

### 38. Компьютерные сети. Сетевой уровень.

#### Маршрутизация в дейтаграммной подсети.

#### Маршрутизация в подсети виртуального канала.

#### Алгоритмы маршрутизации. Борьба с перегрузкой.

Компьютерная сеть – система, обеспечивающая обмен данными между вычислительными устройствами (компьютеры, серверы, маршрутизаторы и другое оборудование).

#### Сетевой уровень.

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 (то есть о множестве всех маршрутизаторов и связей)
  - выбирать нужный путь по этой сети, даже если она достаточно крупная,
  - контролировать нагрузку на маршрутизаторы и линии связи (следить, чтобы она была более равномерной),
  - решать проблемы, связанные с различиями в сетях, если источник и приемник находятся в различных сетях.

#### Маршрутизация в дейтаграммной подсети.

Дейтаграмма - блок информации, передаваемый протоколом без предварительного установления соединения и создания виртуального канала.

Процесс P1 хочет послать длинное сообщение для P2.

P1 передает свое послание транспортному уровню, сообщает ему о том, что доставить данные надо процессу P2, выполняющемуся на хосте H2.

Заголовок транспортного уровня вставляется в начало сообщения, и в таком виде сообщение передается на сетевой уровень.

Если сообщение длиннее максимального размера пакета, то сетевой уровень разбивает его на несколько частей и посылает их все поочередно на маршрутизатор А с использованием какого-нибудь двухточечного соединения (PPP).

Каждый маршрутизатор имеет свою внутреннюю таблицу, по которой он определяет дальнейший путь пакета при каждом из возможных адресов назначения. Каждая запись таблицы состоит из двух полей: пункт назначения (адресат) и выходящая линия для данного адресата. В маршрутизаторе А пакеты, поступившие на вход, кратковременно сохраняются для верификации контрольной суммы. Затем в соответствии с таблицей А (допустим, есть две выходящих линии: на маршрутизаторы В и С) каждый пакет пересылается по исходящему соединению на маршрутизатор С или В с использованием нового кадра. И так далее до хоста H2.

Для каждого пакета маршрут прорабатывается индивидуально. Если где-то на линии при пересылке пакетов возникает затор, то следующий пакет может быть отправлен по другому пути.

Яркий пример сетевого сервиса без установления соединения – протокол IP (межсетевой протокол).

#### Маршрутизация в подсети виртуального канала.

При использовании сервиса, ориентированного на соединение, весь путь от маршрутизатора-отправителя до маршрутизатора-получателя должен быть установлен до начала передачи каких-либо пакетов данных. Такое соединение называется виртуальным каналом.

Идея виртуальных каналов состоит в предотвращении выбора своего маршрута для каждого пакета. Вместо этого маршрут от отправляющей до получающей машины выбирается в процессе установления соединения и хранится в специальных таблицах, встроенных в маршрутизаторы. Один и тот же маршрут используется для всего трафика, проходящего через данное соединение. Когда соединение разрывается, виртуальный канал также прекращает свое существование. При использовании сервиса, ориентированного на установление соединения, каждый пакет включает в себя идентификатор виртуального канала.

*Хост - это определенный компьютер или сервер, подключенный к локальной или глобальной сети.*

Хост H1 установил соединение с хостом H2. Например, через маршрутизаторы А, С и Е.

Это соединение запоминается и становится первой записью во всех таблицах маршрутизации. Так, первая строчка таблицы маршрутизатора А говорит о том, что если пакет с идентификатором соединения 1 пришел с хоста Н1, то его нужно отправить на С с идентификатором соединения 1. Точно так же первая запись С направляет пакет на Е все с тем же идентификатором соединения 1.

Если хост Н3 захочет установить соединение с Н2 через маршрутизатор А, то он выберет идентификатор соединения 1 (у него просто нет выбора, поскольку это на данный момент единственное существующее соединение) и просит сеть установить виртуальный канал. Таким образом, в таблице появляется вторая запись.

А присвоит новый идентификатор соединения исходящему трафику и тем самым создаст второе соединение.

Сравнение виртуальных каналов и дейтаграмм.

Проблема	Дейтаграммы	Виртуальные каналы
Установка канала	Не требуется	Требуется
Адресация	Каждый пакет содержит полный адрес отправителя и получателя	Каждый пакет содержит короткий номер виртуального канала
Информация о состоянии	Маршрутизаторы не содержат информации о состоянии	Каждый виртуальный канал требует места в таблице маршрутизатора
Маршрутизация	Маршрут каждого пакета выбирается независимо	Маршрут выбирается при установке виртуального канала. Каждый пакет следует по этому маршруту
Эффект от выхода из строя маршрутизатора	Никакого, кроме потерянных пакетов	Все виртуальные каналы, проходившие через отказавший маршрутизатор, прекращают существование
Обеспечение качества обслуживания	Трудно реализовать	Легко реализуется при наличии достаточного количества ресурсов для каждого виртуального канала
Борьба с перегрузкой	Трудно реализовать	Легко реализуется при наличии достаточного количества ресурсов для каждого виртуального канала

Алгоритмы маршрутизации.

Алгоритм маршрутизации реализуется той частью программного обеспечения сетевого уровня, которая отвечает за выбор выходной линии для отправки пришедшего пакета.

Алгоритмы выбора маршрута можно разбить на два основных класса: неадаптивные и адаптивные.

Неадаптивные алгоритмы (статическая маршрутизация) 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. Используются, когда выбор маршрута очевиден.

Адаптивные алгоритмы (динамическая маршрутизация) изменяют решение о выборе маршрутов при изменении топологии и также иногда в зависимости от загруженности линий. Отличаются источниками получения информации, моментами изменения маршрутов и данными, используемыми для оптимизации.

Алгоритм нахождения кратчайшего пути (Дейкстры) (статический).

Данный алгоритм находит кратчайшие пути между отправителем и всеми возможными адресами назначения в данной сети.

Топология сети представляется в виде неориентированного графа с указанными для каждого ребра значениями метрики (например, расстояния между двумя соседними узлами). Изначально путь неизвестен, поэтому все вершины графа получают метки с бесконечным значением расстояния до отправителя.

Пусть требуется найти кратчайший путь от А до D.

1. Сначала узел А получает постоянную метку и делается текущим узлом.
2. Затем мы просматриваем все соседние узлы (с которыми вершина А соединена ребром) и отмечаем расстояние до А.
3. После того как все соседние узлы перебраны, узел с наименьшей меткой выбирается в качестве текущего, при этом он получает постоянную метку. Пусть это будет узел В.
4. Теперь мы просматриваем все соседние узлы, и если сумма расстояний от В до А и от данного узла до В меньше значения метки этого узла, то он получает новую метку, причем мы указываем, что путь в А лежит через В, чтобы после завершения процедуры можно было восстановить маршрут.
5. После перебора всех соседних узлов мы вновь ищем по всему графу вершину с наименьшей временной меткой и выбираем этот узел в качестве текущего, при этом он получает постоянную метку и т. д.

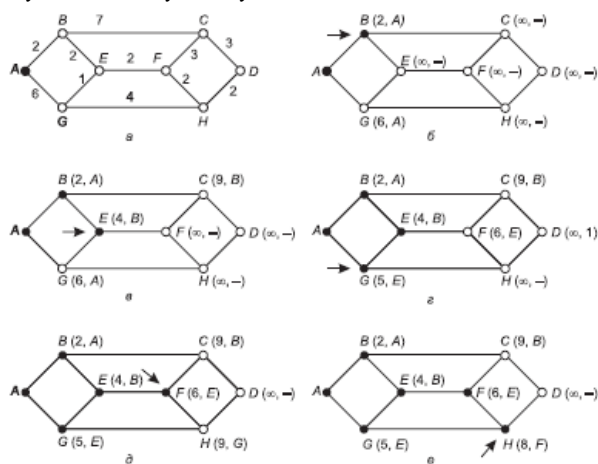


Рис. 5.6. Первые шесть шагов вычисления кратчайшего пути от А к D. Стрелка указывает на рабочий узел

Алгоритм заливки (статический).

Идея: каждый приходящий пакет посылается на все исходящие линии кроме той, откуда он пришел.

Недостаток: порождается большое количество дублированных пакетов которые гуляют по сети.

Для ограничения количества тиражируемых пакетов используются счетчики, которые помещаются в заголовок пакета и уменьшаются при прохождении каждого маршрутизатора. Если счетчик обнуляется, то такой пакет удаляется.

В большинстве случаев использование алгоритма заливки для отправки пакетов является непрактичным, но тем не менее иногда он оказывается очень полезным.

Во-первых, он гарантированно доставляет пакет в каждый узел сети. Если пакет нужно доставить в одно конкретное место, этот метод может не оправдать себя. Однако он оказывается очень эффективным при широковещательной рассылке.

Во-вторых, метод заливки отличается чрезвычайной надежностью. Даже если большая часть маршрутизаторов окажется полностью уничтоженной, любой существующий путь для доставки сообщения будет найден.

Маршрутизация по вектору расстояния (динамический).

В основе его лежит идея, что у каждого маршрутизатора в подсети есть таблица расстояний до каждого маршрутизатора в подсети. Периодически маршрутизатор обменивается информацией со своими соседями и обновляет информацию в таблице. Каждый элемент таблицы состоит из двух полей: первое - номер линии, по которой надо отправлять пакеты, чтобы достичь нужного места, второе - величина задержки до места назначения. Эта величина задержки может быть измерена в разных единицах: скачках, миллисекундах, длине очереди на линии и т.д.

Каждые  $T$  секунд маршрутизатор шлет своим соседям свой вектор задержек до всех маршрутизаторов в подсети. В свою очередь он получает такие же вектора от своих соседей. Кроме этого, он постоянно замеряет задержки до своих соседей. Поэтому, имея вектора расстояний от соседей и зная расстояние до соседей, маршрутизатор всегда может вычислить кратчайший маршрут.

Алгоритм маршрутизации по вектору расстояния теоретически работает хорошо, но у него есть один недостаток (счет до бесконечности): он очень медленно сходится к правильному значению. Информация о появлении хорошего маршрута в подсети распространяется более или менее быстро, а вот данные о потере, разрушении какого маршрута распространяются не столь быстро.

Маршрутизация с учетом состояния линий (динамический).

Маршрутизация на основе векторов расстояний использовалась в сети ARPANET вплоть до 1979 года, когда ее сменил алгоритм маршрутизации с учетом состояния линий. Отказаться от прежнего алгоритма пришлось по двум причинам. Во-первых, старый алгоритм при выборе пути не учитывал пропускную способность линий. Во-вторых, слишком долго приходил к устойчивому состоянию (проблема счета до бесконечности).

В основе алгоритма лежит относительно простая идея, ее можно изложить в пяти требованиях к маршрутизатору. 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.

Иерархическая маршрутизация.

Размер таблиц маршрутов, поддерживаемых маршрутизаторами, увеличивается пропорционально увеличению размеров сети. При этом требуется не только большее количество памяти для хранения этой таблицы, но и большее время центрального процессора для ее обработки. Кроме того, возрастает размер служебных пакетов, которыми обмениваются маршрутизаторы, что увеличивает нагрузку на линии. В определенный момент сеть может вырасти до таких размеров, при которых перестанет быть возможным хранение на маршрутизаторах записи обо всех остальных маршрутизаторах. Поэтому в больших сетях маршрутизация должна осуществляться иерархически.

При использовании иерархической маршрутизации маршрутизаторы разбиваются на отдельные так называемые регионы (regions). Каждый маршрутизатор знает все детали выбора маршрутов в пределах своей области, но ему ничего не известно о внутреннем строении других регионов.

В очень больших сетях двухуровневой иерархии может оказаться недостаточно. Может потребоваться группировать регионы в кластеры, кластеры в зоны, зоны в группы и т. д.

Основным преимуществом иерархической маршрутизации является то, что она имитирует организацию большинства компаний и следовательно, очень хорошо поддерживает их схемы трафика.

Широковещательная маршрутизация.

Широковещанием называется рассылка пакетов по всем пунктам назначения одновременно.

- Многоадресная маршрутизация. Этот метод требует начальных сведений обо всех адресах назначения, и, кроме того, чтобы понять, куда отправить один многоадресный пакет, маршрутизатор должен выполнить столько же действий, сколько и при отправке набора отдельных пакетов.

В каждом пакете содержится либо список адресатов, либо битовая карта, показывающая предпочитаемые хосты назначения.

Когда такой пакет прибывает в маршрутизатор, последний проверяет список, содержащийся в пакете, определяя набор выходных линий, которые потребуются для дальнейшей рассылки.

Маршрутизатором создается копия пакета для каждой из используемых исходящих линий. В нее включаются только те адресаты, для доступа к которым требуется данная линия. Таким образом, весь список рассылки распределяется между исходящими линиями.

После определенного числа пересылок каждый из пакетов будет содержать только один адрес назначения, как и обычный пакет.

- Заливка. Если заливка реализована с помощью порядковых номеров, она эффективно работает с каналами связи, поскольку маршрутизаторы используют относительно простое правило принятия решения.
- Продвижение по встречному пути.

Когда прибывает широкоэмитательный пакет, маршрутизатор проверяет, используется ли та связь, по которой он прибыл, для нормальной передачи пакетов источнику широкоэмитания.

В случае положительного ответа велика вероятность того, что широкоэмитательный пакет прибыл по наилучшему маршруту и является, таким образом, первой копией, прибывшей на маршрутизатор.

Тогда маршрутизатор рассылает этот пакет по всем связям, кроме той, по которой он прибыл.

Однако если широкоэмитательный пакет прибывает от того же источника по другой связи, он отвергается как вероятный дубликат.

Принципиальное преимущество метода продвижения по встречному пути заключается в его вполне приемлемой эффективности при простоте реализации. Как и при заливке, широкоэмитательный пакет отправляется по всем каналам связи только один раз в каждом направлении, и при этом маршрутизатор должен всего лишь знать, как добраться до соответствующего адреса назначения, но не обязан помнить порядковые номера или хранить список всех адресов в пакете.

- Связующее дерево. Представляет собой подмножество сети, включающее в себя все маршрутизаторы, но не содержащее замкнутых путей.

Данный алгоритм в явном виде использует входное дерево или любое другое связующее дерево для маршрутизатора, инициировавшего широкоэмитание.

Если каждый маршрутизатор знает, какие из его линий принадлежат связующему дереву, он может отправить приходящий пакет по всем линиям связующего дерева, кроме той, по которой пакет прибыл.

Такой метод оптимальным образом использует пропускную способность сети, порождая минимальное количество пакетов, требующихся для выполнения работы.

Единственной проблемой этого метода является то, что каждому маршрутизатору необходимо обладать информацией о связующем дереве. Иногда такая информация доступна (например, в случае маршрутизации с учетом состояния линий все маршрутизаторы обладают полными сведениями о топологии сети и поэтому могут вычислить связующее дерево), но иногда — нет (при маршрутизации по векторам расстояний).

#### Борьба с перегрузкой.

Когда количество пакетов, передаваемых одновременно по сети превышает некий пороговый уровень, производительность сети начинает снижаться. Такая ситуация называется перегрузкой.

Когда число пакетов, посылаемых хостами в сеть, не превышает ее пропускной способности, число доставленных пакетов пропорционально числу отправленных. Однако, когда нагрузка на сеть приближается к пропускной способности, большие объемы трафика постепенно заполняют буферы маршрутизаторов, и в результате некоторые пакеты теряются. Эти потерянные пакеты расходуют часть пропускной способности, поэтому число доставленных пакетов оказывается ниже идеальной кривой. Это означает, что сеть перегружена.

Наличие перегрузки означает, что нагрузка на сеть (временно) превышает возможности (сетевых) ресурсов. Соответственно, существует два возможных решения: увеличить ресурсы или снизить нагрузку.

- Обеспечение. Иногда при возникновении перегрузки возможно динамическое добавление ресурсов,

например, подключение свободных маршрутизаторов или резервных линий.

- Маршрутизация с учетом состояния трафика. Чтобы максимально эффективно использовать пропускную способность сети, маршруты могут строиться в соответствии со специальными схемами трафика, которые меняются в течение суток по мере того, как пользователи просыпаются и ложатся спать в различных временных зонах. В таком случае полезно также разделять трафик, направляя его по разным линиям.

Если увеличение пропускной способности оказывается невозможным, то единственным средством борьбы с перегрузкой является снижение нагрузки.

- Управление доступом. В сети виртуальных каналов новые соединения могут быть отклонены, если они приведут к перегрузке сети.
- Регулирование трафика. Когда перегрузка неизбежна, сеть может послать сообщение обратной связи тому отправителю, чей трафик вызывает проблему. Сеть может попросить отправителя уменьшить трафик или же сделать это сама.

Если все эти методы не работают, то используется сброс нагрузки. Сеть вынуждена удалить пакеты, которые она не может доставить.

Ключевой проблемой для маршрутизатора, заваленного пакетами, является выбор пакета, который будет отвергнут. Выбор может зависеть от типа приложений, использующих данную сеть. Для передачи файла более старый пакет ценится выше нового, так как отвержение пакета номер 6 и сохранение пакетов с номерами, например, с 7 по 10 лишь заставит получателя выполнить лишнюю работу: поместить в буфер данные, которые он еще не может использовать. Для мультимедийных приложений, работающих в реальном времени, напротив, новый пакет важнее старого. Причина в том, что пакеты становятся ненужными, если они задерживаются и не приходят вовремя.

Первую стратегию (старое лучше нового) часто называют винной стратегией, а вторую (новое лучше старого) — молочной стратегией, так как большинство людей предпочитают пить свежее молоко и выдержанное вино, а не наоборот.