

Тема. Сетевая модель OSI

План лекции

1. [Сетевая модель OSI](#)
2. [Уровни модели OSI](#)
3. [Понятие "открытая система"](#)
4. [Передача данных по сети и кабелю](#)

Сетевая модель OSI

Работа сети заключается в передаче данных от одного компьютера к другому. В этом процессе можно выделить несколько отдельных задач:

- распознать данные;
- разбить данные на управляемые блоки;
- добавить информацию к каждому блоку, чтобы указать местонахождение данных и указать получателя;
- добавить информацию синхронизации и информацию для проверки ошибок;
- поместить данные в сеть и отправить их по заданному адресу.

Сетевая операционная система при выполнении всех этих задач следует строгому набору процедур.

Опр. Протокол – процедуры (правила поведения) регламентирующие каждую сетевую операцию.

Стандартные протоколы позволяют программному и аппаратному обеспечению различных производителей нормально взаимодействовать. Существуют два главных набора стандартов: модель OSI и ее модификация Project 802. Чтобы изучить техническую сторону функционирования сетей, необходимо иметь четкое представление об этих моделях.

В 1984 году ISO (International Standards Organization) выпустила эталонную модель взаимодействия открытых систем OSI (Open System Interconnection). Она стала международным стандартом: именно ее спецификации используют производители при разработке сетевых продуктов, именно ее придерживаются при построении сетей. Модель OSI определяет различные уровни взаимодействия систем, дает им стандартные имена и указывает, какие функции должен выполнять каждый уровень. В модели OSI средства взаимодействия делятся на семь уровней: прикладной, представительный, сеансовый, транспортный, сетевой, канальный и физический.

Каждый уровень имеет дело с одним определенным аспектом взаимодействия сетевых устройств. Модель OSI описывает только системные средства взаимодействия, реализуемые операционной системой, системными утилитами, системными аппаратными средствами. Модель не включает средства взаимодействия приложений конечных пользователей. Свои собственные протоколы взаимодействия приложения реализуют, обращаясь к системным средствам. Поэтому необходимо различать уровень взаимодействия приложений и прикладной уровень.

Процесс формирования начинается на Прикладном уровне. Итак, пусть приложение обращается с запросом к прикладному уровню, например к файловой службе. На основании этого запроса программное обеспечение прикладного уровня формирует сообщение стандартного формата. Обычное сообщение состоит из заголовка и поля данных. Заголовок содержит служебную информацию, которую необходимо передать через сеть прикладному уровню машины-адресата, чтобы сообщить ему, какую работу надо выполнить. После формирования сообщения прикладной уровень направляет его вниз по стеку представительному уровню. Протокол представительного уровня на основании информации, полученной из заголовка прикладного уровня, выполняет требуемые действия и добавляет к сообщению собственную служебную информацию - заголовок представительного уровня. Полученное в результате сообщение передается вниз сеансовому уровню, который в свою очередь добавляет свой заголовок, и т. д. Наконец, сообщение достигает нижнего, физического уровня, который собственно и передает его по линиям связи машине-адресату. К этому моменту сообщение "обрастает" заголовками всех уровней. Когда сообщение по сети поступает на машину-адресат, оно принимается ее физическим уровнем и последовательно перемещается вверх с уровня на уровень. Каждый уровень анализирует и обрабатывает заголовок своего уровня, выполняя соответствующие данному уровню функции, а затем удаляет этот заголовок и передает сообщение вышележащему уровню.

Уровни модели OSI

Физический уровень (Physical layer) Физический уровень имеет дело с передачей битов по физическим каналам связи, таким, например, как коаксиальный кабель, витая пара, оптоволоконный кабель или цифровой территориальный канал. К этому уровню имеют отношение характеристики

физических сред передачи данных, такие как полоса пропускания, помехозащищенность, волновое сопротивление и другие. На этом же уровне определяются характеристики электрических сигналов, передающих дискретную информацию, например, крутизна фронтов импульсов, уровни напряжения или тока передаваемого сигнала, тип кодирования, скорость передачи сигналов. Кроме этого, здесь стандартизируются типы разъемов и назначение каждого контакта.

Канальный уровень (Data Link layer) На физическом уровне просто пересылаются биты. При этом не учитывается, что в некоторых сетях, в которых линии связи используются (разделяются) попеременно несколькими парами взаимодействующих компьютеров, физическая среда передачи может быть занята. Поэтому одной из задач канального уровня является проверка доступности среды передачи. Другой задачей канального уровня является 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, обрабатывая все байты кадра определенным способом и добавляя контрольную сумму к кадру. Когда кадр приходит по сети, получатель снова вычисляет контрольную сумму полученных данных и сравнивает результат с контрольной суммой из кадра. Если они совпадают, кадр считается правильным и принимается. Если же контрольные суммы не совпадают, то фиксируется ошибка. Канальный уровень может не только обнаруживать ошибки, но и исправлять их за счет повторной передачи поврежденных кадров. Тем не менее, 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.

Сетевой уровень (Network layer) Сетевой уровень служит для образования единой транспортной системы, объединяющей несколько сетей. Проблема выбора наилучшего пути называется маршрутизацией, и ее решение является одной из главных задач сетевого уровня. Сетевой уровень решает также задачи согласования разных технологий, упрощения адресации в крупных сетях и создания надежных и гибких барьеров на пути нежелательного трафика между сетями. Сообщения сетевого уровня принято называть пакетами (packets).

Транспортный уровень (Transport layer) На пути от отправителя к получателю пакеты могут быть искажены или утеряны. Транспортный уровень гарантирует доставку пакетов без ошибок, в той же последовательности, без потерь и дублирования. На этом уровне сообщения переупаковываются: длинные разбиваются на несколько пакетов, а короткие объединяются в один. Это увеличивает эффективность передачи пакетов по сети. На Транспортном уровне компьютер-получатель сообщения распаковываются, восстанавливаются в первоначальном виде, и обычно посылается сигнал подтверждения приема. Транспортный уровень управляет потоком, проверяет ошибки и участвует в решении проблем, связанных с отправкой и получением пакетов.

Сеансовый уровень (Session layer) Сеансовый уровень обеспечивает управление диалогом: фиксирует, какая из сторон является активной в настоящий момент, предоставляет средства синхронизации. Последние позволяют вставлять контрольные точки в длинные передачи, чтобы в случае отказа можно было вернуться назад к последней контрольной точке, а не начинать все с начала. На практике немногие приложения используют сеансовый уровень, и он редко реализуется в виде отдельных протоколов, хотя функции этого уровня часто объединяют с функциями прикладного уровня и реализуют в одном протоколе.

Представительский уровень (Presentation layer) Представительский уровень имеет дело с формой представления передаваемой по сети информации, не меняя при этом ее содержания. За счет уровня представления информация, передаваемая прикладным уровнем одной системы, всегда понятна прикладному уровню другой системы. На этом уровне может выполняться шифрование и дешифрование данных, благодаря которому секретность обмена данными обеспечивается сразу для всех прикладных служб.

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

Понятие "открытая система"

Модель OSI, как это следует из ее названия (Open System Interconnection), описывает взаимосвязи открытых систем. Что же такое открытая система?

В широком смысле открытой системой может быть названа любая система (компьютер, вычислительная сеть, ОС, программный пакет, другие аппаратные и программные продукты), которая построена в соответствии с открытыми спецификациями.

Напомним, что под термином "спецификация" (в вычислительной технике) понимают формализованное описание аппаратных или программных компонентов, способов их функционирования, взаимодействия с другими компонентами, условий эксплуатации, ограничений и особых характеристик. Понятно, что не всякая спецификация является стандартом. В свою очередь, под открытыми спецификациями понимаются опубликованные, общедоступные спецификации, соответствующие стандартам и принятые в результате достижения согласия после всестороннего обсуждения всеми заинтересованными сторонами.

Использование при разработке систем открытых спецификаций позволяет третьим сторонам разрабатывать для этих систем различные аппаратные или программные средства расширения и модификации, а также создавать программно-аппаратные комплексы из продуктов разных производителей.

Для реальных систем полная открытость является недостижимым идеалом. Как правило, даже в системах, называемых открытыми, этому определению соответствуют лишь некоторые части, поддерживающие внешние интерфейсы. Например, открытость семейства операционных систем Unix заключается, кроме всего прочего, в наличии стандартизованного программного интерфейса между ядром и приложениями, что позволяет легко переносить приложения из среды одной версии Unix в среду другой версии. Еще одним примером частичной открытости является применение в достаточно закрытой операционной системе Novell NetWare открытого интерфейса Open Driver Interface (ODI) для включения в систему драйверов сетевых адаптеров независимых производителей. Чем больше открытых спецификаций использовано при разработке системы, тем более открытой она является.

Модель OSI касается только одного аспекта открытости, а именно открытости средств взаимодействия устройств, связанных в вычислительную сеть. Здесь под открытой системой понимается сетевое устройство, готовое взаимодействовать с другими сетевыми устройствами с использованием стандартных правил, определяющих формат, содержание и значение принимаемых и отправляемых сообщений.

Если две сети построены с соблюдением принципов открытости, то это дает следующие преимущества:

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

Ярким примером открытой системы является международная сеть Internet. Эта сеть развивалась в полном соответствии с требованиями, предъявляемыми к открытым системам. В разработке ее стандартов принимали участие тысячи специалистов-пользователей этой сети из различных университетов, научных организаций и фирм-производителей вычислительной аппаратуры и программного обеспечения, работающих в разных странах. Само название стандартов, определяющих работу сети Internet - Request For Comments (RFC), что можно перевести как "запрос на комментарии", - показывает гласный и открытый характер принимаемых стандартов. В результате сеть Internet сумела объединить в себе самое разнообразное оборудование и программное обеспечение огромного числа сетей, разбросанных по всему миру.

Модель OSI – это основа функционирования сетей. Поэтому, если Вы поймете, как взаимодействуют при сетевых соединениях различные уровни модели OSI, Вы сможете разобраться в принципах действиях реальных сетевых функций.

Передача данных по сети и кабелю

Модель OSI – это основа функционирования сетей. Поэтому, если Вы поймете, как взаимодействуют при сетевых соединениях различные уровни модели OSI, Вы сможете разобраться в

принципах действия реальных сетевых функций. Начнем изучение с драйверов – программ которые обеспечивают функционирование аппаратных сетевых компонентов.

Опр. Драйверы (driver)– это программное обеспечение, позволяющее компьютеру работать с различными устройствами.

Драйверы существуют почти для каждого типа устройства компьютера и периферии. Хорошим примером использования драйверов может служить драйвер принтера. Принтеры производятся большим количеством фирм и обладают различными функциями и особенностями. Производители компьютеров просто не в состоянии оснастить свои компьютеры программным обеспечением для работы с каждым типом принтера. Вместо этого производители принтеров создают драйверы для своих принтеров, и чтобы Ваш компьютер мог посылать документы на принтер, сначала надо загрузить драйвер этого принтера, который обеспечивает взаимодействие компьютера с этим устройством.

Драйверы платы сетевого адаптера располагаются на подуровне Управления доступом к среде (Канальный уровень модели OSI). Подуровень Управления доступом к среде отвечает за совместный доступ плат сетевого адаптера к Физическому уровню, т.е. драйверы обеспечивают прямую связь между компьютером и самой платой. Это, в свою очередь, связывает компьютер с сетью.

Сейчас, когда Ваш компьютер и периферийные устройства могут взаимодействовать друг с другом, сеть готова перемещать данные между компьютерами. Однако большинство файлов слишком велико для передачи их по кабелю целиком. Во-первых, такой блок заполняет кабель и «связывает» работу всей сети, во-вторых, возникновение ошибок приведет к повторной передаче всего блока. Чтобы быстро и легко передавать по сети данные, надо разбить их на небольшие управляемые блоки.

Опр. Пакеты – это управляемые блоки данных, на которые разбивается файл.

Пакет - это основная единица информации в компьютерных сетях. При разбиении данных на пакеты скорость их передачи возрастает настолько, что каждый компьютер в сети получает возможность принимать и передавать данные практически одновременно с остальными компьютерами.

Компонентами пакеты являются: заголовок, данные, трейлер. Заголовок включает в себя:

- сигнал, «говорящий» о том, что передается пакет;
- адрес источника;
- адрес местоназначения;
- информацию, синхронизирующую передачу.

Трейлер, чаще всего содержит информацию для проверки ошибок, называемую циклическим избыточным кодом (Cyclical Redundancy Check, CRC). CRC – это число, получаемое в результате математических преобразований над пакетом и исходной информацией. Когда пакет достигает местоназначения, эти преобразования повторяются. Если результат совпадает с CRC – пакет принят без ошибок.

Компьютеры, между которыми установлена связь, чем-то напоминают международные организации. Существует множество языков и способов общения, которыми эти организации могут воспользоваться. Важно лишь помнить: все участники коммуникации, чтобы достичь взаимопонимания, должны говорить на одном языке и следовать единым правилам, или протоколам.

Опр. Протоколы – это набор правил и процедур, регулирующих порядок осуществления некоторой связи.

Запомните три основных момента, касающихся протоколов:

1. Существует множество протоколов. Каждый протокол имеет различные цели, выполняет различные задачи, обладает своими преимуществами и ограничениями.
2. Протоколы работают на разных уровнях модели OSI. Функции протокола определяются уровнем, на котором он работает.
3. Несколько протоколов могут работать совместно. Это так называемый стек, или набор, протоколов.

Протоколы в сетевой среде определяют правила и процедуры передачи данных. Передача данных по сети состоит из ряда шагов, которые должны выполняться в неизменном порядке. Компьютер-отправитель и компьютер-получатель используют протоколы для выполнения следующих процедур:

- разбиение данных на пакеты;
- добавление к пакету адресной информации;
- подготовка пакетов к передаче;
- прием пакетов, передаваемых по кабелю;
- копирование данных из пакета для сборки исходных блоков данных;
- передача этих восстановленных блоков в компьютер.

Установка правильных протоколов еще не гарантия корректной работы сети. Для передачи данных по кабелю компьютер использует различные методы доступа.

Опр. Метод доступа – набор правил, которые определяют, как компьютер должен отправлять и принимать данные по сетевому кабелю.

В сети несколько компьютеров должны иметь совместный доступ к кабелю. Однако, если два компьютера попытаются одновременно передавать данные, их пакеты «столкнутся» друг с другом и будут испорчены. Это так называемая коллизия .

Методы доступа служат для предотвращения одновременного доступа к кабелю нескольких компьютеров. Основные методы доступа:

- ❖ множественный доступ с контролем несущей с обнаружением коллизий;
- ❖ множественный доступ с контролем несущей с предотвращением коллизий;
- ❖ доступ с передачей маркера;
- ❖ доступ по приоритету запроса.

1. CSMA/CD (Carrier-Sense Multiple Access with Collision Detection) - множественный доступ с контролем несущей с обнаружением коллизии используется в топологиях шина и звезда. Станции прослушивают канал передачи данных. Чаще всего сразу несколько компьютеров в сети «хотят» передать данные (отсюда множественный), компьютеры как бы «прослушивают» кабель (отсюда – контроль несущей). Передавая данные, компьютеры «прослушивают» кабель, чтобы, обнаружив коллизии, некоторое время переждать, а затем возобновить передачу, отсюда – обнаружение коллизий.

В то же время способность обнаружить коллизии – причина, которая ограничивает область действия самого CSMA/CD. Из-за ослабления сигнала при расстояниях свыше 2,5 км механизм обнаружения коллизий не эффективен. CSMA/CD известен как состязательный метод, поскольку сетевые компьютеры «состязаются» (конкурируют) между собой за право передавать данные. Он кажется достаточно громоздким, но современные реализации CSMA/CD настолько быстры, что пользователи даже не задумываются над тем, что применяют состязательный метод доступа. Но чем больше компьютеров в сети, тем интенсивнее сетевой трафик. Число коллизий возрастает, а это приводит к замедлению сети. Поэтому в некоторых ситуациях метод CSMA/CD может оказаться не достаточно быстрым.

После каждой коллизии обоим компьютерам приходится возобновлять передачу. Если сеть очень загружена, повторные попытки опять могут привести к коллизии, но уже с другими компьютерами. Теперь уже четыре компьютера будут возобновлять передачу. Такое лавинообразное нарастание повторных передач может парализовать работу всей сети.

2. CSMA/CA (Carrier-Sense Multiple Access with Collision Against) - множественный доступ с контролем несущей с предотвращением коллизий. Используя метод CSMA/CA каждый компьютер перед передачей данных в сеть сигнализирует о своем намерении, поэтому остальные компьютеры «узнают» о готовящейся передаче и могут избежать коллизий.

Однако широковещательное оповещение увеличивает общий трафик сети и уменьшает ее пропускную способность, поэтому сеть с методом CSMA/CA работает медленнее, чем с CSMA/CD.

3. Метод доступа с передачей маркера. Суть в том, что пакет особого типа (маркер) циркулирует по кольцу. Компьютер ждет маркер и захватывает его, наполняет информацией и отправляет адресату. В это время другие компьютеры ожидают освобождения маркера. Так как в каждый момент времени только один компьютер будет использовать маркер, то в сети не возникнет ни состязания, ни коллизий, ни временных пауз.

4. Доступ по приоритету запроса основан на том, сеть строится только из концентраторов и узлов. Концентраторы управляют доступом к кабелю, последовательно опрашивая все узлы в сети и выявляя

запросы на передачу. Концентратор должен знать все адреса, связи и узлы и проверять их работоспособность. Получив одновременно два запроса, концентратор вначале отдаст предпочтение запросу с более высоким приоритетом. Если запросы имеют одинаковый приоритет, они будут обслужены в произвольном порядке. Таким образом, при доступе по приоритету запроса связь осуществляется только между компьютером-отправителем, концентратором и компьютером-получателем. Передачей данных централизованно управляет концентратор, причем на все остальные компьютеры в сети он не вещает.

Тема. Создание больших сетей

План лекции

1. [Модемы](#)
2. [Репитер](#)
3. [Мост](#)
4. [Маршрутизатор](#)
5. [Шлюз](#)

Модемы

Когда компании растут, растут и их сети. В целом локальные сети имеют свойство перерастать начальные проекты. Это становится очевидным, когда:

- трафик сети достиг предела пропускной способности;
- увеличилось время ожидания очередной обработки заданий на печать;
- увеличилось время отклика интенсивно работающих с сетью приложений, таких, как база данных.

В работе каждого администратора рано или поздно наступает момент, когда он должен увеличить размер сети или улучшить ее производительность. Сеть не могут расширять за счет простого добавления новых компьютеров и прокладки дополнительного кабеля. Любая топология или архитектура имеет свои ограничения. Тем не менее, существуют устройства, назначение которых – увеличить размер сети в действующей среде.

1. **Опр. Модем** – это устройство, которое позволяет компьютерам обмениваться данными по телефонной линии.

Функция модема – передающий модем модулирует цифровой сигнал в аналоговый, а принимающий демодулирует аналоговый в цифровой. Существует внутренний модем и внешний модем.

Промышленные стандарты существуют практически для каждой области сетевых технологий, и модем не является исключением. Стандарты обеспечивают взаимодействие модемов от разных производителей. В начале 1980-х годов компания Hayes Microcomputer Product, Inc разработала модем, который получил название Hayes Smartmodem. Он превратился в стандарт, на который стали ориентировать другие модемы. Изначально скорость модема измерялась в битах в секунду или в единицах, называемых «бод». Бод относится к частоте осцилляции звуковой волны, переносящая биты данных по телефонной линии. Свое название эта единица получила от имени французского офицера-связиста Ж.Бода. В начале 80-х годов скорость в бодах равнялась скорости передачи модемов. Затем инженеры связи разработали методы сжатия и кодирования данных. В результате каждая модуляция звука могла перенести больше одного бита информации. Это означает, что скорость в битах в секунду может быть выше скорости в бодах.

Существуют различные типы модемов, поскольку существуют различные типы среды передачи, для которых требуются разные методы передачи данных. Эти среды грубо можно поделить на два типа, взяв как критерий синхронизацию связи.

Асинхронный (последовательный) модем, поддерживает асинхронную связь, используемую в стандартных телефонных линиях. Каждый символ раскладывается в последовательность битов. каждая из этих последовательностей отделяется от других стартовым битом и стоповым битом. передающее и принимающее устройства должны согласовывать комбинацию стартовых и стоповых битов. Связь этого типа не синхронизируется, следовательно, нет специального устройства для синхронизации. Данные просто передаются и принимаются, 25% трафика составляет управляющая информация.

Синхронный модем основан на схеме синхронизации, согласованной между двумя устройствами. Ее цель – выделить биты из группы при передаче их блоками, которые называются кадрами. Передача завершается в конце одного кадра и начинается вновь на следующем кадре. Этот метод более эффективен, чем асинхронная передача.

Из-за высокой стоимости и сложности синхронные модемы для домашних условий, как правило, не предлагаются.

Репитер

2. **Опр. Репитер** – устройство, усиливающее электрический сигнал.

Репитер работает на Физическом уровне модели OSI, восстанавливая сигнал и передавая его в другие сегменты. Репитер не выполняет функции преобразования и фильтрации. Чтобы репитер работал, оба сегмента, им соединяемые, должны иметь одинаковый метод доступа. Это самый дешевый способ расширения сети. Применение репитеров оправдано, когда при расширении сети необходимо преодолеть ограничения по длине сегмента или по количеству узлов, причем ни один из сегментов не генерирует повышенный трафик, а стоимость – главный фактор.

Мост

3. **Опр. Мост** – устройство соединения сегментов или сетей. Задачи моста:

- увеличить размерность сети;
- увеличить максимальное количество компьютеров в сети;
- устранить узкие места в сети;
- соединить разнородные сегменты сети;
- соединить разнородные физические носители.

Мосты работают на Канальном уровне модели OSI, поэтому им недоступна информация, содержащаяся на более высоких уровнях этой модели. Мосты допускают использование в сети всех протоколов, не отличая при этом один протокол от другого. Поскольку любые протоколы могут работать через мосты, каждый компьютер должен определять, с какими протоколами он работает. Мосты уровня Управления доступом к среде выполняют следующие действия:

- «слушает» весь трафик;
- проверяет адреса источника и получателя каждого пакета;
- строит таблицу маршрутизации;
- передает пакеты.

Работа моста основана на принципе, согласно которому каждый узел сети имеет собственный адрес. В начале работы таблица маршрутизации моста пуста. Затем, когда узлы передают пакеты, адрес источника копируется в таблицу маршрутизации. Имея эти данные, мост изучает расположение компьютеров в сегментах сети.

Принимая пакет, мост ищет адрес источника в таблице маршрутизации.

- ✓ Если адрес источника не найден, он добавляет его в таблицу. Затем мост сравнивает адреса назначения с базой данных таблицы маршрутизации.
- ✓ Если адрес получателя есть в таблице маршрутизации и адресат находится в одном сегменте с источником, пакет отбрасывается. Эта фильтрация уменьшает сетевой трафик и изолирует сегменты сети.
- ✓ Если адрес получателя есть в таблице маршрутизации, а адресат и источник находятся в разных сегментах, мост передает пакет адресату через соответствующий порт. Если адрес получателя нет в таблице маршрутизации, мост передает пакет во все свои порты, исключая тот, через который пакет был принят.

Если для расширения сети Вы решили использовать мосты, учитывайте следующие факты:

- мосты обладают всеми возможностями репитера;
- соединяют два сегмента и восстанавливают сигналы на уровне пакетов;
- функционируют на Канальном уровне модели OSI;
- не подходят для распределенных сетей со скоростями передачи менее 56 Кбит/с;
- не могут одновременно использовать несколько маршрутов;
- пропускают все широковещательные сообщения, допуская перегрузку сети;
- считывают адрес источника и получателя каждого пакета;
- пропускают пакеты с неизвестным адресом получателя;

Основное назначение мостов:

- соединить два сегмента для увеличения длины сети или количества узлов в ней;
- уменьшить трафик за счет сегментации сети;
- соединить разнородные сети.

Маршрутизатор

4. **Опр. Маршрутизатор** – это устройство, которое знает адрес каждого сегмента, определяет наилучший маршрут и фильтрует широковещательные сообщения. Он работает на Сетевом уровне модели OSI. Это означает, что они могут переадресовывать и маршрутизировать пакеты через множество сетей.

Таблица маршрутизации содержит сетевые адреса. Для каждого протокола, используемого в сети, строится своя таблица. Она включает следующую информацию:

- все известные сетевые адреса;
- способы связи с другими сетями;
- возможные пути между маршрутизаторами;
- стоимость передачи данных по этим путям.

Маршрутизатор выбирает наилучший маршрут для данных, сравнивая стоимость и доступность различных вариантов.

С маршрутизаторами работают не все протоколы. Протоколы, работающие с маршрутизатором, называются маршрутизируемые. К ним относятся: DECnet, IP, IPX, OSI, XNS. К немаршрутизируемым протоколам относятся: LAT, NetBEUI.

Маршрутизатор может «прослушивать» сеть и определять, какие ее части сильнее загружены. Он устанавливает также количество транзитов между сегментами сети. Используя эту информацию, маршрутизатор выбирает маршрут передачи данных. Если один путь перегружен, он укажет альтернативный.

Различие между мостами и маршрутизаторами в том, что они работают на разных уровнях модели OSI. У маршрутизатора больше информации, он распознает не только адрес, но и тип протокола. Мост может распознать только один путь между сетями. Маршрутизатор среди нескольких возможных путей определяет самый лучший на данный момент.

Шлюз

5. **Опр. Шлюз** – устройство для объединения информационных сетей, использующих различные протоколы. Шлюзы работают на прикладном уровне модели OSI.

Шлюзы обеспечивают связь между различными архитектурами и средами. Они переупаковывают и преобразуют данные, передаваемые из одной среды в другую, чтобы каждая среда могла понимать данные других сред. В частности, шлюз переупаковывает информацию в соответствии с требованиями системы назначения; изменяет формат сообщения, чтобы прикладная программа на принимающей стороне могла распознать данные. Например, шлюзы электронной почты принимают сообщения в одном формате, транслируют его и пересылают в другом формате.

Шлюз связывает две системы, которые используют разные:

- коммуникационные протоколы;
- структуры и форматы данных;
- языки;

- архитектуры.

Шлюзы создаются для выполнения конкретного типа задач, т.е. для конкретного типа преобразования данных.

Обработывая данные, шлюз выполняет следующие операции:

- извлекает данные из входящих пакетов, пропуская их снизу вверх через полный стек протоколов из передающей сети;
- заново упаковывает полученные данные, пропуская их сверху вниз через стек протоколов сети назначения.

Обычно роль шлюзов в сети выполняют выделенные серверы, поэтому это достаточно дорогое устройство для расширения сети.