'
255.255.254.0	0.0.1.255	/23	510	2 классов 'c'
255.255.255.0	0.0.0.255	/24	254	1 класс 'c'
255.255.255.128	0.0.0.127	/25	126	128 хостов
255.255.255.192	0.0.0.63	/26	62	64 хоста
255.255.255.224	0.0.0.31	/27	30	32 хоста
255.255.255.240	0.0.0.15	/28	14	16 хостов
255.255.255.248	0.0.0.7	/29	6	8 хостов
255.255.255.252	0.0.0.3	/30	2	4 хоста
255.255.255.254	0.0.0.1	/31	0	2 хоста
255.255.255.255	0.0.0.0	/32	1	1 хост

Рисунок 5.20. Таблицы соответствия

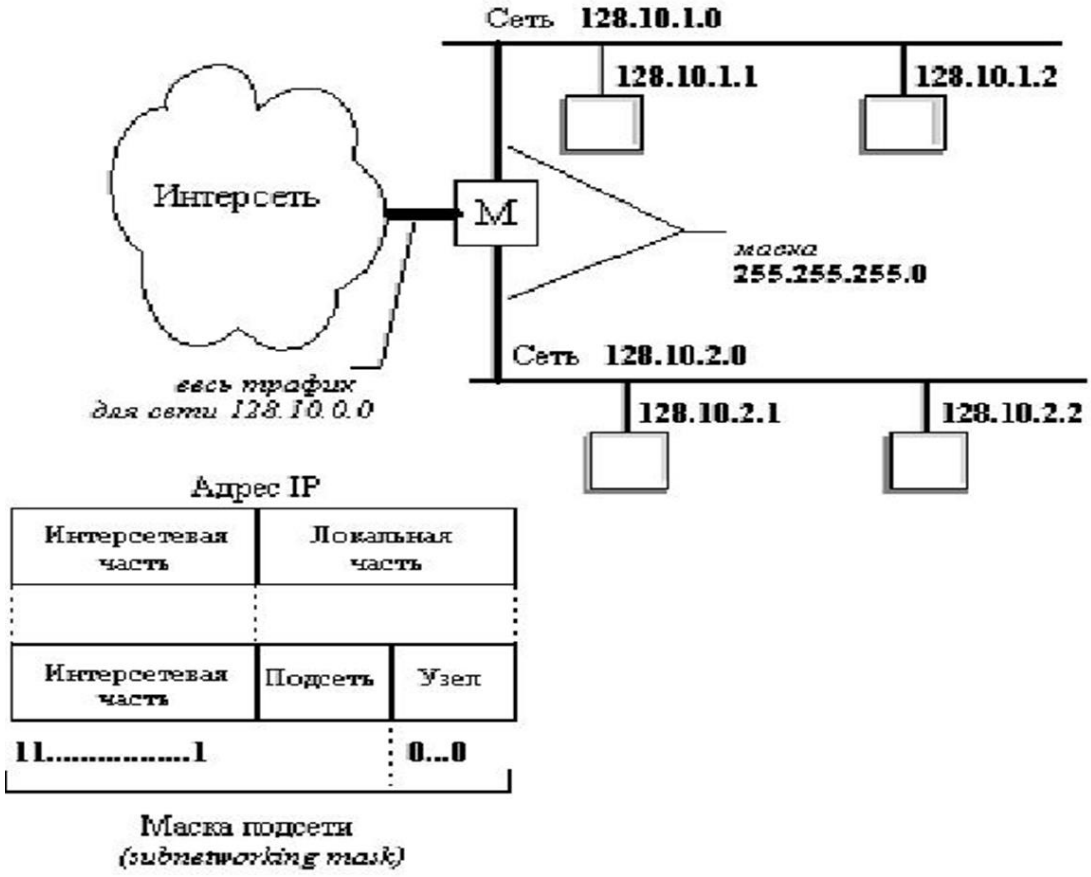


Рисунок 5.21. Деление сетей на подсети

Поле *версия* принимает значение «4»

Длина IP-заголовка - сам заголовок, включая необязательные опции и символы заполнения

Поле *тип сервиса* определяет способ обслуживания пакета в конкретных сетях и, главным образом, связано с возможностью задержки *(delay)* пакета в сети.

Поле *общая длина* определяет длину IP-пакета без заголовка.

Поле *идентификация* предназначено для помощи при «сборке» сообщения.

Поле *флаги* определяет место датаграммы в сообщении.

Рисунок 5.22. Структура IP-пакета

Поле *смещение фрагмента* определяет смещение датаграммы относительно начала сообщения.

Поле время жизни предназначено для определения срока, после которого пакет должен быть удален из сети.

Поле *протокол* определяет тип датаграммы.

Поле *контрольная сумма* служит для идентификации повреждений пакета при передаче.

Поле *адрес получателя* - IP-адрес места назначения.

Поле *адрес отправителя* - IP-адрес отправителя.

Опции могут иметь переменную длину и обычно применяются для трассировки пакетов, обеспечения безопасности.

Заполнитель применяется для выравнивания заголовка на 32-битовую границу.

IPv6

В IPv6 длина адреса расширена до 128 бит (в IPv4 32 бита), что и позволяет увеличить адресное пространство до 2128.

Последняя версия стандарта RFC1924 (A Compact Representation of IPv6 Addresses R. Elz Apr-01-1996).

Адреса в компактной форме записываются в виде 8 шестнадцатеричных чисел (x:x:x:x:x:x:x:x).

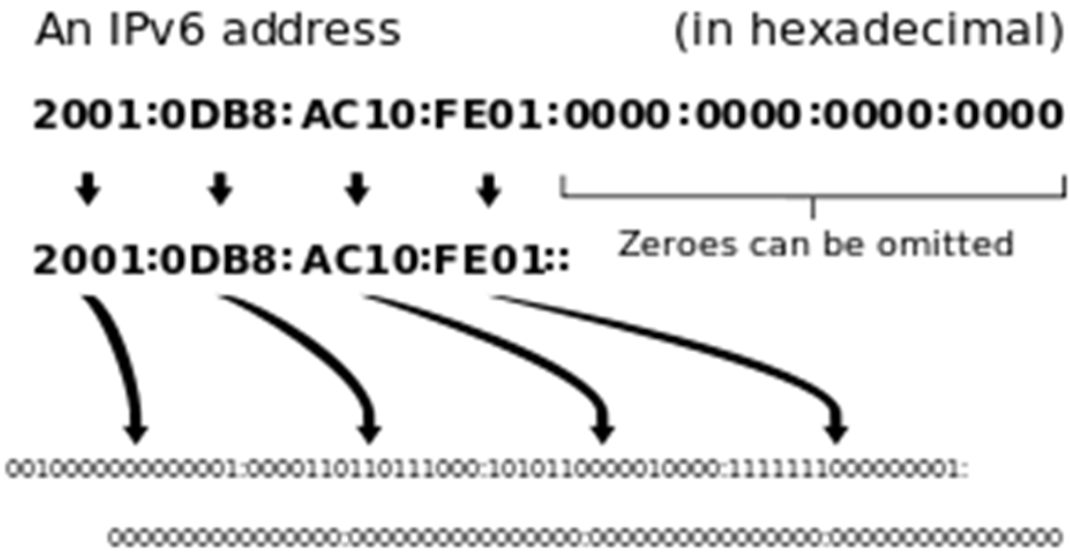


Рисунок 5.23. IPv6

Все, что не является обязательным для маршрутизации пакета из точки в А в точку Б, стало опциональным – переехало в extension header, который лежит между IPv6-заголовком и TCP/UDP-данными. В этом самом extension-заголовке уже и проживают фрагментирование, IPsec, source routing и множество другого функционала.

Упростили задачу маршрутизаторам, ведь уже не надо пересчитывать контрольные суммы.

Контрольные суммы убрали вовсе - целостность доставки будут обеспечивать вышележащие протоколы (TCP).

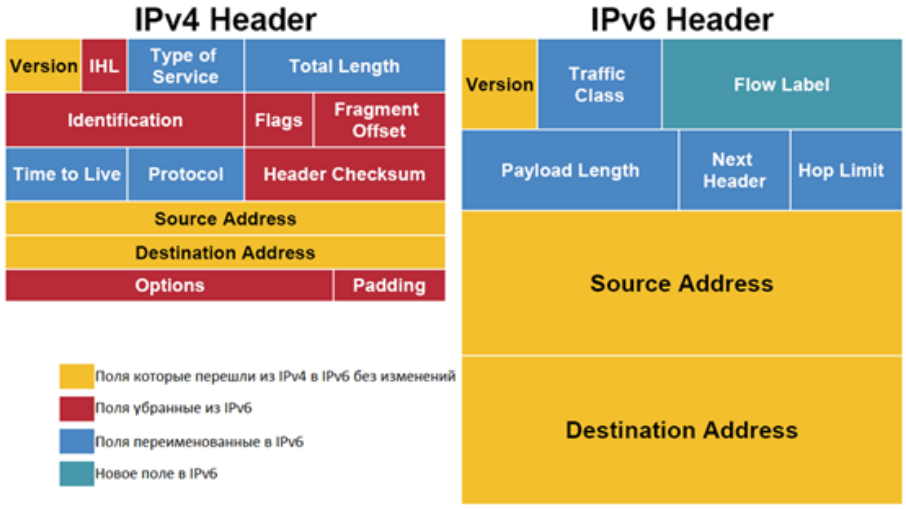


Рисунок 5.23. Сравнение структуры IPv4 и IPv6

Маршрутизация

В стандартной модели взаимодействия открытых систем в функции сетевого уровня входит решение следующих задач:

- ✓ передача пакетов между конечными узлами в составных сетях;
- ✓ выбор маршрута передачи пакетов, наилучшего по некоторому критерию;
- ✓ согласование разных протоколов канального уровня, использующихся в отдельных подсетях одной составной сети.

Функции маршрутизатора

физический и канальный уровень:

- получение доступа к среде;
- формирование битовых сигналов;
- прием кадра;
- подсчет его контрольной суммы и передача поля данных кадра верхнему уровню, в случае если контрольная сумма имеет корректное значение.

сетевой уровень:

- извлекает из пакета заголовок сетевого уровня и анализирует содержимое его полей;
- поверяется контрольная сумма;
- выполняется проверка, не превысило ли время, которое провел пакет в сети (время жизни пакета), допустимой величины. На этом этапе вносятся корректировки в содержимое некоторых полей, например, наращивается время жизни пакета, пересчитывается контрольная сумма;
- определение маршрута пакета.

Таким образом осуществляется фильтрация трафика.

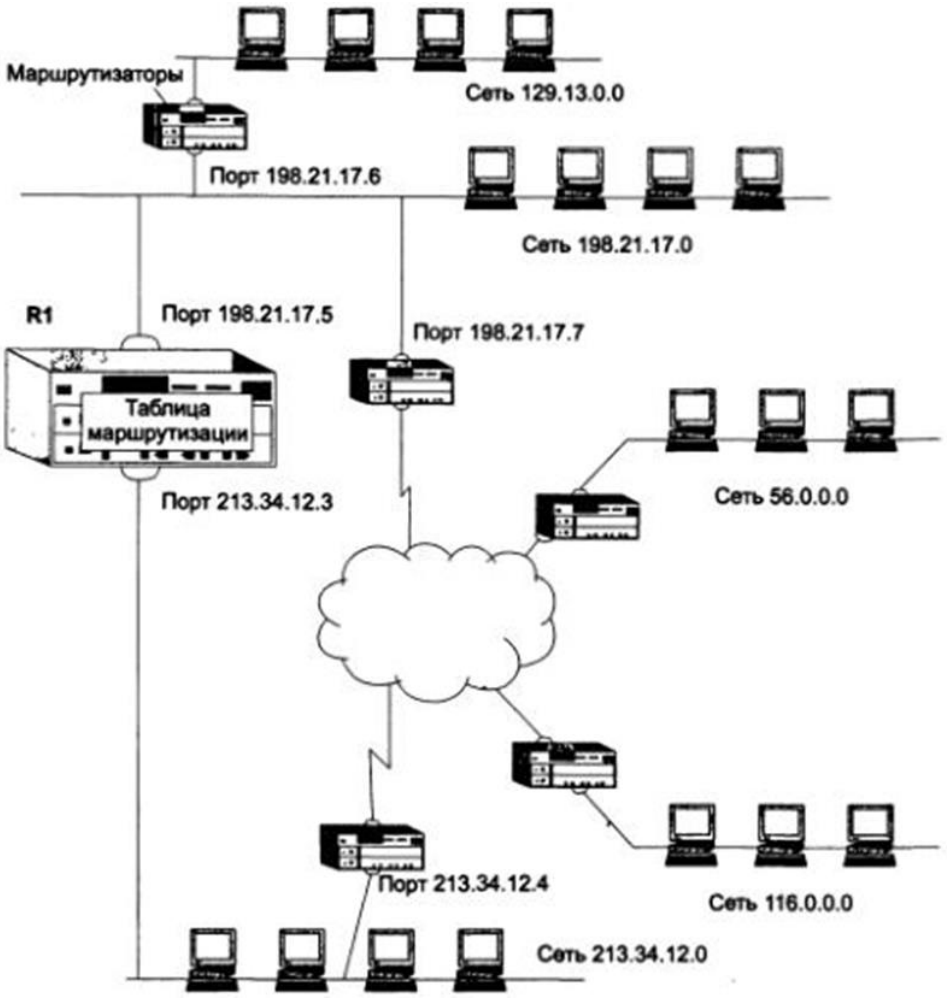


Рисунок 5.24. Схема маршрутизации

Таблица 5.3. Упрощенная таблица маршрутизации маршрутизатора R1

Адрес сети назначения	Адрес следующего маршрутизатора	Адрес выходного интерфейса	Расстояние до сети назначения
56.0.0.0	213.34.12.4	213.34.12.3	15
116.0.0.0	213.34.12.4	213.34.12.3	13
129.13.0.0	198.21.17.6	198.21.17.5	2
198.21.17.0	198.21.17.5	198.21.17.5	1(подсоединена)
213.34.12.0	213.34.12.3	213.34.12.3	1(подсоединена)
Маршрут по умолчанию	198.21.17.7	198.21.17.5	-

Управление фрагментацией

В большинстве типов локальных и глобальных сетей определяется такое понятие как максимальный размер поля данных кадра или пакета, в которые должен инкапсулировать свой пакет протокол IP. Эту величину обычно называют максимальной единицей транспортировки - MaximumTransferUnit, MTU. Сети Ethernet: MTU=1500 байт, Сети FDDI: MTU=4096 байт,

Сети X.25: MTU=128 байт.

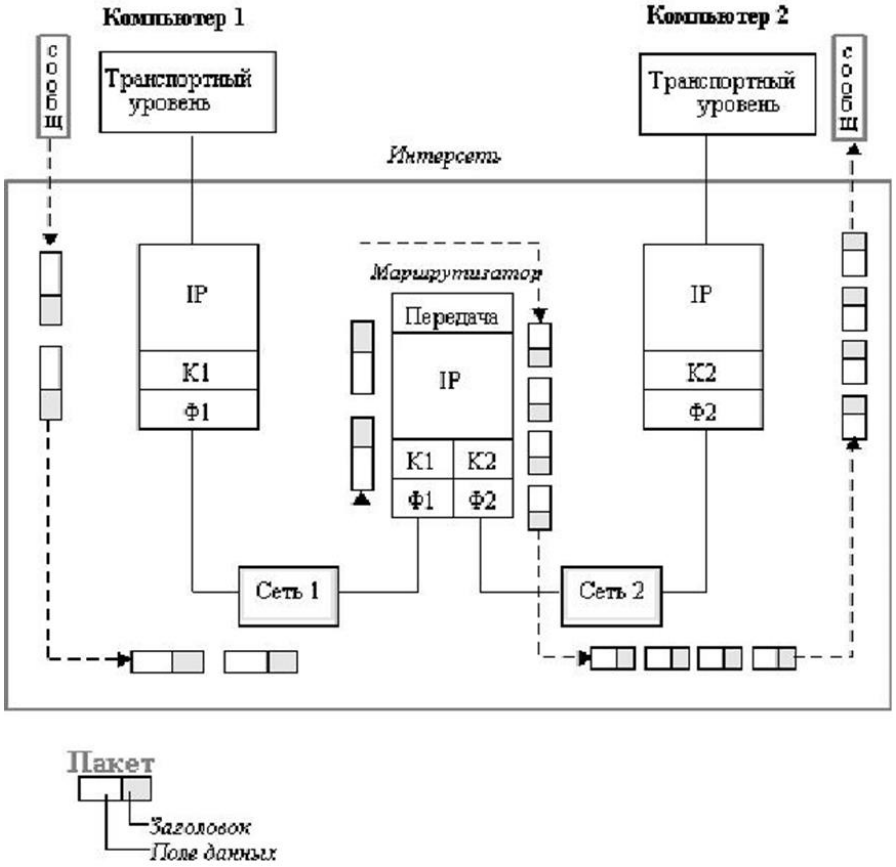


Рисунок 5.25. Управление фрагментацией

Маршрутизация с помощью IP-адресов.

В стеке TCP/IP существуют несколько подходов к оптимизации маршрута продвижения пакета:

- Одношаговый подход;
- Маршрутизация от источника.

Алгоритмы построения таблиц для одношаговой маршрутизации.

- Фиксированная маршрутизация:
- Простая маршрутизация;
- Случайная маршрутизация;
- Лавинная маршрутизация;
- Маршрутизация по предыдущему опыту.

Адаптивные протоколы обмена маршрутной информацией:

дистанционно-векторные алгоритмы (Distance Vector Algorithms, DVA) (протокол RIP);

алгоритмы состояния связей (Link State Algorithms, LSA) (протоколы IS-Is, OSPF, NLSP).

Маршрутизация от источника.

Существует и прямо противоположный, многошаговый подход – маршрутизация от источника (Source Routing). В соответствии с ним узел-источник задает в отправляемом в сеть пакете полный маршрут его следования через все промежуточные маршрутизаторы. При использовании многошаговой маршрутизации нет необходимости строить и анализировать таблицы маршрутизации. Это ускоряет прохождение пакета по сети, разгружает маршрутизаторы, но при этом большая нагрузка ложится на конечные узлы.

В вычислительных сетях применяется сегодня гораздо реже, чем схема распределенной одношаговой маршрутизации.

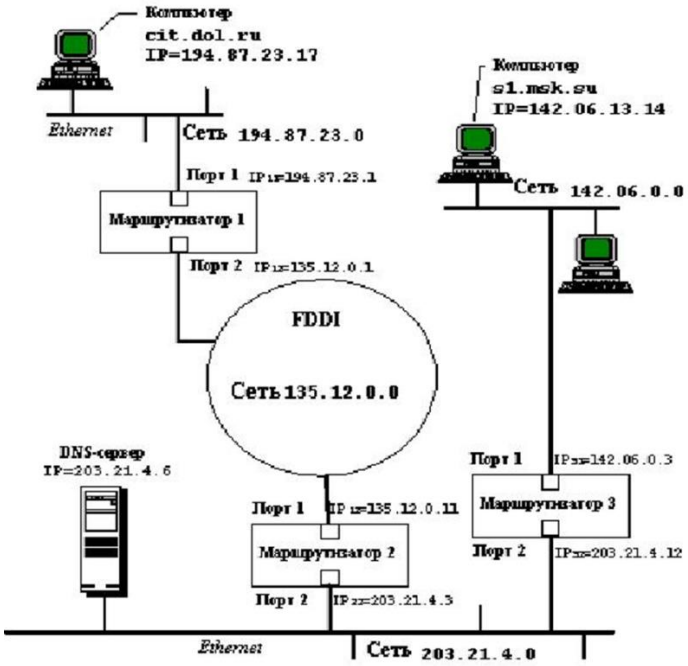


Рисунок 5.26. Пример взаимодействия узлов с использованием протокола IP.

Статическая маршрутизация.

Конфигурация таблицы маршрутизации.

Статическая запись включает в себя следующее:

– Адрес сети - идентификатор сети или имя сети получателя;

– Сетевую маску - маску подсети для адреса сети;

– Адрес шлюза - IP адрес или имя узла, являющегося интерфейсом к сети назначения.

Таблица 5.4. Статическая маршрутизация

Добавление или изменение статической записи	Функция
route add \сеть\ mask [сетевая маска] \шлюз\	Добавляет маршрут.
route -p add [сеть] mask [сетевая маска] [шлюз]	Добавляет постоянный маршрут
route delete \сеть\ [шлюз]	Удаляет маршрут
route change [сеть] [шлюз]	Изменяет маршрут
route print	Показывает таблицу маршрутизации
route -f	Стирает все маршруты

Динамическая IP-маршрутизация.

Самыми распространенными являются алгоритмы адаптивной (или динамической) маршрутизации. Эти алгоритмы обеспечивают автоматическое обновление таблиц маршрутизации после изменения конфигурации сети.

Протоколы, построенные на основе адаптивных алгоритмов, позволяют всем маршрутизаторам собирать информацию о топологии связей в сети, оперативно отрабатывая все изменения конфигурации связей. В таблицах маршрутизации при адаптивной маршрутизации обычно имеется информация об интервале времени, в течение которого данный маршрут будет оставаться действительным. (Time To Live, TTL).

Адаптивные алгоритмы обычно имеют распределенный характер, который выражается в том, что в сети отсутствуют какие-либо выделенные маршрутизаторы которые собирали бы и обобщали топологическую информацию: эта работа проделана между всеми маршрутизаторами.

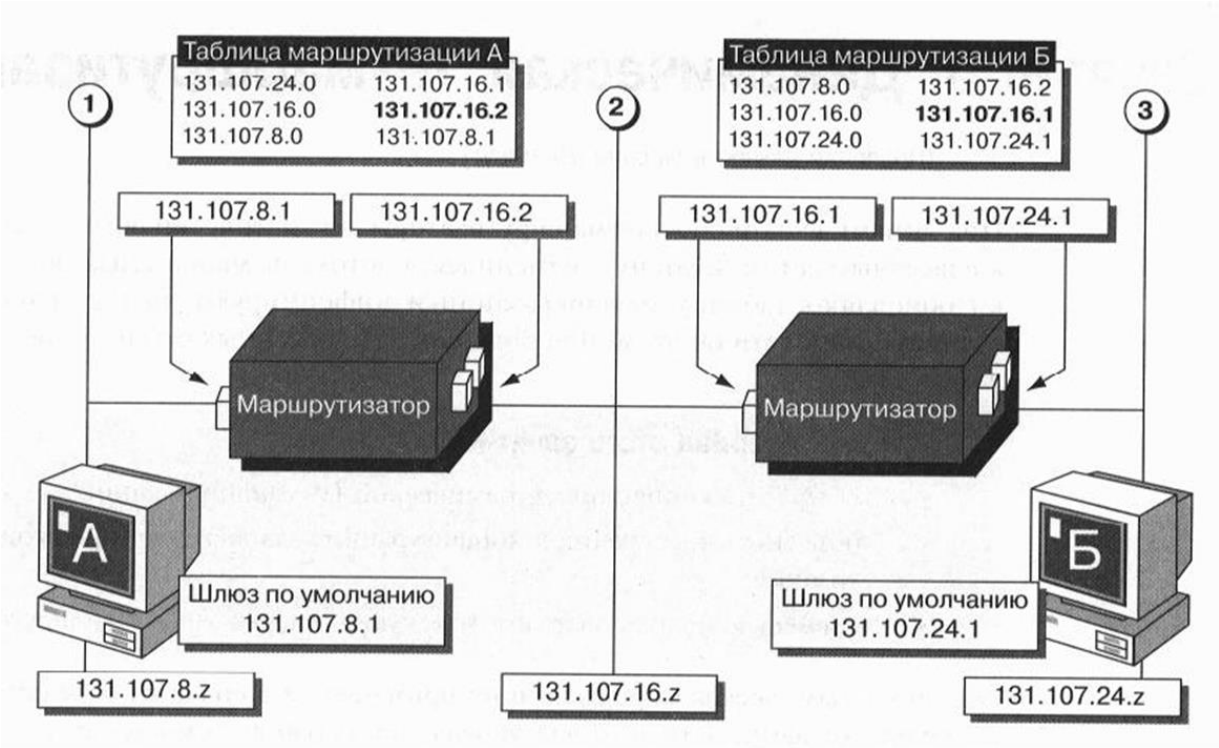


Рисунок 5.27. Динамическая IP-маршрутизация

В последнее время тенденция – использовать серверы маршрутов. Сервер маршрутов собирает маршрутную информацию, а затем раздает ее по запросам маршрутизатора которые освобождаются в этом случае от функции создания таблиц маршрутизации, либо создают части этих таблиц.

Созданы специальные протоколы взаимодействия маршрутизаторов с сервером маршрутов, например Next Hop Resolution Protocol (NHRP).

Адаптивные алгоритмы маршрутизации должны отвечать нескольким важным требованиям:

✓ Должны обеспечивать, если не оптимальность, то хотя бы рациональность маршрута.

✓ Должны быть достаточно простыми, (не тратить слишком много сетевых ресурсов, не требовать слишком большого объема вычислений, не порождать интенсивный служебный трафик).

✓ Должны обладать свойством сходимости, то есть всегда приводить к однозначному результату за приемлемое время.

Адаптация RIP - маршрутизаторов к изменениям состояния сети

Протокол адаптивной маршрутизации RIP:

✓ Истечение времени жизни маршрута.

✓ Время распространения сведений об отказавших маршрутизаторах кратно времени жизни записи, а коэффициент кратности равен количеству хопов между самыми дальними маршрутизаторами сети.

✓ Используют указание специального расстояния (бесконечности (16

хопов)) до сети, ставшей недоступной.

RIP - позволяет маршрутизаторам обмениваться идентификаторами сетей, которых может достичь маршрутизатор, и расстоянием до этих сетей.

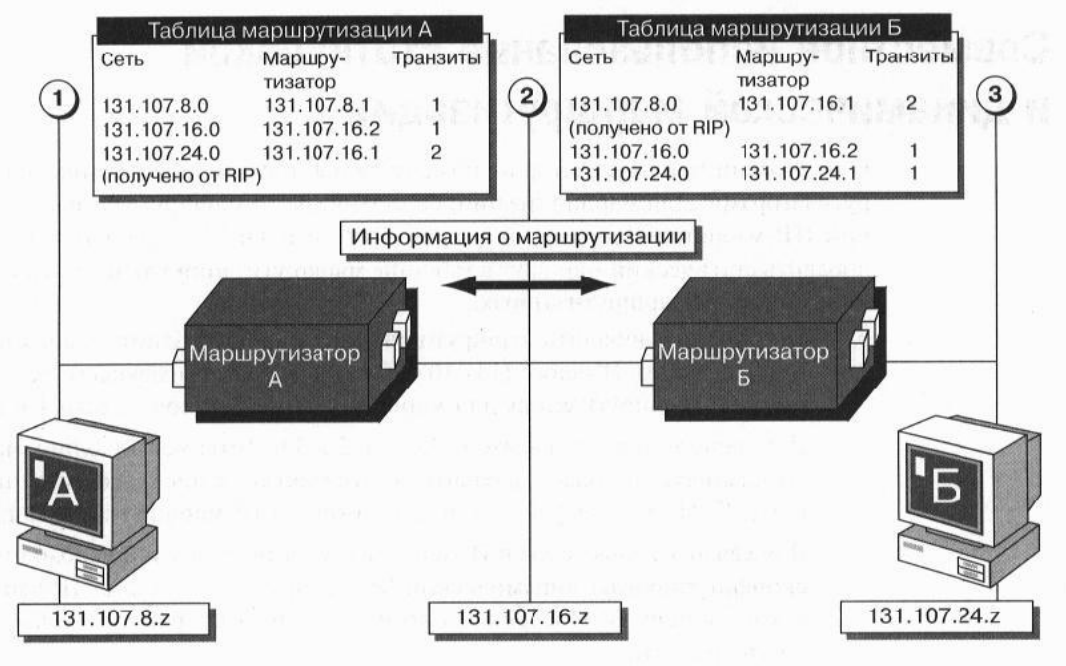


Рисунок 5.28. Адаптация RIP - маршрутизаторов

Недостатки:

Разрабатывался для локальных сетей. Из-за этого RIP хорошо работает только в малых объединенных IP-сетях с небольшим числом маршрутизаторов;

Максимальный размер одного RIP-пакета - 512 байт, для отправки больших таблиц маршрутизации содержащей сотни или даже тысячи записей требуется множество RIP-пакетов;

В таблице маршрутизации каждой записи о маршруте, полученном по RIP, назначен 3-минутный тайм-аут (отсчитывается с момента получения), по истечении которого не обновленные записи удаляются;

Если маршрутизатор выходит из строя, распространение изменений по объединенной сети может занять несколько минут. Это называется проблемой медленной конвергенции.

Сервисы транспортного уровня

Приложения не формируют IP-пакеты. Для этого разработчикам приложений пришлось бы разбираться с неспецифичными для них сетевыми задачами и приложения стали бы зависимы от типа сети. Транспортный уровень принимает потоки данных или сообщения, «упаковывает» данные приложений в IP-пакеты и передает в сеть. Сервис негарантированной доставки единичных сообщений обеспечивает транспортный протокол UDP. Потоковый транспортный сервис с надежной доставкой обеспечивает протокол TCP.

Немного о портах

Порты транспортных протоколов бывают предписанные (well-known) и динамически назначаемые. Номера предписанных портов лежат в диапазоне от 1 до 1023. Распределением (предписанием) well-known портов занимается специальная организационная структура Интернет – IANA. Well-known порты приписываются серверам известных (широко распространенных) приложений. Клиенты обычно используют эфемеридные, динамически назначаемые порты. Динамически назначаемые порты используются либо известными приложениями для установления временных соединений, либо нераспространенными приложениями. Когда нераспространенному приложению необходимо получить номер порта, либо когда возникает конфликт использования номеров портов (например, когда на сервере работают два однотипных известных приложения), приложения запрашивают динамический номер порта у TCP/IP-стека.

Идентификация приложений

Транспортный уровень принимает из сети пакеты для множества приложений, возникает проблема разобраться – где чьи данные. Сетевые приложения идентифицируются 16-разрядным числом – портом (port). Одно приложение может использовать несколько портов. Сетевое соединение (между приложениями) однозначно определяется набором параметров: (T-protocol, SRC-IP, SRC-port, DST-IP, DST-port).

Протокол ARP

ARP (англ. Address Resolution Protocol — протокол определения адреса) — протокол в компьютерных сетях, предназначенный для определения MAC-адреса по известному IP-адресу.

Наибольшее распространение ARP получил благодаря повсеместности сетей IP, построенных поверх Ethernet, поскольку практически в 100% случаев при таком сочетании используется ARP.

В семействе протоколов IPv6 ARP не существует, его функции возложены на ICMPv6

Рассмотрим суть функционирования ARP на простом примере.

1) Компьютер А (IP-адрес 10.0.0.1) и компьютер Б (IP-адрес 10.22.22.2) соединены сетью Ethernet.

2) Компьютер А желает переслать пакет данных на компьютер Б, IP-адрес компьютера Б ему известен.

3) Однако сеть Ethernet, которой они соединены, не работает с IP-адресами. Поэтому компьютеру А для осуществления передачи через Ethernet требуется узнать адрес компьютера Б в сети Ethernet (MAC-адрес в терминах Ethernet). Для этой задачи и используется протокол ARP.

4) По этому протоколу компьютер А отправляет широковещательный запрос, адресованный всем компьютерам в одном с ним широковещательном домене. Суть запроса: «компьютер с IP-адресом 10.22.22.2, сообщите свой

MAC-адрес компьютеру с MAC-адресом (напр. a0:ea:d1:11:f1:01)».

5) Сеть Ethernet доставляет этот запрос всем устройствам в том же сегменте Ethernet, в том числе и компьютеру Б.

6) Компьютер Б отвечает компьютеру А на запрос и сообщает свой MAC-адрес (напр. 00:ea:d1:11:f1:11)

7) Теперь, получив MAC-адрес компьютера Б, компьютер А может передавать ему любые данные через сеть Ethernet.

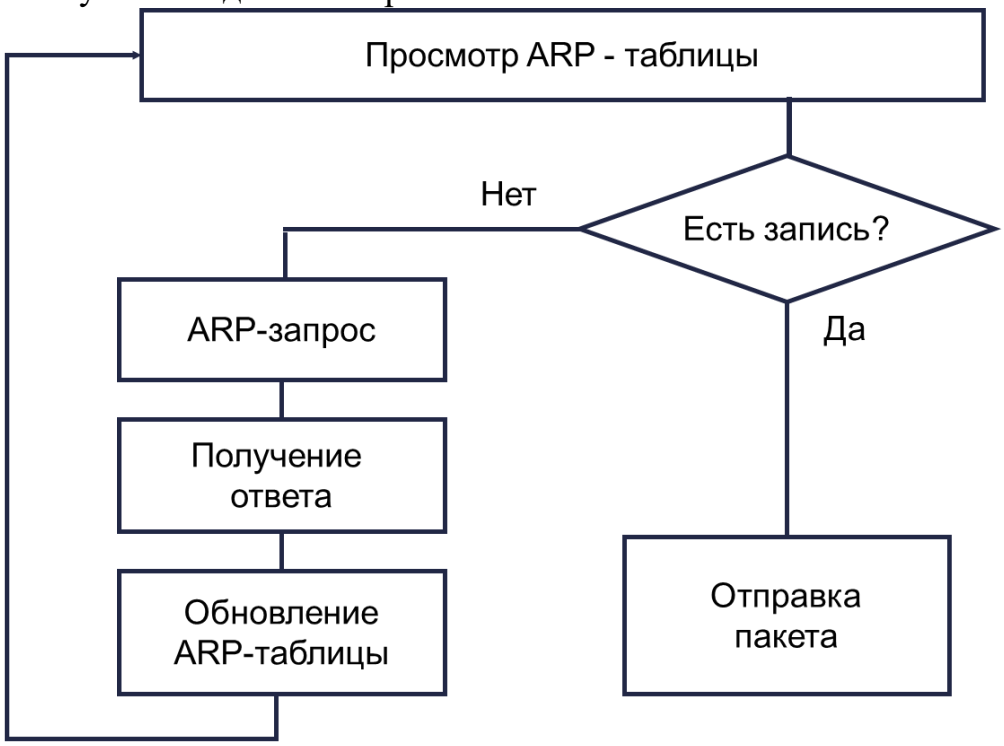


Рисунок 5.29. Схема работы ARP

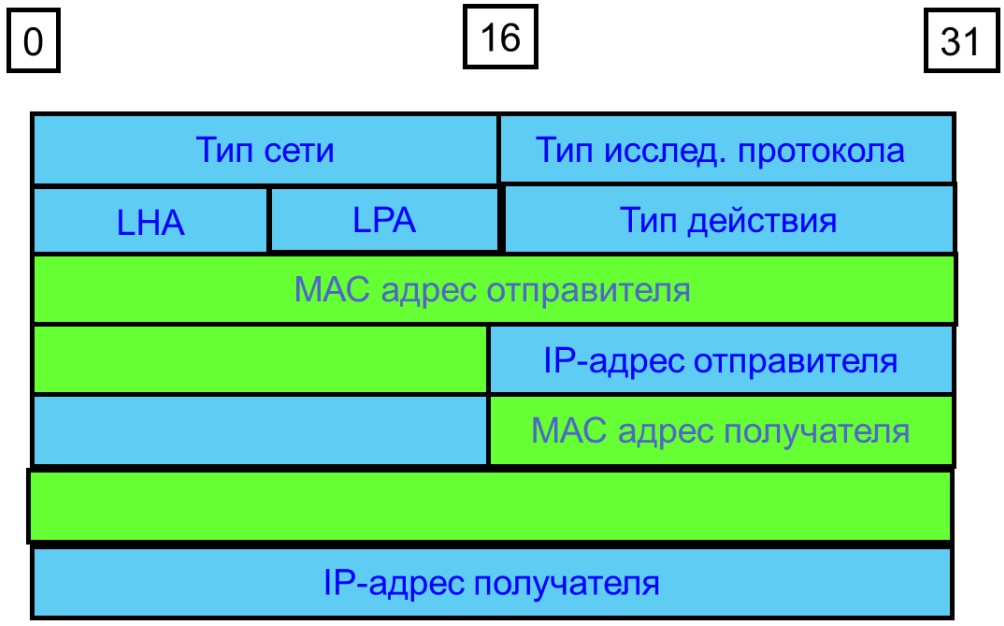


Рисунок 5.30. Формат ARP – пакета

ICMPv6

Протокол ICMPv6 выполняет ряд важных функций:

1) Автоконфигурации рабочих станций и серверов (RFC-4862)
2) Для определения адресных префиксов и другой конфигурационной информации
3) Для выявления адресов-дублеров
4) Для определения MAC-адресов (L2)
5) Для выявления ближайшего маршрутизатора, способного переадресовать пакеты.
6) Детектирование изменения адресов канального уровня
7) Отслеживание достижимых и недостижимых сетевых объектов

Выявление MAC-адреса по IP осуществляется следующим образом:

1) Пусть MAC-адрес отправителя равен AA-BB-CC-00-00-AA, а его IPv6-адрес - 2001:CB8::1234:5678:AAAA.
2) MAC-адрес будущего адресата - AA-BB-CC-00-00-BB,
3) а IPV6 - 2001:DB8::1234:5678:BBBB.
4) Отправитель посылает мультикаст-сообщение выявления соседа по адресу FF02::1:FF78:BBBB (кто имеет адрес 2001:DB8::1234:5678:BBBB?).
5) В результате посылается сообщение анонсирование соседа по адресу 2001:CB8::1234:5678:AAAA (MAC=AA-BB-CC-00-00-AA) (MAC-адрес отправителя = AA-BB-CC-00-00-BB).
6) Отправитель на основе префикса знает, что уникастный адрес получателя является локальным.
7) При формировании мультикастного адреса места назначения используются 24 младшие бит IPv6-адреса получателя.

Протокол UDP. Сеансовый уровень.

Сервис протокола UDP

User Datagram Protocol, UDP (RFC 768) обеспечивает обмен единичными сообщениями между приложениями

UDP очень прост, это прямая ретрансляция сервиса протокола IP приложениям

UDP - дейтаграммный протокол, не гарантирующий доставку (может как терять, так и дуплицировать сообщения) и не сохраняющий порядка следования сообщений

Сообщение протокола UDP называют *пользовательской дейтаграммой* (user datagram).

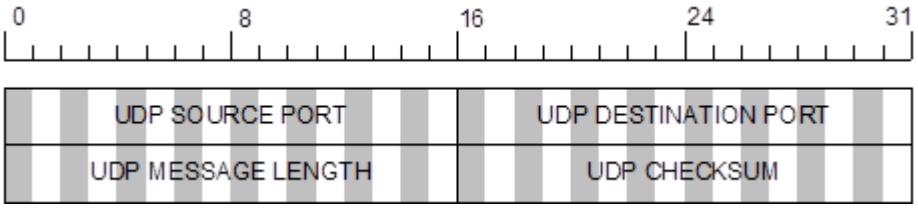


Рисунок 5.30. Дейтаграмма UDP

UDP SOURCE PORT и UDP DESTINATION PORT – порты процесса-отправителя и процесса-получателя. Source port имеет ненулевое заполнение, если процесс-отправитель должен получить ответное сообщение.

UDP MESSAGE LENGTH – полная длина заголовка и сегмента данных. UDP CHECKSUM – контрольная сумма.

Контрольная сумма Т-протоколов.

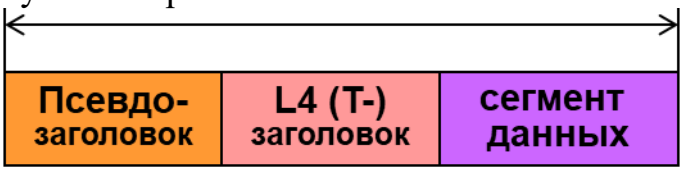


Рисунок 5.31. Пространство расчета контрольной суммы

Расчет контрольной суммы в TCP обязателен, а в UDP опционален: заполнение поля CHECKSUM нулями означает в UDP отказ от расчета контрольной суммы. При отказе от расчета контрольной суммы в UDP следует иметь ввиду, что сохранность блока данных не гарантирована ничем, кроме канального протокола.

Расчет контрольной суммы производится по трем структурам данных:

- ✓ Псевдозаголовку;
- ✓ Транспортному заголовку;
- ✓ Сегменту данных транспортного сообщения.

Протокол TCP. Сеансовый уровень.

Сервис протокола TCP (Transfer Control Protocol, TCP (RFC793)):

- ✓ обеспечивает транспорт потоков (stream) т.е. приложение, передающее данные, не заботится о том, чтобы передавать транспортному протоколу информацию порциями;
- ✓ обрабатывает неструктурированные потоки данных, т.е. не накладывает никаких ограничений на состав потока и взаимосвязи между его элементами;
- ✓ буферизует данные, передаваемые в сеть;
- ✓ организует т.н. виртуальные соединения посредством предварительной согласовательной процедуры;
- ✓ обеспечивает полнодуплексное соединение при этом обеспечивается:
- – управление потоком (в зависимости от пропускной способности и

загрузки сети);

– обеспечивает целостность потока и гарантирует доставку данных.

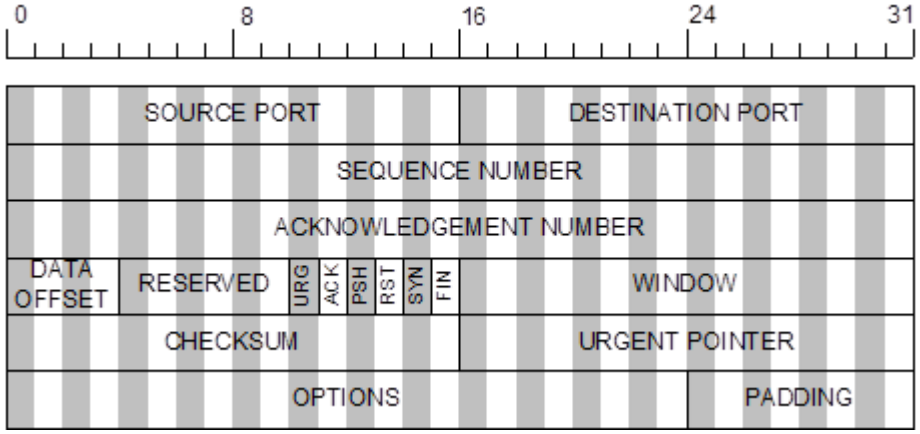


Рисунок 5.32. Сегмент TCP

TCP передает данные порциями (сегментами), каждый из которых включается затем в IP пакет.

Заголовок сегмента (транспортный заголовок TCP) обеспечивает возможность для передачи управляющей информации протокола вместе с трафиком (piggybacking).

Поля заголовка TCP

SOURCE PORT, DESTINATION PORT – номера портов отправителя и получателя сообщения

SEQUENCE, ACKNOLEDGEMENT NUMBER, WINDOW, URGENT POINTER – поля для управления потоком

DATA OFFSET – указатель на конец заголовка (начало блока данных)

CHECKSUM – контрольная сумма по сегменту данных

OPTIONS – варианты

PADDING - заполнение

Биты управления

URG, urgent – срочная передача данных

ACK, acknowledgement – подтверждение приема

PSH, push – очистка буфера

RST, reset – переустановление соединения

SYN, synchronize – синхронизация потоков

FIN, finish – окончание потока данных

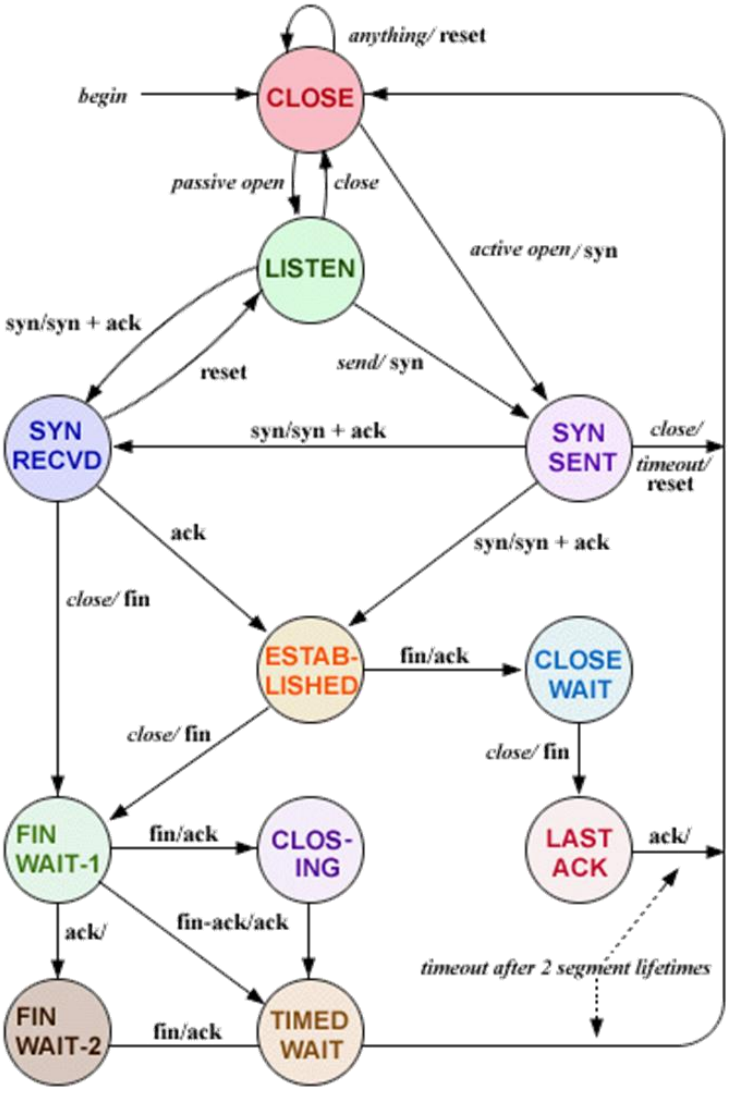


Рисунок 5.33. Конечный автомат протокола TCP

Работу протокола TCP удобно пояснить на основе конечного автомата; состояния:

CLOSE – холостое состояние, отсутствие соединения.

LISTEN, SYN RECVD, SYN SENT – промежуточные состояния фазы установления соединения.

ESTABLISHED – соединение установлено, передача данных.

CLOSE WAIT, LAST ACK, FIN WAIT 1,2, CLOSING, TIMED WAIT – промежуточные состояния фазы завершения соединения.

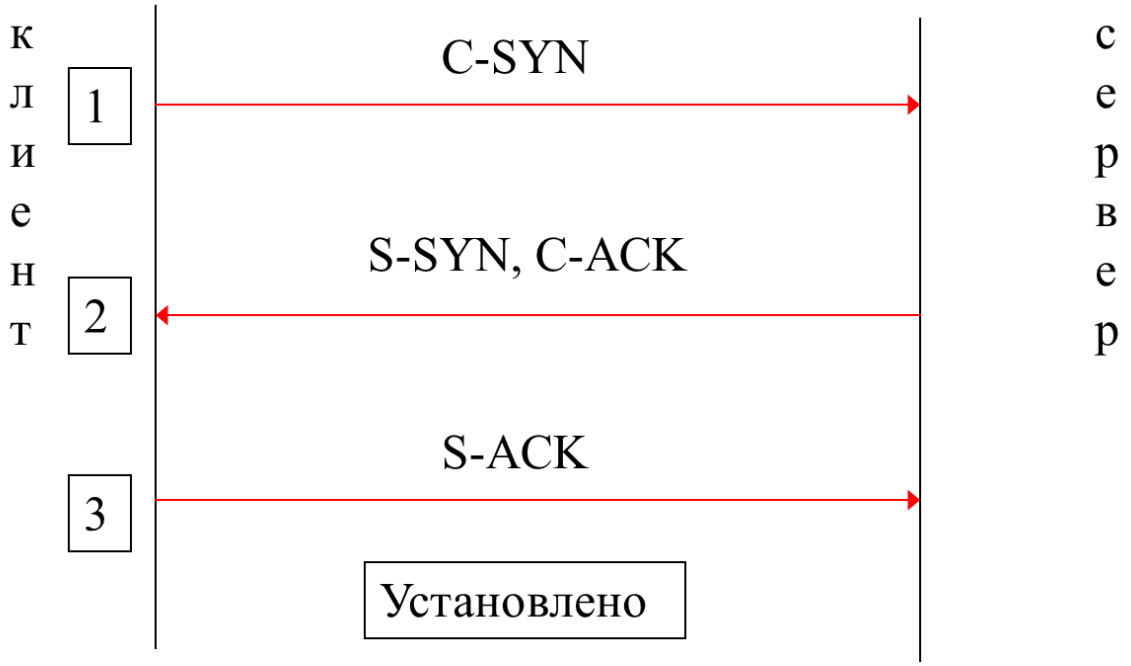


Рисунок 5.34. Установка TCP соединения (3-way handshake)

Передача данных. Квитирование.

Отправитель берет из выходного буфера очередную порцию данных, формирует TCP-сегмент, рассчитывает контрольную сумму, высылает сегмент, устанавливает тайм-аут на ожидание квитанции

Получатель получает сегмент, сверяет контрольную сумму, выдает квитанцию:

– если контрольная сумма сошлась – ACK SQNC=X+L+1 (ожидание порции потока со следующего байта);

– если сумма не сошлась – квитанция не высылается.

Получив квитанцию, отправитель перемещает счетчик переданного потока в позицию, соответствующую ACK SQNC.

Настройка тайм-аута (очень упрощенно).

Если сегмент (или квитанция) потеряны – отправитель по истечении тайм-аута повторяет передачу сегмента. Длительность тайм-аута должна быть настроена для пары отправитель-получатель: Если оба в одной локальной сети – тайм-аут м.б. несколько миллисекунд. Если на разных концах земли – требуется тайм-аут 1-10 с

TCP производит измерение времени до прихода квитанции (round trip time, RTT).

Результаты измерений RTT усредняются с убывающим для более ранних измерений весом. Длительность тайм-аута выбирается пропорциональной усредненному (с убывающим во времени весом) времени двойного прохода.

Оконное управление потоком

Если отправлять сегменты только после поступления квитанций (верхний

рисунок), пропускная способность линии сильно падает из-за больших времен ожидания квитанций. Эффективность можно существенно поднять, если позволить отправителю высылать N сегментов до поручения квитанции на 1й сегмент из серии N (нижний рис. 5.35.).

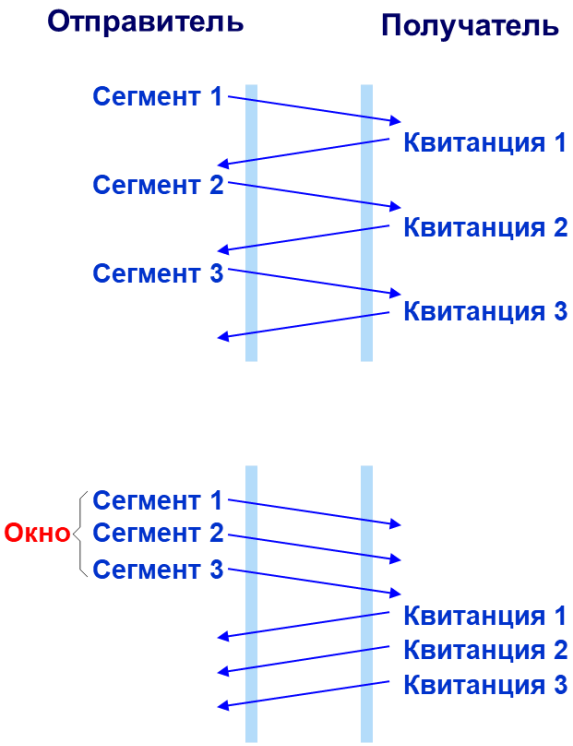


Рисунок 5.35. Оконное управление потоком

Число N называется [*скользящим*] *окном*, а этот механизм – *оконным управлением потоком.*

Оконное управление потоком.

Изменение размера окна позволяет эффективно управлять интенсивностью потока данных.

✓ при N=1 реализуется последовательная передача сегмент-квитанция;

✓ при больших N реализуется практически непрерывный дуплексный поток сегментов и квитанций.

Механизм оконного управления потоком используется в TCP/IP для управления загрузкой сети (при перегрузке производится уменьшение окон передающих трафик узлов).

Управление перегрузкой сети (упрощенно)

Перегрузка на промежуточном устройстве диагностируется по увеличению задержки передачи пакетов (дополнительно – по ICMP-сообщениям от промежуточных маршрутизаторов).

Методы управления перегрузкой:

на оконечных устройствах:

✓ мультипликативное уменьшение окна (всякий раз вдвое, вплоть до 1) и увеличение тайм-аута

✓ медленный старт: после восстановления работоспособности сети (устранение перегрузки) – увеличение окна вдвое (на 1 сегмент) по всякому факту подтверждения приема до размера окна получателя

на промежуточных устройствах:

усечение (сброс) хвоста очереди или, более поздний и оптимальный механизм – произвольный ранний сброс хвоста очереди.

Принудительная передача данных.

Отправитель накапливает данные во входном буфере. Иногда, например, после набора команды в терминальном режиме, требуется передать данные срочно, не ожидая наполнения буфера. Для этого в прикладном интерфейсе TCP используется команда push, «выталкивающая» данные из выходного буфера в сеть. Бит PSH устанавливается в значение 1, чтобы принимающий трафик узел немедленно произвел прием данных.

Передача вне [приемной] очереди. В случае необходимости передать данные срочно, вне очереди (out of band), например, для передачи запроса на перезагрузку удаленного компьютера нужно указать получателю на подлежащие срочному приему данные: бит URG=1; указатель срочных данных указывает на позицию срочных данных в сегменте.

Получатель примет данные, игнорируя необработанную входную очередь.

Завершение TCP-соединения. Сброс TCP-соединения.

Когда следует прекратить связь, а штатное завершение TCP-соединения по каким-либо причинам невозможно, используется аварийный механизм сброса соединения. Инициатор высылает сегмент с установленным битом RST. Получатель немедленно разрывает соединение.