ФИЗИЧЕСКАЯ ИНФРАСТРУКТУРА СЕТИ

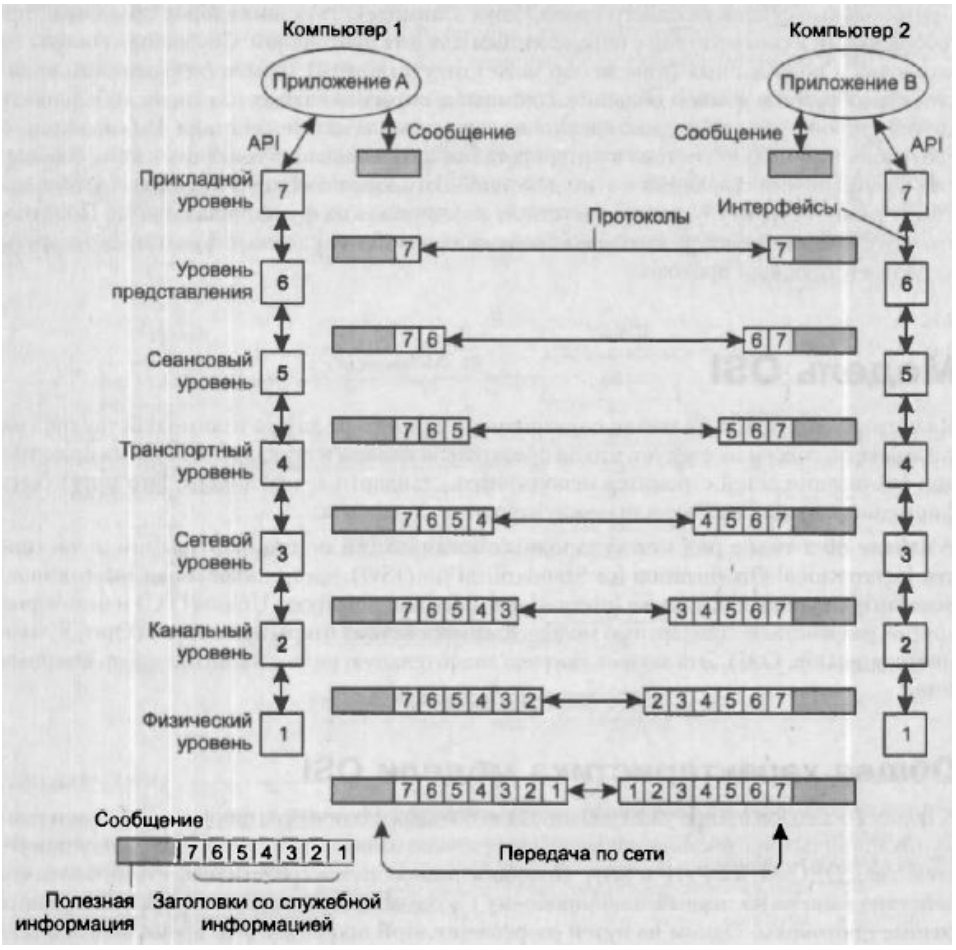
Под физической инфраструктурой сети подразумевают её топологию, то есть физическое строение сети со всем её оборудованием: кабелями, маршрутизаторами, коммутаторами, мостами, концентраторами, серверами и узлами. К физической инфраструктуре также относятся транспортные технологии: Ethernet 802.11b, коммутируемая телефонная сеть общего пользования (PSNT), ATM – в совокупности они определяют, как осуществляется связь на уровне физических подключений.

ЛОГИЧЕСКАЯ ИНФРАСТРУКТУРА СЕТИ

Логическая инфраструктура сети состоит из всего множества программных элементов, служащих для связи, управления и безопасности узлов сети, и обеспечивает связь между компьютерами с использованием коммуникационных каналов, определённых в физической топологии. Примеры элементов логической инфраструктуры сети: система доменных имён (Domain Name System - DNS), сетевые протоколы (например TCP/IP), сетевые клиенты (например Клиент для сетей NetWare – Client Service for NetWare), а также сетевые службы (например планировщик пакетов качества службы QoS – Quality of Service Packet Scheduler).

ЭТАЛОННАЯ МОДЕЛЬ OSI

Сетевая модель OSI – это сетевая модель стека сетевых протоколов OSI/ISO. Эта модель основана на разработке Международной организации по стандартизации и является первым шагом к международной стандартизации протоколов, используемых на различных уровнях. Называется эта структура эталонной моделью взаимодействия открытых систем ISO, поскольку она связывает открытые системы, то есть системы, открытые для связи с другими системами. Модель OSI имеет семь уровней: прикладной, представления, сеансовый, транспортный, сетевой, канальный и физический. Каждый уровень имеет дело с совершенно определённым аспектом взаимодействия сетевых устройств.



Итак, пусть приложение узла А хочет взаимодействовать с приложением узла В. Для этого приложение А обращается с запросом к прикладному уровню, например, к файловой службе. На основании этого запроса программное обеспечение прикладного уровня формирует сообщение стандартного формата. После формирования сообщения прикладной уровень направляет его вниз по стеку уровню представления. Протокол уровня представления на основании информации, полученной из заголовка сообщения прикладного уровня, выполняет требуемые действия и добавляет к сообщению собственную служебную информацию – заголовок уровня представления, в котором содержатся указания для протокола уровня представления машины-адресата. Полученное в результате сообщение передаётся вниз сеансовому уровню, который, в свою очередь, добавляет свой заголовок и т.д. Наконец, сообщение достигает нижнего, физического уровня, который собственно и передаёт его по линиям связи машине-адресату. К этому моменту сообщение «обрастает» заголовками всех уровней. Физический уровень помещает сообщение на физический выходной интерфейс компьютера 1, и оно начинает своё «путешествие» по сети. Когда сообщение по сети поступает на входной интерфейс компьютера 2, оно принимается его физическим уровнем и последовательно перемещается вверх с уровня на уровень. Каждый уровень анализирует и обрабатывает заголовок своего уровня, выполняя соответствующие функции, а затем удаляет этот заголовок и передаёт сообщение вышележащему уровню.

Как видно из описания, протокольные сущности одного уровня не общаются между собой непосредственно, в этом сообщении всегда участвуют посредники – средства протоколов нижележащих уровней. И только физические уровни различных узлов взаимодействуют непосредственно.

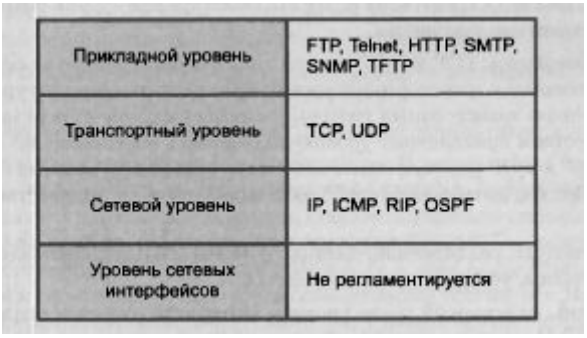
ЭТАЛОННАЯ МОДЕЛЬ TCP/IP

Эталонная модель TCP/IP представляет собой стек протоколов TCP/IP – набор сетевых протоколов передачи данных, используемых в сетях, включая сеть Интернет. Название происходит из двух наиважнейших протоколов семейства – Transmission Control Protocol (TCP) и Internet Protocol (IP), которые были разработаны и описаны первыми в данном стандарте. Важную часть технологии TCP/IP составляют задачи адресации, к числу которых относятся следующие:

- Согласованное использование адресов различного типа;
- Обеспечение уникальности адресов;
- Конфигурирование сетевых интерфейсов и сетевых приложений.

Масштабируемость – ключевое слово, которое характеризует подход к решению этих проблем, принятый в TCP/IP.

Сегодня стек TCP/IP широко используется как в глобальных, так и в локальных сетях. Этот стек имеет иерархическую структуру, в которой определено уровня.



Прикладной уровень	FTP, Telnet, HTTP, SMTP, SNMP, TFTP
Транспортный уровень	TCP, UDP
Сетевой уровень	IP, ICMP, RIP, OSPF
Уровень сетевых интерфейсов	Не регламентируется

Прикладной уровень стека TCP/IP соответствует трём верхним уровням модели OSI. Он объединяет сервисы, предоставляемые системой пользовательским приложениям.

Транспортный уровень стека TCP/IP может предоставлять вышележащему уровню два типа сервиса:

- Гарантированную доставку, которую обеспечивает протокол управления передачей (Transmission Control Protocol, TCP);
- Доставку по возможности, или с максимальными усилиями, которую обеспечивает протокол пользовательских дейтаграмм (User Datagram Protocol, UDP).

Сетевой уровень, называемый также уровнем Интернета, является стержнем всей архитектуры TCP/IP. Именно этот уровень обеспечивает перемещение пакетов в пределах составной сети, образованной объединением нескольких подсетей. Протоколы сетевого уровня поддерживают интерфейс с вышележащим транспортным уровнем, получая от него запросы на передачу данных по составной сети, а также с нижележащим уровнем сетевых интерфейсов.

Уровень сетевых интерфейсов отвечает только за организацию взаимодействия с подсетями разных технологий, входящими в составную сеть.TCP/IP рассматривает любую подсеть, входящую в составную сеть, как средство транспортировки пакетов между двумя соседними маршрутизаторами.

КРИТИКА ЭТАЛОННЫХ МОДЕЛЕЙ OSI И TCP/IP.

Недостатки эталонной модели OSI:

- Несвоевременность (к моменту её появления среди исследовательских университетов уже получили широкое распространение протоколы TCP/IP);
- Плохая технология (невероятно сложная, тяжело реализуемая, неэффективная в работе, сеансовый и уровень представления – почти пустые, сетевой и передачи данных - перегружены);
- Неудачная реализация (громоздко и медленно);
- Неудачная политика.

Критика эталонной модели TCP/IP:

1 Нет чёткого разграничения концепций служб, интерфейсов и протоколов (в результате модель TCP/IP довольно бесполезна при разработке сетей, использующих новые технологии);
2 Не является общей и довольно плохо описывает любой стек протоколов, кроме TCP/IP;
3 Канальный уровень в действительности не является уровнем в том смысле, который обычно используется в контексте уровневых протоколов;
4 В модели TCP/IP не различаются физический уровень и уровень передачи данных.
5 Многие другие протоколы устарели (TELNET).

ГИБРИДНАЯ ЭТАЛОННАЯ МОДЕЛЬ

Несмотря на то, что эталонная модель OSI пользуется большой популярностью в информационных сетях, её протоколы не получили широкого распространения. Для TCP/IP верно обратное утверждение: модель практически не существует, тогда как протоколы чрезвычайно популярны. Поэтому на практике используется так называемая гибридная эталонная модель, состоящая из пяти уровней: *физический, канальный, сетевой, транспортный и прикладной.*

Первые четыре уровня выполняют те же функции, что и одноимённые уровни эталонной модели OSI. Прикладной уровень объединяет в себя три верхних уровня (прикладной, представления и сеансовый) модели OSI. В качестве протоколов взаимодействия между уровнями гибридной эталонной модели выступают протоколы стека протоколов TCP/IP.

ФИЗИЧЕСКИЙ УРОВЕНЬ

Физический уровень имеет дело с передачей потока битов по физическим каналам связи, таким как коаксиальный кабель, витая пара, оптоволоконный кабель или цифровой территориальный канал. Функции физического уровня реализуются на всех устройствах, подключённых к сети. Со стороны компьютера функции физического уровня выполняются сетевым адаптером или последовательным портом.

Физический уровень определяет электрические, временные и прочие характеристики сетей, по которым биты информации пересылаются в форме электрических сигналов.

Физический уровень не вникает в смысл информации, которую он передаёт. Для него эта информация представляет собой однородный поток битов, который нужно доставить без искажений и в соответствии с заданной тактовой частотой (интервалом между соседними битами).

ФИЗИЧЕСКИЕ НОСИТЕЛИ

Для передачи битов от одной машины к другой могут использоваться различные физические носители информации, называемые также средой распространения сигнала. Каждый из них имеет характерный набор полос пропускания, задержек, цен и простоты установки и использования. Среды передачи информации можно разделить на две группы: проводниковые среды, такие как медный провод и оптоволоконный кабель, и беспроводные, например, предназначенные для наземной беспроводной и спутниковой связи, а также передача по лазерному лучу без кабеля.

Проводниковые среды:
Магнитные носители;
Витая пара;
Коаксиальный кабель;
Линии электропитания;
Оптоволоконные кабели.

Беспроводные среды:
Электромагнитный спектр;
Радиосвязь;
Микроволновый диапазон;
Инфракрасный диапазон;
Видимый диапазон.

КАНАЛЬНЫЙ УРОВЕНЬ

Канальный уровень обеспечивает прозрачность соединения для сетевого уровня. Для этого он предлагает ему следующие услуги:

- Установление логического соединения между взаимодействующими узлами;
- Согласование в рамках соединения скоростей передатчика и приёмника информации;
- Обеспечение надёжной передачи, обнаружение и коррекция ошибок.

Канальный уровень использует определённые службы физического уровня для отправки и получения битов по коммуникационным каналам. Его функции:

- Обеспечение строго очерченного служебного интерфейса для сетевого уровня;
- Обработка ошибок передачи данных;
- Управление потоком данных, исключающее затопление медленных приёмников быстрыми передатчиками.

Для решения этих задач канальный уровень формирует из пакетов, полученных с сетевого уровня, собственные протокольные единицы данных – кадры, состоящие из поля данных, заголовка и концевика. Канальный уровень помещает пакет в поле данных одного или нескольких кадров и заполняет собственной служебной информацией заголовок кадра. Управление кадрами – это основа деятельности канального уровня.

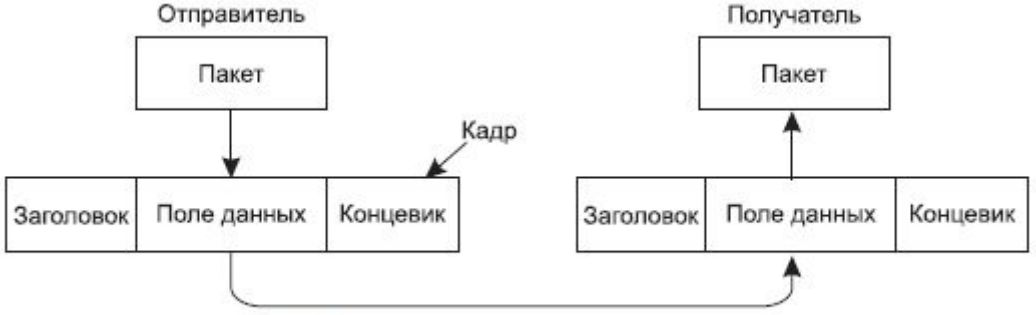


Рис. 3.1. Соотношение между пакетами и кадрами

СЛУЖБЫ КАНАЛЬНОГО УРОВНЯ

- Формирование кадра. Почти все протоколы канального уровня помещают каждую дейтаграмму сетевого уровня в кадр канального уровня, перед тем как отправить её по каналу. Кадр состоит из поля данных, в которое помещается дейтаграмма сетевого уровня, и нескольких полей заголовка. Структура кадра специфицируется канальным протоколом передачи данных.
- Доступ к каналу связи. Протокол управления доступом к среде передачи (MAC, Media Access Control) определяет правила передачи кадра в канал.

- Надёжная доставка. Когда протокол канального уровня предоставляет услугу по надёжной доставке, он гарантирует перемещение каждой дейтаграммы сетевого уровня по линии связи без ошибок.
- Обнаружение и исправление ошибок.

СЕТЕВЫЕ АДАПТЕРЫ

Сетевые адаптеры или NIC (Network Interface Card) – это сетевое оборудование, обеспечивающее функционирование сети на физическом и канальном уровнях.

Сетевой адаптер относится к периферийному устройству компьютера, непосредственно взаимодействующему со средой передачи данных, которая прямо или через другое коммуникационное оборудование связывает его с другими компьютерами. Это устройство решает задачи надёжного обмена двоичными данными, представленными соответствующими электромагнитными сигналами, по внешним линиям связи. Сетевой адаптер работает под управлением драйвера операционной системы.

Функции сетевого адаптера:

1 Гальваническая развязка с коаксиальным кабелем или витой парой;

2 Буферизация. Для согласования скоростей пересылки данных в адаптер или из него со скоростью обмена по сети используются буфера. Во время обработки в сетевом адаптере, данные хранятся в буфере. Буфер позволяет адаптеру осуществлять доступ ко всему пакету информации. Использование буферов необходимо для согласования между собой скоростей обарботки информации различными компонентами ЛВС.

3 Приём (передача) данных. Данные передаются из ОЗУ ПК в адаптер или из адаптера в память ПК через программируемый канал ввода-вывода, канал прямого доступа и разделяемую память.

4 Формирование пакета. Сетевой адаптер должен разделить данные на блоки в режиме передачи (или соединить их в режиме приёма) данных и оформить в виде кадра определённого формата. Кадр включает несколько служебных полей, среди которых имеется адрес компьютера назначения и контрольная сумма кадра, по которой сетевой адаптер станции назначения делает вывод о корректности доставленной по сети информации.

5 Доступ к каналу связи (к среде передачи). Выявление конфликтных ситуаций и контроль состояния сети.

6 Идентификация своего адреса в принимаемом пакете.

7 Преобразование параллельного кода в последовательный код при передаче данных, и из последовательного кода в параллельный при приёме. В режиме передачи данные передаются по каналу связи в последовательном коде.

8 Кодирование и декодирование данных. (Большинство сетевых адаптеров используют манчестерское кодирование)

9 Передача или приём импульсов

Сетевые адаптеры вместе с сетевым программным обеспечением способны распознавать и обрабатывать ошибки, которые могут возникнуть из-за электрических помех, коллизий или плохой работы оборудования.

ОБНАРУЖЕНИЕ И ИСПРАВЛЕНИЕ ОШИБОК

Разработчики сетей создали две основные стратегии для борьбы с ошибками. Каждый метод основывается на добавлении к передаваемым данным некоторой избыточной информации. В одном случае этой информации должно быть достаточно, чтобы принимающая сторона могла выявить, какие данные должны были прийти. В другом случае избыточной информации должно быть достаточно только для того, чтобы получатель понял, что произошла ошибка (без указания её типа), и запросил повторную передачу. Первая стратегия использует коды, называющиеся корректирующими или

кодами с исправлением ошибок (error-correcting codes). Вторая – коды с обнаружением ошибок (error-detecting codes). Использование кода с исправлением ошибок часто называют прямым исправлением ошибок (Forward Error Correction - FEC).

Коды с исправлением ошибок:

1 Коды Хэмминга;
2 Двоичные свёрточные коды;
3 Коды Рида-Соломона;
4 Коды с малой плотностью проверок на чётность.

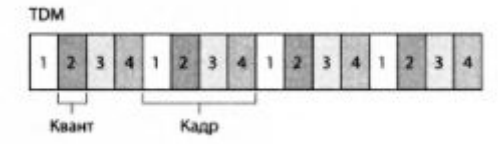
Коды с обнаружением ошибок:

1 Код с проверкой на чётность;
2 Код с контрольными суммами;
3 Циклический избыточный код.

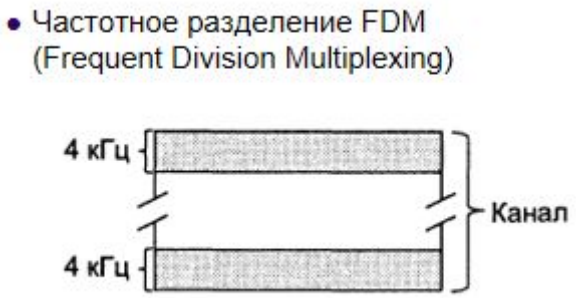
ПРОТОКОЛЫ РАЗДЕЛЕНИЯ КАНАЛА

- TDMA;
- FDMA;
- CDMA;
- SDMA

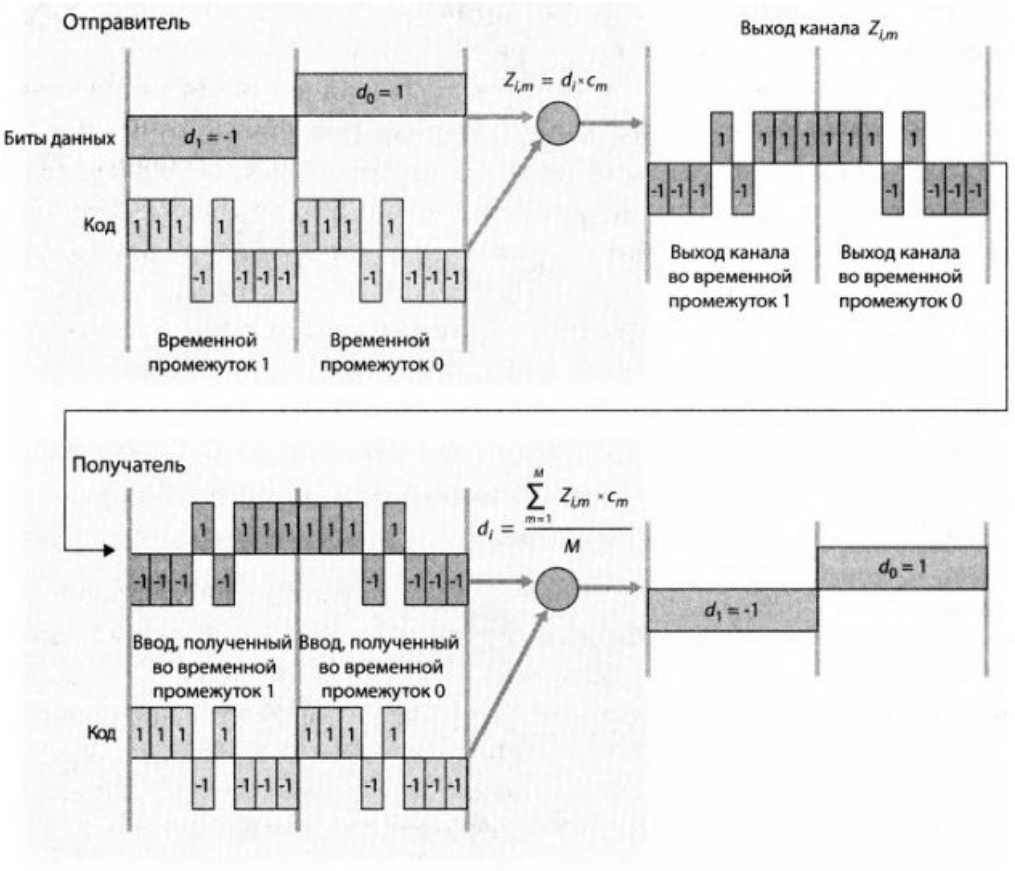
Протоколы разделения каналов используют идеологию коммутации каналов. Данные протоколы разделяют общий канал на столько частей, каково количество компьютеров (узлов). Например, канал поддерживает N узлов и скорость передачи данных в канале равна R бит/с. При временном разделении канала время делится на интервалы, называемые кадрами, каждый из которых делится на N элементарных интервалов времени, называемые кадрами. Затем каждому из N узлов назначается один временной квант. Когда у узла есть кадр для отправки, он передаёт биты этого кадра в течение назначенного ему элементарного интервала времени. Как правило, длина кванта выбирается таким образом, чтобы за этот интервал можно было передать один кадр. Временное разделение TDM (Tine Division Multiplexing). Привлекательность временного разделения канала заключается в том, что такая схема полностью устраняет конфликты (коллизии) и обладает идеальной справедливостью: каждый узел получает выделенную скорость передачи данных, равную R/N бит/с в течение каждого временного кадра. Недостатки: каждый узел ограничен средней скоростью передачи данных, даже в том случае, когда этот узел единственный, кому нужно отсылать данные в этот момент. Во-вторых, при передаче узел всегда должен ждать своей очереди, даже когда кроме него никто не отправляет данные.



Метод мультиплексирования с частотным разделением делит канал с пропускной способностью R бит/с на частотные диапазоны с полосой пропускания R/N бит/с, при этом каждому узлу выделяется собственный частотный диапазон. Таким образом, при методе частотного разделения из одного канала с пропускной способностью R бит/с создаётся N каналов с пропускной способностью R/N бит/с. Мультиплексирование с частотным разделением канала обладает теми же преимуществами и недостатками, что и с временным.



Протокол множественного доступа с кодовым разделением (Code Division Multiple Access, CDMA). Этот протокол назначает каждому узлу собственный код. Затем каждый узел использует этот уникальный код для кодирования передаваемых данных. Протокол CDMA позволяет нескольким узлам передавать данные одновременно, при этом получатели могут корректно их принимать (при условии, что получателю известен код передатчика), даже при воздействии помех со стороны других узлов. Протокол CDMA в течении некоторого времени использовался в военных системах связи (благодаря своей устойчивости к попыткам подавления сигнала), а в настоящее время получает всё более широкое распространение в гражданских беспроводных средствах связи коллективного доступа.



Пространственное разделение канала SDMA – Space Division Multiple Access. Деление региона на соты. Использование направленных антенн, излучающих сигнал по выделенным секторам.

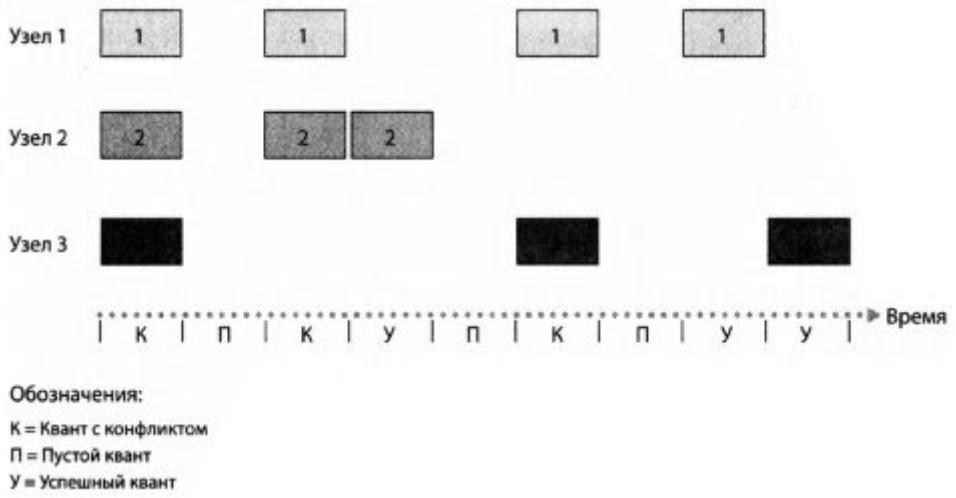
ПРОТОКОЛЫ ПРОИЗВОЛЬНОГО ДОСТУПА

В протоколе произвольного доступа передающий узел всегда передаёт данные в канал с максимальной скоростью, то есть R бит/с. Когда возникает коллизия, каждый вовлечённый в неё узел отправляет свой кадр (т.е. пакет) повторно до тех пор, пока ему не

удастся осуществить передачу без коллизий. Однако, претерпев коллизию, узел, как правило, не повторяет передачу сразу же, а выжидает в течение случайного интервала времени. Благодаря разной длительности случайных интервалов времени существует ненулевая вероятность того, что интервал, выбранный одним из узлов, окажется меньше, чем у других вовлечённых в коллизию узлов, и он успеет вытолкнуть свой кадр в канал беспрепятственно.

Работа дискретного протокола ALOHA:

- Когда у узла появляется новый кадр для отправки, он дожидается начала следующего кванта времени и за этот квант передаёт весь кадр;
- Если коллизия не возникает, это значит, что узел успешно, это значит, что узел успешно передал свой кадр и повторная передача не требуется;
- Если коллизия возникает, то узел замечает её до окончания кванта времени. Узел будет пытаться передать свой кадр в течение последующих квантов времени, что рано или поздно удастся ему с вероятностью p.



В так называемом чистом протоколе ALOHA, когда прибывает первый кадр (то есть дейтаграмма сетевого уровня передаётся на более низкий уровень передающего узла), узел немедленного передаёт весь кадр целиком в широковещательный канал. Если переданный кадр сталкивается с одним или несколькими другими кадрами (происходит коллизия), с вероятностью p узел немедленно передаёт кадр повторно. В противном случае узел выжидает в течение времени, необходимого для передачи одного кадра, после чего опять с вероятностью p передаёт кадр либо пережидает ещё один интервал времени с вероятностью 1-p.

В семействе протоколов CSMA (Carrier Sense Multiple Access – множественный доступ с контролем несущей) и CSMA/CD (CSMA with Collision Detection – множественный доступ с контролем несущей и обнаружением коллизий). Реализованы 2 правила:

- Слушайте, прежде чем говорить. Если кто-то уже говорит, подождите, пока он не закончит. Это правило называется контролем несущей и заключается в том, что узел прослушивает канала перед тем, как начать передачу. Если по каналу передаётся кадр, узел выжидает в течение случайного периода времени, а затем начинает передачу.
- Если кто-то начал говорить, прекращайте разговор. Это правило называется обнаружение коллизий. Оно заключается в том, что во время передачи узел прослушивает канал. Если он обнаруживает, что другой узел в этот момент времени тоже ведёт передачу, он прекращает свою и выжидает в течение случайного периода времени, после чего начинает новый цикл проверки канада и передачи, если канал оказывается не занят.

ПРОТОКОЛЫ ПОСЛЕДОВАТЕЛЬНОГО ДОСТУПА

Протоколы последовательного доступа позволяют безопасно разделять или полностью использовать канал, если он не занят.

- Протокол опроса: главный узел сети по очереди разрешает узлам передавать данные;
- Протокол передачи маркера: маркер (спец. кадр) передаётся от одного узла к другому.

СЕТЕВЫЕ УСТРОЙСТВА: КОНЦЕНТРАТОРЫ, КОММУТАТОРЫ, МОСТЫ.

Концентратор (hub) – центральное устройство, работающее на физическом (первом) уровне модели OSI и повторяющее сигналы, поступившие с одного из его портов на все остальные активные порты. Использование концентраторов позволило повысить надёжность сети, так как обрыв одного из кабелей не приводил к сбою в работе сети.

Мост (bridge) представляет собой устройство канального (второго) уровня модели OSI (обычно двухпортовое), предназначенное для объединения сегментов сети. В отличие от концентратора мост не просто пересылал пакеты данных из одного сегманта в другой, а анализировал и передавал их только в том случае, если такая передача была действительно необходима, т.е. если адрес рабочей станции назначения принадлежал другому сегменту. Таким образом, мост изолировал трафик одного сегмента от другого, уменьшая домен коллизий и повышая общую производительность сети.

Коммутатор (switch) представляет собой многопортовый мост, функционирующий также на канальном уровне модели OSI. Основное отличие коммутатора от моста заключается втом, что он мог одновременно устанавливать соединения между разными парами портов.

СЕТЕВОЙ УРОВЕНЬ.

Сетевой уровень (Network layer) служит для образования единой транспортной системы, объединяющей несколько сетей (называемой составной сетью), причём эти сети могут использовать совершенно различные принципы передачи сообщений между конечными узлами и обладать произвольной структурой связей.

Сетевой уровень занимается разработкой маршрутов доставки пакетов от отправителя до получателя. Чтобы добраться до пункта назначения, пакету может потребоваться преодолеть несколько транзитных участков между маршрутизаторами.

Для достижения этих целей сетевой уровень должен обладать информацией о топологии сети (т.е. о множестве всех маршрутизаторов и связей) и выбирать нужный путь по этой сети. При выборе маршрутизаторов он должен также заботиться о том, чтобы нагрузка на них и линии связи была, по возможности, более равномерной. Наконец, если источник и приёмник находятся в различных сетях, именно сетевой уровень должен уметь решать проблемы, связанные с различиями в сетях.

МОДЕЛИ СЕТЕВОГО ОБСЛУЖИВАНИЯ

Модель сетевого обслуживания на основе виртуальных каналов (3 фазы):

1. Установка виртуального канала. Во время этой фазы отправитель связывается с сетевым уровнем, указывает адрес получателя и ждёт, пока сеть установит виртуальный канал. Сетевой уровень определяет путь от отправителя до получателя, то есть последовательность линий связи и пакетных коммутаторов, через которые будут проходить все пакеты данного виртуального канала.
2. Передача данных. Как только виртуальный канала установлен, данные могут начать перемещение по виртуальному каналу.

3. Разрыв виртуального канала. Эта процедура начинается, когда отправитель (или получатель) информирует сетевой уровень о своём желании разорвать виртуальный канал. Затем, как правило, сетевой уровень информирует оконченную систему на другой стороне сети о разрыве соединения и обновляет таблицы в каждом пакетном коммутаторе пути, показывая, что виртуального канала больше не существует.

Модель сетевого обслуживания на основе дейтаграмм: каждый раз, когда оконечная система хочет послать пакет, она указывает в нём адрес получающей оконечной системы, а затем передаёт этот пакет в сеть (канальному уровню). Эта процедура выполняется без предварительной установки виртуального канала. Коммутаторы пакетов в дейтаграммной сети (маршрутизаторы) не содержат информации о состоянии виртуальных каналов. Вместо этого коммутаторыпакетов продвигают пакет по направлению к адресату, изучая адрес получателя пакета. При этом они ищут нужную им для этого информацию в своей таблице продвижения данных, используя адрес получателя в качестве индекса. Посколько таблицы продвижения данных могут быть изменены в любое время, пакеты, относящиеся к одной серии пакетов, посланных одной оконечной системы другой оконечной системе, могут следовать по разным маршрутам и прибыть к получателю не в том порядке, в котором они были посланы.

СЛУЖБЫ СЕТЕВОГО УРОВНЯ

В системе-отправителе, когда пакет передаётся с транспортного уровня на сетевой, тот может предоставить следующие службы (функции):

- Гарантированная доставка;
- Гарантированная доставка с ограниченной задержкой

Существуют также службы, предназначенные для организации потока пакетов по пути от источника до места назначения:

- Упорядоченная доставка пакетов;
- Гарантированная минимальная пропускная способность;
- Гарантированный максимальный джиттер (интервалы времени между последовательной передачей двух пакетов отправителем будут равняться интервалам между их приёмом, либо интервалы могут отличаться в пределах указанного значения);
- Службы безопасности (Используя секретный сеансовый ключ, известный только системе-отправителю и системе-получателю, сетевой уровень отправителя может шифровать содержимое всех дейтаграмм, посылаемых в пункт назначения. Сетевой уровень получателя в таком случае будет отвечать за расшифровку содержимого. Данная служба обеспечивает конфиденциальность всех сегментов транспортного уровня между источником и получателем, а также целостность данных и работу служб аутентификации).

ИНТЕРНЕТ-ПРОТОКОЛ

Протокол межсетевого взаимодействия (Internet Protocol, IP) обеспечивает передачу дейтаграмм от отправителя к получателям через объединённую систему компьютерных сетей.

Название протокола отражает его суть: он должен передавать пакеты между сетями. В каждой очередной сети, лежащей на пути перемещения пакета, протокол IP вызывает средства транспортировки, принятые в этой сети, чтобы с их помощью передать этот пакет на маршрутизатор, ведущий к следующей сети, или непосредственно на узел получателя.

Протокол IP относится к протоколам без установления соединения. Перед IP не ставится задача надёжной доставки сообщений от отправителя к получателю. Протокол IP обрабатывает каждый IP-пакет как независимую единицу, не имеющую связи ни с какими другими IP-пакетами. В протоколе IP нет механизмов, обычно применяемых для увеличения достоверности конечных данных. Если во время продвижения пакета произошла какая-либо ошибка, то протокол IP по своей инициативе ничего не предпринимает для исправления этой ошибки (все вопросы решает протокол TCP, работающий непосредственно над протоколом IP).

Важной особенностью протокола IP, отличающей его от других сетевых протоколов, является его способность выполнять динамическую фрагментацию пакетов при передаче их между сетями с различными, максимально допустимыми значениями поля данных кадров MTU (Maximum Transmission Unit).

АДРЕСАЦИЯ В ПРОТОКОЛЕ IPv4

Адрес IPv4 представляет собой 32-разрядное (4 байта) двоичное поле. Для удобства восприятия и запоминания этот адрес разделяют на 4 части по 8 бит (октеты), каждый октет переводят в десятичное число и при записи октеты разделяют точками. Это представление адреса называется десятично-точечной нотацией. (Н-р, 192.168.1.12)

Следует отметить, что максимальное значение октета равно 11111111=255, поэтому IP-адреса, в которых хотя бы один октет превышает максимальное значение, считаются недействительным.

Маршрутизация пакетов в сетях передачи данных возможна благодаря тому, что IPv4-адрес структурирован и состоит из двух логических частей: идентификатор сети (NetID) – сетевая часть адреса и идентификатора узла (HostID), который однозначно определяет устройство в сетевом сегменте. Такая структура IP-адреса представляет собой двухуровневую иерархическую модель и позволяет устройству при передаче данных в составную сеть указывать не только удалённую сеть, но и узел этой сети.

Идентификатор сети определяет конкретную сеть или сегмент сети, в которой находится узел и используется для передачи данных на нужный сетевой интерфейс маршрутизатора или L3-коммутатора.

После того как данные достигают нужной сети, они передаются уникальному узлу в соответствии с идентификатором узла. Все узлы, использующие один и тот же идентификатор сети, должны быть расположены в одной сети или подсети (логическом сегменте сети).

ФРАГМЕНТАЦИЯ ДЕЙТАГРАММ

Процедура фрагментации заключается в разбивке единой дейтаграммы на несколько частей, которые позже будут собраны вместе. Каждый из фрагментов должен быть снабжён полноценным заголовком IP. Некоторые из полей заголовка (идентификатор, TTL, флаги DF и MF, смещение) непосредственно предназначены для последующей сборки фрагментов в исходное сообщение.

- Идентификатор пакета используется для распознавания пакетов, образовавшихся путём деления на части исходного пакета. Все части одного пакета должны иметь одинаковое значение этого поля, уникальное по отношению к другим пакетам.
- Поле времени жизни (Time To Live - TTL) занимает один байт и определяет предельный срок, в течение которого пакет может перемещаться по сети. При сборке фрагментов хост-получатель использует значение TTL как крайний срок ожидания недостающих фрагментов.
- Поле смещения фрагмента предоставляет получателю информацию о положении фрагмента относительно начала поля данных исходного нефрагментированного

пакета. (первый или не разбитый будут иметь нулевое значение). Смещение в байтах кратное 8.

- Флаг MF (More Fragments – больше фрагментов) установленный в 1 говорит о том, что данный пакет является промежуточным (не последним). 0 –последний или нефрагментированный.
- Флаг DF (Do not Fragment – не фрагментировать), установленный в единицу, запрещает маршрутизатору фрагментировать данный пакет. Если помеченный таким образом пакет не может достигнуть получателя без фрагментации, то модуль IP его уничтожает, а узлу-отправителю посылается диагностическое сообщение.

ОСНОВЫ МАРШРУТИЗАЦИИ

Важнейшей задачей сетевого уровня является маршрутизация – передача пакетов между двумя конечными узлами в составной сети.

В сложных составных сетях почти всегда существует несколько альтернативных маршрутов для передачи пакетов между двумя конечными узлами. Маршрут – это последовательность маршрутизаторов, который должен пройти пакет от отправителя до пункта назначения.

Задачу выбора маршрута из нескольких возможных решают маршрутизаторы, а также конечные узлы. Маршрут выбирается на основании имеющейся у этих устройств информации о текущей конфигурации сети, а также на основании указанного критерия выбора маршрута. Обычно в качестве критерия выступает задержка прохождения маршрута отдельным пакетом или средняя пропускная способность маршрута для последовательности пакетов. Часто также используется весьма простой критерий, учитывающий только количество пройденных в маршруте промежуточных маршрутизаторов (хопов).

Чтобы по адресу сети назначения можно было бы выбрать рациональный маршрут дальнейшего следования пакета, каждый конечный узел и маршрутизатор анализируют специальную информационную структуру, которая называется таблицей маршрутизации. Она хранится на маршрутизаторе и содержит записи, представляющие собой список наилучших маршрутов в соответствующие сети. В том случае, если к сети назначения имеется несколько путей, в таблицу маршрутизации будет помещён маршрут, у которого наилучшая метрика, определяемая на основании загрузки, полосы пропускания, задержки, стоимости или надёжности канала связи.

КЛАССИФИКАЦИЯ АЛГОРИТМОВ МАРШРУТИЗАЦИИ

Одношаговые алгоритмы маршрутизации – каждый маршрутизатор ответственен за выбор только одного шага маршрута, а окончательный маршрут складывается в результате работы всех маршрутизаторов, через которые проходит данный пакет.

Маршрутизация от источника – узел-источник задаёт 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.

Одношаговые алгоритмы выбора маршрута, в зависимости от способа формирования таблиц маршрутизации, можно разбить на три основных класса: неадаптивные (статические), простые и адаптивные (динамические).

Неадаптивные алгоритмы не учитывают при выборе маршрута топологию и текущее состояние сети и не измеряют трафик на линиях. Вместо этого выбор маршрута для каждой пары станций производится заранее, в автономном режиме, и список маршрутов загружается в маршрутизаторы во время загрузки сети. Такая процедура иногда называется статической маршрутизацией.

В алгоритмах простой маршрутизации таблица маршрутизации либо вовсе не используется, либо строится без участия протоколов маршрутизации. Выделяют три типа простой маршрутизации:

- Случайная маршрутизация, когда прибывший пакет посылается в первом попавшемся случайном направлении, кроме исходного;
- Лавинная маршрутизация, когда пакет широковещательно посылается по всем возможным направлениям, кроме исходного;
- Маршрутизация по предыдущему опыту, когда выбор маршрута осуществляется по таблице, но таблица строится по принципу моста путём анализа адресных полей пакетов, появляющихся на входных портах.

Адаптивные алгоритмы, напротив, изменяют решение о выборе маршрутов при изменении топологии и также иногда в зависимости от загруженности линий. Эти динамические алгоритмы маршрутизации отличаются источниками получения информации, моментами изменения маршрутов и данными, использующимися для оптимизации. Они обычно имеют распределённый характер, который выражается в том, что в сети отсутствуют какие-либо выделенные маршрутизаторы, которые собирали бы и обобщали топологическую информацию: эта работа распределена между всеми маршрутизаторами.

Адаптивные алгоритмы маршрутизации должны обеспечивать, если не оптимальность, то хотя бы рациональность маршрута. Во-вторых, алгоритмы должны быть достаточно простыми, чтобы при их реализации не тратилось слишком много сетевых ресурсов, в частности они не должны требовать слишком большого объёма вычислений или порождать интенсивный служебный трафик. И наконец, алгоритмы маршрутизации должны обладать свойством сходимости, то есть всегда приводить к однозначному результату за приемлемое время.

Адаптивные алгоритмы маршрутизации в свою очередь делятся на две группы:

- Дистанционно-векторные алгоритмы;
- Алгоритмы состояния канала.

АЛГОРИТМЫ ДИНАМИЧЕСКОЙ МАРШРУТИЗАЦИИ: НА ОСНОВЕ ВЕКТОРА РАССТОЯНИЙ И НА ОСНОВЕ СОСТОЯНИЯ КАНАЛА

В алгоритмах дистанционно-векторного типа каждый маршрутизатор периодически и широковещательно рассылает по сети вектор, компонентами которого являются расстояния от данного маршрутизатора до всех известных ему сетей. Под расстоянием обычно понимается число хопов (промежуточных маршрутизаторов). Возможна и другая метрика, учитывающая не только число промежуточных маршрутизаторов, но и время прохождения пакетов по сети между соседними маршрутизаторами. При получении вектора от соседа маршрутизатор наращивает расстояния до указанных в векторе сетей на расстояние до данного соседа. Получив вектор от соседнего маршрутизатора, каждый маршрутизатор добавляет к нему информацию об известных ему других сетях, о которых он узнал непосредственно (если они подключены к его портам) или из аналогичных объявлений других маршрутизаторов, а затем снова рассылает новое значение вектора по сети. В конце концов, каждый маршрутизатор узнаёт информацию обо всех имеющихся в интерсети сетях и о расстоянии до них через соседние маршрутизаторы. Наиболее распространённым протоколом, основанным на дистанционно-векторном алгоритме, является протокол RIP.

Алгоритмы состояния связей обеспечивают каждый маршрутизатор информацией, достаточной для построения точного графа связей сети. Каждый маршрутизатор должен:

1 Обнаруживать своих соседей и узнавать их сетевые адреса;
2 Задавать метрику расстояния или стоимости связи с каждым из своих соседей;

3 Создавать пакет, содержащий всю собранную информацию;
4 Посылать этот пакет всем маршрутизаторам и принять все пакеты, отправленные другими маршрутизаторами
5 Вычислять кратчайший путь ко всем маршрутизаторам.

В результате каждому маршрутизатору высылается полная топология. После этого для обнаружения кратчайшего пути ко всем остальным маршрутизаторам каждый маршрутизатор может использовать алгоритм Дейкстры.

Протоколами, основанными на алгоритме состояний связей, являются протоколы IS-IS (Intermediate System to Intermediate System) стека OSI, OSPF (Open Shortest Path First) стека TCP/IP.

ПРОТОКОЛЫ МАРШРУТИЗАЦИИ RIP, OSPF

Протокол RIP (Routing Information Protocol) является представителем класса IGP (Interior Gateway Protocol – внутренний, внутришлюзовый протокол)-протоколов стека TCP/IP. Этот протокол используется в небольших однородных сетях, то есть имеющих одинаковые характеристики каналов связи, где самый длинный путь между любыми сетями составляет максимум 15 переходов.

Протокол RIP основан на дистанционно-векторном алгоритме маршрутизации и в качестве метрики при выборе маршрута использует количество переходов (hops count). Он не учитывает ситуации, когда маршрут должен быть выбран на основе таких параметров, как загруженность канала, надёжность или задержка передачи. Если маршрутизатор непосредственно подключён к сети, то расстояние до неё (количество переходов) равно 1. По умолчанию маршрутизаторы, использующие протокол RIP, отправляют на широковещательный адрес своим соседям обновления с маршрутной информацией каждые 30 секунд. При получении обновления от соседа маршрутизатор заносит новые записи в таблицу маршрутизации и увеличивает метрику (число переходов) к соответствующей сети на 1 (по умолчанию).

В настоящее время существует три версии протокола:
1 RIP версии 1 (RIPv1) используется для поддержки классовой адресации протокола IPv4;
2 RIP версии 2 (RIPv2) используется для поддержки бесклассовой адресации IPv4;
3 RIPng (next generation) используется для протокола IPv6.

Протокол OSPF (Open Shortest Path First – открытый протокол «кратчайший путь первым») является достаточно современной реализацией алгоритма состояния связей и обладает особенностями, ориентированными на применение в больших гетерогенных сетях. OSPF представляет собой протокол, основанный на учёте состояний каналов и использующих метод лавинной рассылки для распространения информации о состоянии каналов, а также алгоритм определения пути наименьшей стоимости Дейкстры. Маршрутизатор, работающий по протоколу OSPF, формирует полную топологическую карту (направленный граф) всей автономной системы. Затем маршрутизатор локально запускает алгоритм определения кратчайшего пути Дейкстры, чтобы найти дерево кратчайших путей ко всем сетям автономной системы. Далее из этого дерева кратчайших путей формируется таблица перенаправления маршрутизатора.

Маршрутизатор, работающий по протоколу OSPF, путём широковещательной рассылки переправляет информацию о маршрутах всем маршрутизаторам автономной системы, а не только соседним. Маршрутизатор рассылает всем информацию о состоянии каналов при каждом изменении состояния какого-либо из них (или раз в 30 минут, даже если состояние линии не изменилось). Протокол OSPF также проверяет работоспособность каналов (при помощи сообщения HELLO, посылаемого соседу) и позволяет маршрутизатору OSPF получать информацию из базы данных соседнего маршрутизатора о состоянии каналов всей сети.

АЛГОРИТМЫ ПОСТРОЕНИЯ ТАБЛИЦЫ МАРШРУТОВ ПРОТОКОЛА RIP

Этап 1 – создание минимальных таблиц.

В исходном состоянии в каждом маршрутизаторе программным обеспечением стека TCP/IP автоматически создаётся минимальная таблица маршрутизации, в которой учитываются только непосредственно подсоединённые сети.

Этап 2 – рассылка минимальных таблиц соседям.

После инициализации каждого маршрутизатора он начинает посылать своим соседям сообщения протокола RIP, в которых содержится его минимальная таблица. RIP-сообщения передаются в пакетах протокола UDP и включают два параметра для каждой сети: её IP-адрес и расстояние до неё от передающего сообщения маршрутизатора. Соседями являются те маршрутизаторы, которым данный маршрутизатор непосредственно может передать IP-пакет по какой-либо своей сети, не пользуясь услугами промежуточных маршрутизаторов.

Этап 3 – получение RIP-сообщений от соседей и обработка полученной информации.

После получения аналогичных сообщений от соседей, маршрутизатор наращивает каждое полученное поле метрики на единицу и запоминает, через какой порт и от какого маршрутизатора получена новая информация. Затем маршрутизатор начинает сравнивать новую информацию с той, которая хранится в его таблице маршрутизации.

Протокол RIP замещает запись о какой-либо сети только в том случае, если новая информация имеет лучшую метрику (расстояние в хопах меньше), чем имеющаяся. В результате в таблице маршрутизации о каждой сети остаётся только одна запись.

Аналогичные операции с новой информацией выполняют и остальные маршрутизаторы сети.

Этап 4 – рассылка новой, уже не минимальной таблицы соседям.

Каждый маршрутизатор отсылает новое RIP-сообщение всем своим соседям. В этом сообщении он помещает данные о всех известных ему сетях – как непосредственно подключённых, так и удалённых, о которых маршрутизатор узнал из RIP-сообщений.

Этап 5 - получение RIP-сообщений от соседей и обработка полученной информации.

Этап 5 повторяет этап 3 – маршрутизаторы принимают RIP-сообщения, обрабатывают содержащуюся в них информацию и на её основании корректируют свои таблицы маршрутизации.

УСТРОЙСТВО МАРШРУТИЗАТОРОВ

Основная функция маршрутизатора – чтение заголовков пакетов сетевых протоколов, принимаемых и буферизуемых по каждому парту, и принятие решения о дальнейшем маршруте следования пакета по его сетевому адресу, включающему, как правило, номер сети и номер узла.

Основные компоненты маршрутизатора:

- Входные порты. Они выполняют несколько ключевых функций. Они функционируют на физическом уровне, завершая входящую физическую связь в маршрутизаторе. Входной порт также выполняет функции сетевого уровня, требуемые для взаимодействия с сетевым уровнем на другой стороне входящего соединения. Функции поиска и выбора. Именно в этом компоненте маршрутизатор сверяется с таблицей перенаправления и определяет выходной порт, в который прибывший пакет будет перенаправлен через коммутирующую матрицу. Управляющие пакеты (например, пакеты, содержащие информацию о протоколе маршрутизации) передаются из входного пора в процессор маршрутизации.

- Коммутирующая матрица. Она соединяет входные порты маршрутизатора с выходными портами. Матрица конкретного маршрутизатора работает только в его пределах, создавая условную сеть внутри сетевого маршрутизатора.
- Выходные порты. Выходной порт хранит пакеты, полученные от коммутирующей матрицы, и передаёт их в исходящий канал, выполняя функции сетевого и физического уровней. Когда связь двунаправленная, выходной порт будет, как правило, спарен с входным портом того же канала на интерфейсной плате.
- Процессор маршрутизации. Он выполняет протоколы маршрутизации, обрабатывает таблицы маршрутизации и прилагаемую информацию о состоянии соединения, а также составляет таблицу перенаправления для маршрутизатора. Кроме того, он выполняет функции управления сетью.

КЛАССЫ СЕТЕЙ

Согласно классовой модели всё пространство IP-адресов делится на 5 классов в зависимости от значения первых четырёх бит IPv4-адреса. Классам присвоены имена от A до E.

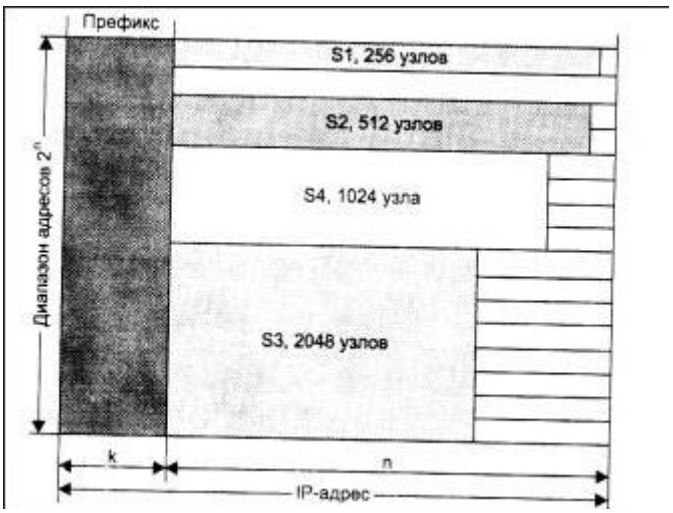
Первые 3 класса – A, B и C используются для индивидуальной адресации сетей и узлов, класс D – для многоадресной или групповой рассылки, а класс E зарезервирован для экспериментов.

Класс	Первые биты	Количество байт под сеть	Количество байт под узел
A	0	1	3
B	10	2	2
C	110	3	1
D	1110	Адрес многоадресной группы	
E	1111	Зарезервировано	

CIDR

Технология бесклассовой междоменной маршрутизации (Classless Inter Domain Routing). Суть технологии: каждому поставщику услуг Internet должен назначиться непрерывный диапазон в простарнстве IP-адресов. При таком подходе адреса всех сетей каждого поставщика услуг имеют общую старшую часть – префикс, поэтому маршрутизация на магистралях Internet может осуществляться на основе префиксов, а не полных адресов сетей. Агрегирование адресов позволит уменьшить объём таблиц в маршрутизаторах всех уровней, а следовательно ускорить работу маршрутизаторов и повысить пропускную способность Internet.

Деление IP-адреса на номер сети и номер узла в технологии CIDR происходит не на основе нескольких старших бит, определяющих класс сети, а на основе маски переменной длины, назначаемой поставщиком услуг.



Все адреса имеют общую часть в k старших разрядах – префикс. Оставшиеся n разрядов используются для дополнения неизменяемого префикса переменной частью адреса. Диапазон имеющихся адресов в таком случае составляет 2^n. k+n=32 (размер IPv4 - адреса).

Когда потребитель обращается к поставщику услуг с просьбой о выделении ему некоторого числа адресов, то в имеющемся пуле адресов «вырезается» непрерывная область, в зависимости от требуемого количества адресов. При этом должны быть выполнены следующие условия:

- Количество адресов в выделяемой области должно быть равно степени двойки;
- Начальная граница выделяемого пула адресов должна быть кратна требуемому количеству узлов.

Очевидно, что префикс каждой из показанных на рисунке областей имеет собственную длину – чем меньше количество адресов в данной сети, тем длиннее её префикс.

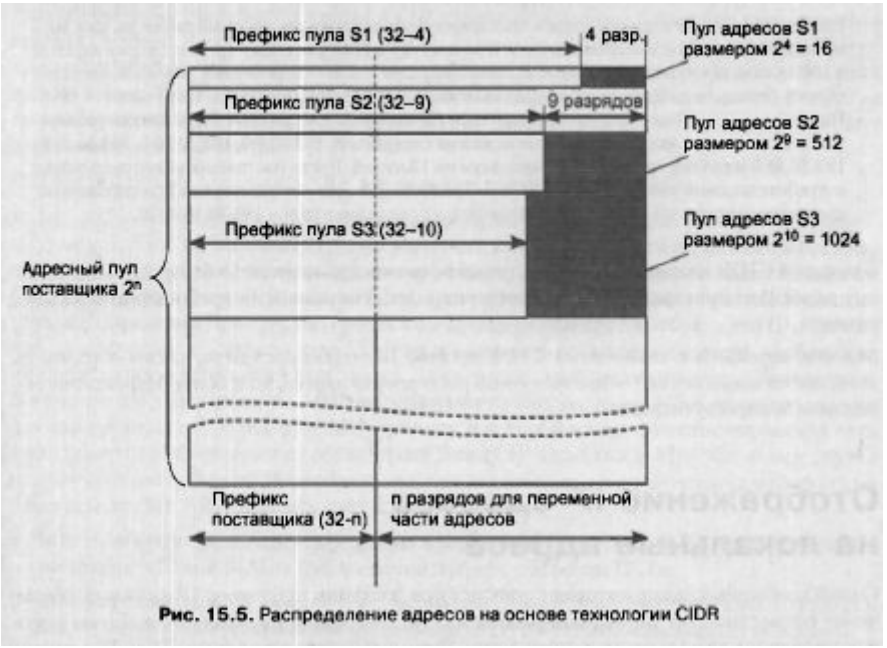


Рис. 15.5. Распределение адресов на основе технологии CIDR

Внедрение технологии CIDR позволяет решить две основные задачи:

- Более экономное расходование адресного пространства
- Уменьшение числа записей в таблицах маршрутизаторов за счёт объединения маршрутов – одна запись в таблице маршрутизации может представлять большое количество сетей.

МАСКА ПОДСЕТИ

Маска подсети (subnet mask) – это 32-битное число, двоичная запись которого содержит в единицы в тех разрядах, которые должны определяться как идентификатор сети. Поскольку идентификатор сети является цельной частью IPv4-адреса, последовательность единиц в маске должна быть также непрерывной.

РАЗБИЕНИЕ НА ПОДСЕТИ

В RFC 950 была описана процедура разбиения сетей на подсети, и в структуру IPv4-адреса был добавлен ещё один уровень – подсеть (subnetwork). Появление ещё одного уровня иерархии не изменило самого IPv4-адреса, он остался 32-разрядным, а часть адреса, отведённая ранее под идентификатор узла, была разделена на 2 части – идентификатор подсети и идентификатор узла.

Разбиение одной крупной сети на несколько мелких позволяет:

- Рационально использовать адресное пространство;
- Повысить безопасность и управляемость сети.

С появлением трёхуровневой иерархии IPv4-адреса потребовались дополнительные методы, которые позволяли бы определить, какая часть IPv4-адреса указывает на идентификатор подсети, а какая – на идентификатор узла. Было предложено использовать битовую маску (bit mask), которая отделяла бы часть адресного пространства идентификаторов узлов от адресного пространства идентификатора подсети. Такая битовая маска называется маской подсети.

Чтобы получить адрес сети, зная IPv4-адрес и маску подсети, необходимо применить к ним операцию логического «И». Другими словами, в тех позициях IPv4-адреса, в которых в маске подсети стоят двоичные единицы, находится идентификатор сети, где двоичные 0 – идентификатор узла.

Для сетей класса A, B, C определены фиксированные маски подсети, которые жёстко определяют количество возможных IPv4-адресов и механизм маршрутизации.

При применении масок подсети сети можно разделять на меньшие по размеру подсети путём расширения сетевой части адреса и уменьшения узловой части. Технология разделения сетей даёт возможность создавать большее число сетей с меньшим количеством узлов в них, что позволяет эффективно использовать адресное пространство.

ТРАНСПОРТНЫЙ УРОВЕНЬ

Транспортный уровень, расположенный между прикладным и сетевым уровнями, представляет собой центральную часть сетевой архитектуры. Он играет важную роль в предоставлении коммуникационных служб непосредственно для прикладных процессов, запущенных на разных хостах.

Протокол транспортного уровня обеспечивает логическое соединение между прикладными процессами, выполняющимися на разных хостах. Логическое соединение с точки зрения приложений выглядит как канал, непосредственно соединяющий процессы, даже если хосты находятся на разных уголках планеты и реальная связь между ними осуществляется с помощью длинной цепи маршрутизаторов и разнообразных линий связи. Процессы прикладного уровня задействуют логическое соединение, предоставляемое транспортным уровнем, для обмена информацией, без учёта деталей физической инфраструктуры, используемой для передачи этих сообщений.

Транспортный уровень (transport layer) обеспечивает приложениям или верхним уровням стека – прикладному, представления и сеансовому – передачу данных с той степенью надёжности, которая им требуется. Модель OSI определяет пять классов транспортного сервиса от низшего класса до высшего класса. Эти виды сервиса отличаются качеством предоставляемых услуг: срочностью, возможностью восстановления прерванной связи, наличием средств мультиплексирования нескольких

соединений между различными прикладными протоколами через общий транспортный протокол, а главное – способностью к обнаружению и исправлению ошибок передачи, таких как искажение, потеря и дублирование пакетов.

СЛУЖБЫ ТРАНСПОРТНОГО УРОВНЯ

Основной задачей UDP и TCP является обеспечение обмена данными между процессами, выполняющимися на конечных системах, при помощи службы обмена данными между конечными системами, предоставляемой протоколом сетевого уровня. Протоколы UDP и TCP также обеспечивают отсутствие искажений данных при передаче, включая в свои заголовки поля обнаружения ошибок. Протокол UDP предоставляет минимальный набор служб транспортного уровня: службы обмена данными между процессами и контроля ошибок. Протокол UDP предоставляет только эти службы! В частности, подобно протоколу IP, UDP является ненадёжной службой и не гарантирует, что данные, отправленные одним процессом, будут доставлены неповреждёнными (и вообще будут доставлены) принимающему процессу.

Протокол TCP, напротив, предоставляет несколько дополнительных служб для приложений. Первая и наиболее важная: служба надёжной передачи данных. Используя управление потоком, порядковые номера, подтверждения и таймеры, протокол TCP гарантирует, что данные будут доставлены от передающего процесса принимающему корректно и в верном порядке. Таким образом, протокол TCP преобразует ненадёжную службу протокола IP между конечными системами в службу надёжной передачи данных между процессами. Протокол TCP также предоставляет службу управления перегрузкой. Можно сказать, что управление перегрузкой в протоколе TCP оберегает любое TCP-соединение от заполнения огромными объёмами трафика, текущего по каналам между маршрутизаторами и взаимодействующими хостами. Протокол TCP призван предоставить каждому соединению равную долю пропускной способности для обхода перегруженного канала. Для этого применяется регулировка скорости, с которой передающая сторона TCP-соединения может отправить трафик в сеть. С другой стороны, в протоколе UDP, трафик неуправляемый. Приложения, использующие протокол UDP, могут осуществлять отправку на любой скорости, и так долго, сколько потребуется.

МЕХАНИЗМЫ ИДЕНТИФИКАЦИИ ДВУХ ПРОЦЕССОВ В СЕТЕВОМ ВЗАИМОДЕЙСТВИИ

Для успешного обмена сообщениями между процессами, выполняющимися на двух различных хостах, необходимо, чтобы они могли идентифицировать друг друга. Идентификация требует наличия следующей информации о процессе:

- Имя или адрес хоста, которому принадлежит процесс;
- Идентификатор процесса внутри хоста.

Адрес хоста: в интернет-приложениях хосты идентифицируются с помощью IP-адресов, представляющих собой 32-разрядное двоичное число, уникальное для каждого хоста сети (говоря точнее для каждого интерфейса, с помощью которого осуществляется подключение хоста к сети).

Идентификация процесса внутри хоста производится с помощью уникального для каждого процесса хоста номера порта. Популярные Интернет-протоколы прикладного уровня имею стандартизированные (хорошо известные) значения номеров портов. Так, процесс, использующий протокол HTTP, получает порт номер 80, а процесс, использующий протокол, - порт номер 25.

ПРОТОКОЛ UDP

Протокол UDP (User Datagram Protocol) является одним из основных протоколов, расположенных непосредственно над IP. Он предоставляет прикладным процессам

транспортные услуги, немногим отличающиеся от услуг протокола IP. Протокол UDP обеспечивает доставку дейтаграмм, но не требует подтверждения их получения. Протокол UDP не требует соединения с удалённым модулем UDP («бессвязный» протокол). По этой причине протокол UDP называют протоколом без установления логического соединения. К заголовку IP-пакета UDP добавляет поля порт отправителя и порт получателя, которые обеспечивают мультиплексирование информации между различными прикладными процессами, а также поля длина UDP-дейтаграммы и контрольная сумма, позволяющие поддерживать целостность данных. Таким образом, если на уровне IP для определения места доставки пакета используется адрес, на уровне UDP – номер порта. Протокол UDP предоставляет ненадёжный сервис, и дейтаграммы могут прийти не по порядку, дублироваться или вовсе исчезнуть без следа. Размер заголовка UDP – 8 байт.

ПРОТОКОЛ TCP

TCP (Transmission Control Protocol, протокол управления передачей) – один из основных протоколов передачи данных интернета, предназначенный для управления передачей данных.

Механизм TCP предоставляет поток данных с предварительной установкой соединения, осуществляет повторный запрос данных в случае потери данных и устраняет дублирование при получении двух копий одного пакета, гарантируя тем самым, в отличие от UDP, целостность передаваемых данных и уведомление отправителя о результатах передачи. Протокол TCP применяется в тех случая, когда требуется гарантированная доставка сообщений. Он использует контрольные суммы пакетов для проверки их целостности и освобождает прикладные процессы от необходимости таймаутов и повторных передач для обеспечения надёжности. Для отслеживания подтверждения доставки в TCP реализуется алгоритм «скользящего» окна. Размер заголовка TCP – 20 байт.

БОРЬБА С ПЕРЕГРУЗКАМИ В ПРОТОКОЛЕ TCP

Когда в какую-либо сеть поступает больше данных, чем она способна обработ

Для управления перегрузкой протокол TCP должен использовать контроль перегрузки конечными системами. Подход протокола TCP заключается в задании ограничений каждому отправителю на скорость, с которой он отсылает трафик по соединению, как функции, зависящей от загруженности сети. Если TCP-отправитель понимает, что загруженность на пути до удалённого хоста невелика, то он увеличивает скорость отправки; если отправитель определяет, что на пути соединения сеть перегружена, то снижает скорость отправки.

Теоретически, также, с перегрузкой можно бороться с помощью закона сохранения пакетов. Идея состоит в том, чтобы не передавать в сеть новые пакеты, пока её не покинут (то есть не будут доставлены) старые. Протокол TCP пытается достичь этой цели с помощью динамического управления размером окна.

{При обнаружении перегрузки должен быть выбран подходящий размер окна. Получатель может указать размер окна, исходя из количества свободного места в буфере. Если отправитель будет иметь в виду размер отведённого ему окна, переполнение буфера у получателя не сможет стать причиной проблемы, однако она все равно может возникнуть из-за перегрузки на каком-либо участке сети между отправителем и получателем.

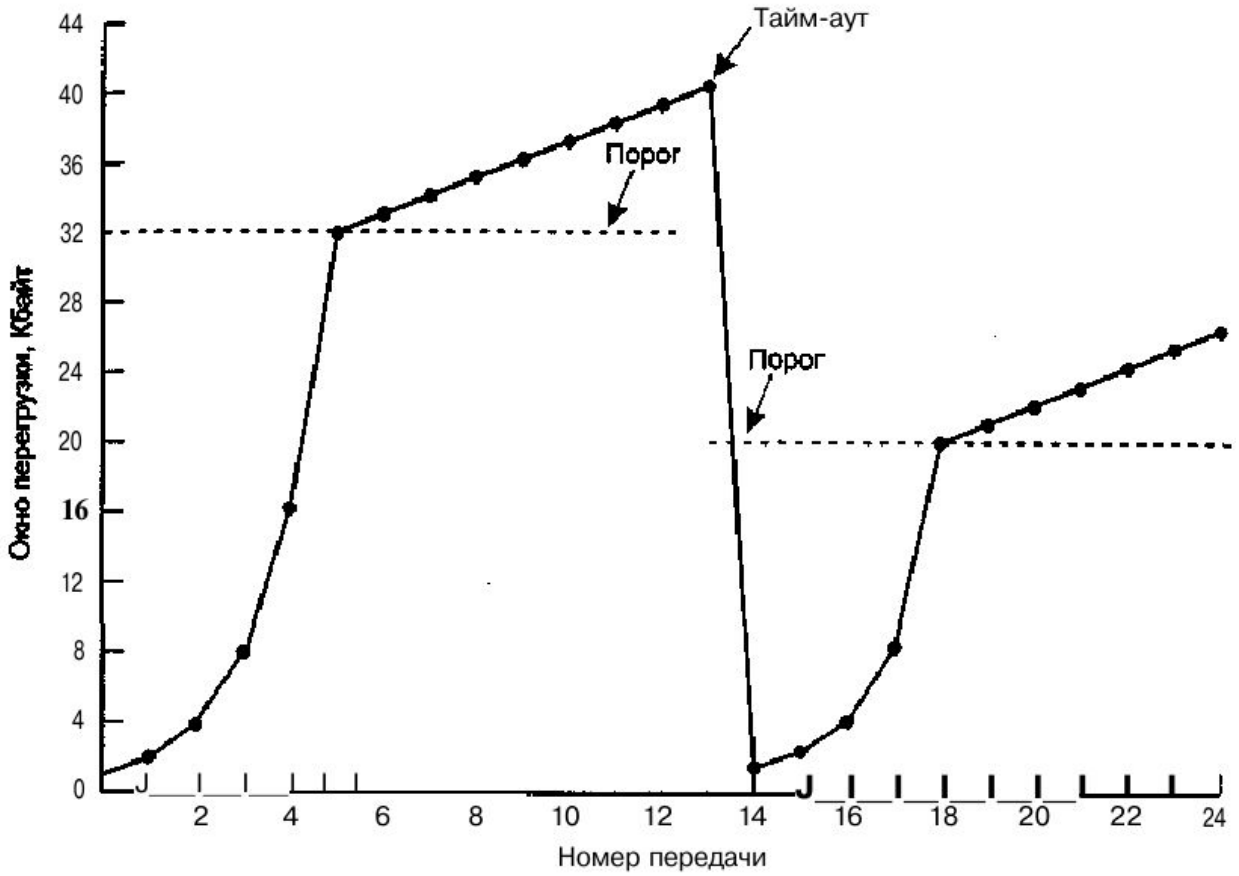
Решение, применяемое в Интернете, состоит в признании существования двух потенциальных проблем: низкой пропускной способности сети и низкой ёмкости получателя — и в раздельном решении обеих проблем. Для этого у каждого отправителя есть два окна: окно, предоставленное получателем, и окно перегрузки. Размер каждого из них соответствует количеству байтов, которое отправитель имеет право передать.

Отправитель руководствуется минимальным из этих двух значений. Например, получатель говорит: «Посылайте 8 Кбайт», но отправитель знает, что если он пошлёт более 4 Кбайт, то в сети образуется затор, поэтому он посылает все же 4 Кбайт. Если же отправитель знает, что сеть способна пропустить и большее количество данных, например 32 Кбайт, он передаст столько, сколько просит получатель (то есть 8 Кбайт).

При установке соединения отправитель устанавливает размер окна перегрузки равным размеру максимального используемого в данном соединении сегмента. Затем он передаёт один максимальный сегмент. Если подтверждение получения этого сегмента прибывает прежде, чем истекает период ожидания, к размеру окна добавляется размер сегмента, то есть размер окна перегрузки удваивается, и посылаются уже два сегмента. В ответ на подтверждение получения каждого из сегментов производится расширение окна перегрузки на величину одного максимального сегмента. Допустим, размер окна равен n сегментам. Если подтверждения для всех сегментов приходят вовремя, окно увеличивается на число байтов, соответствующее n сегментам. По сути, подтверждение каждой последовательности сегментов приводит к удвоению окна перегрузки.

Этот процесс экспоненциального роста продолжается до тех пор, пока не будет достигнут размер окна получателя или не будет выработан признак тайм-аута, сигнализирующий о перегрузке в сети. Например, если пакеты размером 1024, 2048 и 4096 байт дошли до получателя успешно, а в ответ на передачу пакета размером 8192 байта подтверждение не пришло в установленный срок, окно перегрузки устанавливается равным 4096 байтам. Пока размер окна перегрузки остается равным 4096 байтам, более длинные пакеты не посылаются, независимо от размера окна, предоставляемого получателем. Этот алгоритм называется затяжным пуском, или медленным пуском. Однако он не такой уж и медленный (Jacobson, 1988). Он экспоненциальный. Все реализации протокола TCP обязаны его поддерживать.

Рассмотрим теперь механизм борьбы с перегрузкой, применяемый в Интернете. Помимо окон получателя и перегрузки, в качестве третьего параметра в нем используется пороговое значение, которое изначально устанавливается равным 64 Кбайт. Когда возникает ситуация тайм-аута (подтверждение не возвращается в срок), новое значение порога устанавливается равным половине текущего размера окна перегрузки, а окно перегрузки уменьшается до размера одного максимального сегмента. Затем, так же как и в предыдущем случае, используется алгоритм затяжного пуска, позволяющий быстро обнаружить предел пропускной способности сети. Однако на этот раз экспоненциальный рост размера окна останавливается по достижении им порогового значения, после чего окно увеличивается линейно, на один сегмент для каждой следующей передачи. В сущности, предполагается, что можно спокойно урезать вдвое размер окна перегрузки, после чего постепенно наращивать его.}



Максимальный размер сегмента в данном примере равен 1024 байт. Сначала окно перегрузки было установлено равным 64 Кбайт, но затем произошёл тайм-аут, и порог стал равным 32 Кбайт, а окно перегрузки — 1 Кбайт (передача 0). Затем размер окна перегрузки удваивается на каждом шаге, пока не достигает порога (32 Кбайт). Начиная с этого момента, размер окна увеличивается линейно. И в благоприятном варианте развития событий становиться константой. Но у нас передача 13 оказывается несчастливой (как и положено), так как срабатывает тайм-аут. При этом пороговое значение устанавливается равным половине текущего размера окна (40 Кбайт пополам, то есть 20 Кбайт), и опять происходит затяжной пуск. После достижения порогового значения экспоненциальный рост размера окна сменяется линейным. Если тайм-аутов больше не возникает, окно перегрузки может продолжать расти до размера окна получателя. Затем рост прекратится, и размер окна останется постоянным, пока не произойдёт тайм-аут или не изменится размер окна получателя.

ПРИКЛАДНОЙ УРОВЕНЬ

Прикладной уровень (application layer) – это в действительности просто набор разнообразных протоколов, с помощью которых пользователи сети получают доступ к разделяемым ресурсам, таким как файлы, принтеры или гипертекстовые веб-страницы, а также организуют свою совместную работу, например, по протоколу электронной почты. Единица данных, которой оперирует прикладной уровень, обычно называется сообщением.

ПРОТОКОЛЫ ПРИКЛАДНОГО УРОВНЯ

HTTP (HyperText Transfer Protocol – протокол передачи гипертекста) – протокол прикладного уровня передачи данных, с помощью которого браузер взаимодействует с веб-сервером.

FTP (File Transfer Protocol, протокол передачи файлов) – стандартный протокол, предназначенный для передачи файлов.

NFS (Network File System) – протокол сетевого доступа к файловым системам, разработанный в 1984 году.

SMTP (Simple Mail Transfer Protocol) – это широко используемый протокол, предназначенный для передачи электронной почты в сетях.

POP3 (Post Office Protocol Version 3 – протокол почтового отделения, версия 3) – стандартный интернет-протокол прикладного уровня, используемый клиентами электронной почты для получения почты с удалённого сервера по TCP-соединению.

СЕТЕВЫЕ СЛУЖБЫ ПРИКЛАДНОГО УРОВНЯ

Служба имён доменов DNS, суть которой заключается в иерархической схеме имён, основанной на доменах, и распределённой базе данных, реализующих эту схему имён. В первую очередь эта система используется для преобразования имён хостов в IP-адреса.

Электронная почта

Всемирная паутина (WWW, World Wide Web) представляющая собой архитектуру, являющуюся основой для доступа к связанному контенту, находящемуся на миллионах машин по всему Интернету.

Потоковая передача аудио и видео.

Доставка контента

ЭЛЕКТРОННАЯ ПОЧТА

Электронная почта – технология и служба по пересылке и получению электронных сообщений между пользователями компьютерной сети.

Система электронной почты состоит из двух подсистем: пользовательских агентов, позволяющих пользователям читать и отправлять электронную почту, и агентов передачи сообщений, пересылающих сообщения от отправителя к получателю.

Пользовательские агенты представляют собой программы, предоставляющие графический интерфейс или иногда интерфейс, базирующийся на тексте или командах, позволяющий пользователям взаимодействовать с системой электронной почты. В него входят средства написания сообщений и ответов на сообщения, отображения входящих сообщений и организации писем при помощи распределения их по папкам, поиска и удаления.

Агенты передачи сообщений, как правило, являются системными процессами. Они работаю в фоновом режиме на машинах почтовых серверов и всегда должны быть доступными. Они должны автоматически перемещать почтовые сообщения по системе от отправителя к получателю при помощи простого протокола передачи почтовых сообщений SMTP. Это шаг, на котором передаётся сообщение.

Агенты передачи сообщений также используют списки рассылки, которые позволяют доставлять идентичные копии сообщения всем, чьи адреса были включены в список адресов электронной почты.