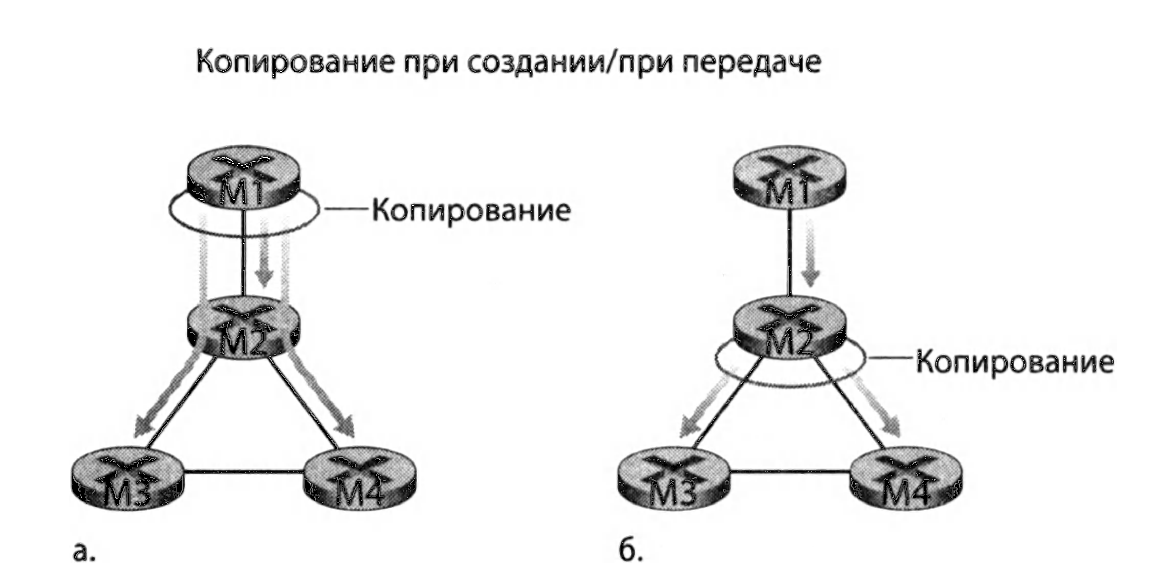
**Широковещательная и групповая маршрутизация**

Ранее мы говорили в основном о таких протоколах маршрутизации, которые поддерживают одноадресное взаимодействие по двухточечному принципу («точка-точка»). В таких случаях имеется всего один хост-отправитель, отсылающий пакет только одному хосту-получателю. В этом разделе мы изучим протоколы для широковеща­тельной и групповой маршрутизации. При **широковещательной марш­рутизации** (broadcast routing) сетевой уровень обеспечивает доставку пакета от отправителя всем остальным хостам в сети. При **групповой маршрутизации** (multicast routing) один отправитель отсылает копии пакета подмножеству других хостов сети.

**Алгоритмы широковещательной маршрутизации**

Вероятно, простейший способ обеспечения широковещательной коммуникации таков: хост-отправитель отправляет по одной копии па­кета каждому из хостов-получателей, как показано на рис. 4.43(a). Имея JV получателей, исходный узел просто создает *N* копий пакета, посылает каждую копию на конкретный адрес при помощи одноадресной марш­рутизации. Такой метод N-адресной широковещательной маршрутиза­ции прост и не требует никаких новых протоколов маршрутизации се­тевого уровня, дублирования пакетов или специального функционала пересылки. Однако у этого подхода есть ряд недостатков. Первый из них — неэффективность. Если исходный узел соединен с остальной се­тью единственным каналом, то *N* отдельных копий (одного и того же) пакета будут проходить по этому каналу. Разумеется, было бы более це­лесообразно послать одну копию пакета на расстояние одного перехода (до первого транзитного узла), а затем приказать этому транзитному узлу создать нужное количество копий пакета на основе первой и уже оттуда переслать их все куда следует. Соответственно, работа строи­лась бы более эффективно, если бы разные узлы сети (а не только ис­ходный) умели создавать дополнительные копии пакета. Например, на рис. 4.43(6) по каналу М1-М2 проходит всего одна копия пакета. Затем на узле М2 этот пакет копируется, и по каналам М2-МЗ и М2-М4 отсы­лаются полученные таким образом два экземпляра пакета.

****

**Рис. 4.43. Сравнение копирования на исходном узле и копирования**

**на транзитных узлах**

Другие недостатки N-адресной маршрутизации не столь очевидны, но не менее существенны. При N-адресной маршрутизации неявно пред­полагается, что отправителю известны адреса всех хостов-получателей широковещательной рассылки. Но как он получает эту информацию? Скорее всего, при этом потребуется задействовать дополнительные ме­ханизмы работы с протоколами (например, широковещательный запрос членства или протокол регистрации узлов назначения). В результате возникнет еще больше издержек и, что гораздо важнее, сам протокол значительно усложнится, тогда как в исходном варианте он казался до­вольно простым. Наконец, последний недостаток N-адресной маршру­тизации заключается в том, для каких целей обычно применяется ши­роковещание. В разделе 4.5 мы узнали, что протоколы маршрутизации с учетом состояния каналов применяются для широковещательного распространения информации о состоянии каналов, которая, в свою очередь, используется для расчета одноадресных *маршрутов.* Разумеет­ся, в тех случаях, когда широковещание применяется для создания и об­новления одноадресных маршрутов, будет (мягко говоря!) неразумно выстраивать широковещательную передачу информации на одноадрес­ной архитектуре.

С учетом недостатков N-адресной широковещательной маршрути­зации больший интерес вызывают такие подходы, при которых сами сетевые узлы активно участвуют в размножении и перенаправлении пакетов, а также в расчете широковещательных маршрутов. Ниже мы исследуем ряд таких подходов в виде графа, который мы уже рассматри­вали ранее. Итак, вновь представим сеть как граф *G = (N,E)>* где *N—* это множество узлов и набор ребер *Еу* причем каждое ребро объеди­няет пару узлов из множества *N* В нашей нотации мы допустим неболь­шую неаккуратность и будем обозначать через *N* и множество узлов, и мощность (размер) этого множества *(\N\)* в случаях, когда значение *N* является недвусмысленным.

**Неуправляемая лавинообразная маршрутизация**

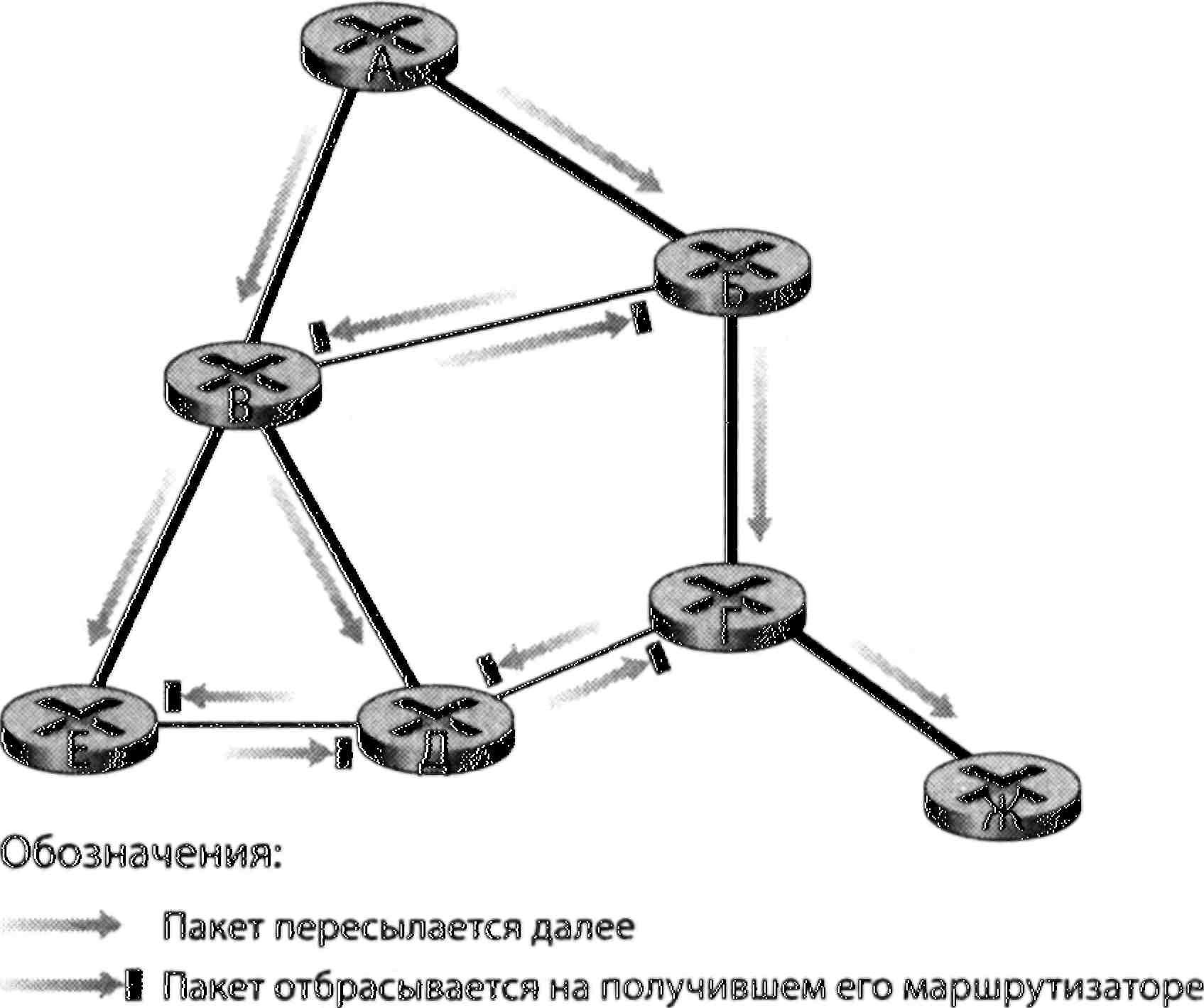
Самый простой подход к широковещательной передаче данных на­зывается **лавинообразной маршрутизацией** (флудингом). В таком слу­чае исходный узел посылает копию пакета всем своим соседним узлам. Как только узел получает пакет, он копирует его и пересылает всем своим соседям (кроме того, от которого он получил пакет). Разумеет­ся, если граф является связанным, то такая схема рано или поздно при­ведет к доставке широковещательно распространяемого пакета ко всем его узлам. Хотя эта схема проста и красива, у нее есть катастрофический недостаток (прежде чем читать далее, попробуйте описать его сами). Если в графе есть циклы, то одна или несколько копий широковеща­тельного пакета попадут в бесконечный цикл. Например, на рис. 4.43 лавинообразная маршрутизация пойдет следующим образом: от М2 к МЗ, от МЗ к М4, от М4 к М2, а от М2 (опять!) к МЗ и т. д. В таком про­стом сценарии в бесконечный цикл попадают два пакета, один движется по часовой стрелке, другой — против часовой. Но возможен еще более катастрофический исход: если узел соединен более чем с двумя други­ми узлами, он начнет создавать и передавать множество копий пакета (которые будут только множиться на других узлах, соединенных более чем с двумя соседями) и т. д. Такое состояние называется **широковеща­тельным штормом** и возникает по причине бесконечного размножения передаваемых пакетов. Вскоре пакетов станет настолько много, что сеть будет буквально забита ими. В домашнем задании к этой главе есть во­просы, посвященные анализу этой проблемы и расчету скорости нарас­тания широковещательного шторма.

**Управляемая лавинообразная маршрутизация**

Чтобы не допустить возникновения широковещательного шторма, узел должен осмысленно выбирать, когда следует и когда не следует до­бавлять пакет в лавину (например, пакет не должен попадать в лавину, если этот узел уже получал и отправлял в лавину его копию). На прак­тике такой механизм реализуется одним из нескольких способов.

При **контролируемой лавинообразной маршрутизации с использо­ванием порядковых номеров** исходный узел записывает в широкове­щательный пакет свой адрес (или другой уникальный идентификатор), а также **порядковый широковещательный номер.** Затем узел отправля­ет этот пакет всем своим соседям. Каждый узел ведет список исходных адресов и порядковых номеров всех пакетов, которые он уже получил, скопировал и переслал широковещательным образом. Когда узел полу­чает широковещательный пакет, он сначала проверяет, указан ли этот пакет в его списке. Если это так, то пакет отбрасывается; в противном случае пакет копируется и по одному его экземпляру передается всем узлам, смежным с данным (кроме того узла, от которого был получен пакет). В протоколе Gnutella управляемая ла­винообразная маршрутизация с применением порядковых номеров ис­пользуется для широковещания запросов по наложенной (оверлейной) сети. В протоколе Gnutella копирование и передача сообщений выпол­няется на прикладном, а не на сетевом уровне.

Другой способ управляемой лавинообразной маршрутизации имену­ется **переадресацией в обратном направлении** (reverse path forwarding, RPF), иногда также называется «широковещание в обратном направ­лении» (reverse path broadcast, RPB). В основе этого механизма лежит простая, но элегантная идея. Когда при широковещательной передаче маршрутизатор получает пакет с заданным адресом, он пересылает этот пакет по всем своим исходящим каналам (кроме того, по которому он поступил), но лишь при условии, что пакет прибыл по тому каналу, ко­торый находится на кратчайшем одноадресном пути к источнику пакета. В противном случае узел просто отбрасывает такой входящий пакет, не пересылая его ни по одному из своих исходящих каналов. Такой пакет действительно можно смело отбросить, поскольку маршрутизатору известно, что он либо уже получал такой пакет, либо получит его именно по тому каналу, который расположен на кратчайшем пути от данного узла к узлу-отправителю. Можете сами подробнее исследовать этот случай и убедиться, что он действительно не допускает возникновения ни пе­тель, ни широковещательного шторма. Обратите внимание: в механизме RPF не применяется одноадресная маршрутизация для доставки пакета к месту назначения; кроме того, от маршрутизатора не требуется знать полный кратчайший путь от себя до источника пакета. RPF требует, что­бы маршрутизатор знал лишь ближайший узел, смежный с этим марш­рутизатором и расположенный на кратчайшем пути к отправителю. Он использует идентификатор смежного узла лишь для того, чтобы опреде­лить, отправлять или нет полученный пакет в лавинообразную рассылку.

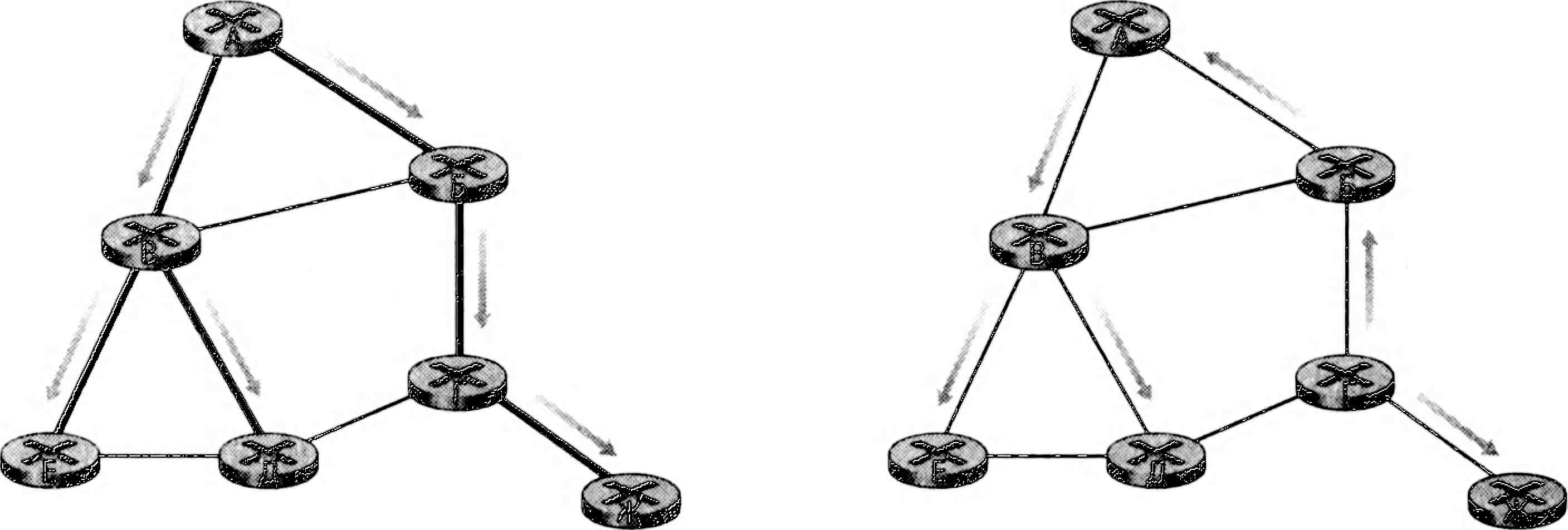


**Рис. 4.44. Переадресация в обратном направлении**

Механизм RPF проиллюстрирован на рис. 4.44. Предположим, что ка­налы, обозначенные жирными линиями, соответствуют путям с наимень­шей стоимостью от адресатов к исходному узлу (Л). Узел Л сначала широ­ковещательно передает пакет из источника Л узлам *Б* и 5. Узел *Б* перешлет пакет, полученный из источника Л (поскольку Л находится на кратчайшем пути от Б к Л) узлам *В и Г.* Узел *Б* проигнорирует (то есть, отбросит, не пересылая) любые пакеты из источника Л, которые он получит от других узлов (например, от маршрутизаторов *В* или *Г).* Теперь давайте обратим внимание на узел В, который получит пакет Л и непосредственно от Л, и от *Б.* Поскольку *Б* не находится на кратчайшем пути от *В* до Л, *В* проигнори­рует все пакеты с источника Л, полученные от *Б.* Но если *В* получит пакет Л непосредственно от Л, то он перешлет этот пакет на узлы *Б,ДиЕ.*

**Широковещательная маршрутизация по методу связующего дерева**

Итак, при управляемой лавинообразной маршрутизации с приме­нением порядковых номеров и при RPF удается не допускать широко­вещательных штормов, но эти алгоритмы не позволяют полностью ис­ключить широковещательную передачу избыточных пакетов. Так, на рис. 4.44 узлы *Б,В,Г,Д и Е* получат по одному или по два лишних пакета каждый. В идеале при широковещательной передаче каждый узел дол­жен принимать всего по одному экземпляру пакета. Изучив изображен­ное на рис. 4.45(a) дерево узлов, соединенных жирными линиями, мы ви­дим, что если бы при широковещательной передаче пакеты шли только по каналам из этого дерева, то каждый узел в сети получил бы ровно по одному экземпляру широковещательно передаваемого пакета. Именно этого мы и добиваемся! Здесь мы видим пример **связующего дерева** — такое дерево включает в себя абсолютно все узлы графа. Более строгая формулировка будет звучать так: связующее дерево графа *G* = *(N,E) —* это такой граф *G'* = *(N,E'),* в котором *Е'* является подмножеством *Е>* граф *G* является связным, граф *G'* не содержит циклов и включает в себя все исходные узлы *G.* Если каждый канал имеет определенную стоимость, а стоимость дерева равна сумме стоимостей всех его каналов, то связую­щее дерево графа, обладающее минимальной стоимостью, выделяется из всех связующих деревьев графа и (что неудивительно) называется **минимальным связующим деревом** этого графа.

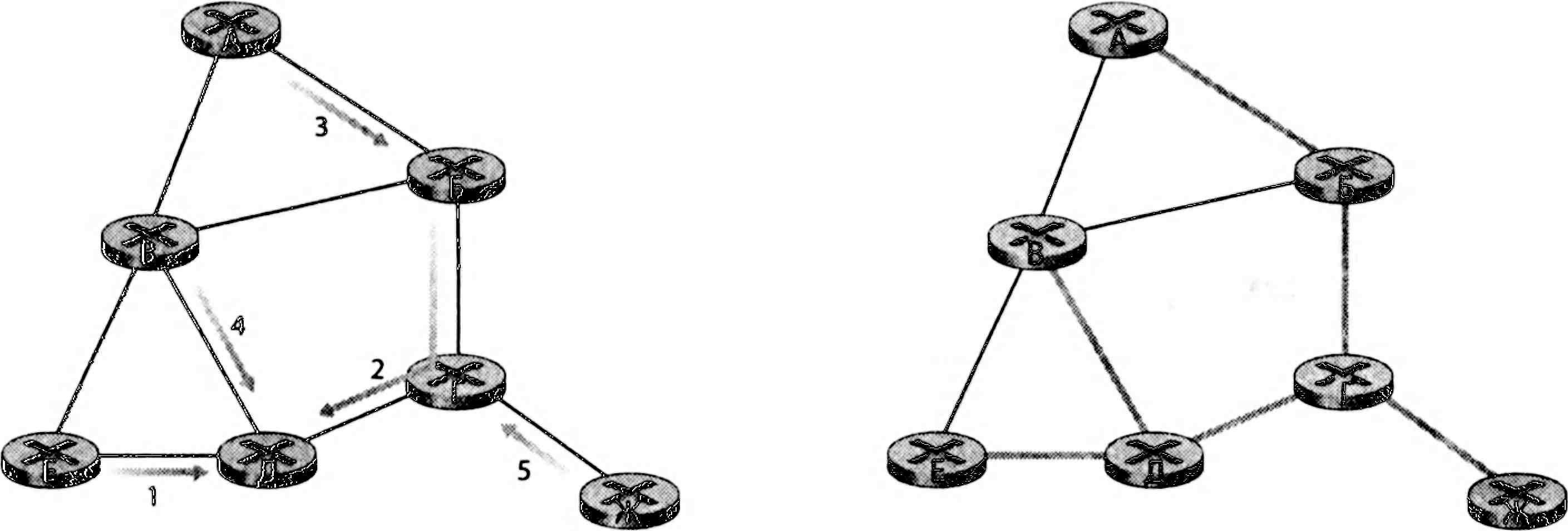


а. Широковещание инициировано на узле А б. Широковещание инициировано на узле Г

**Рис. 4.45. Широковещание по связующему дереву**

Итак, альтернативный подход к обеспечению широковещательной передачи данных в сети — создание связующего дерева. Когда исходно­му узлу требуется послать широковещательный пакет, он рассылает его по всем инцидентным каналам, относящимся к связующему дереву. Затем узел, получивший широковещательный пакет, рассылает его всем своим соседним узлам в связующем дереве (кроме того, от которого он получил этот пакет). Связующее дерево позволяет не только избавить­ся от лишних широковещательных пакетов, но и начинать широковеща­тельную передачу с любого узла, как показано на рис. 4.45(a) и 4.45(6). Обратите внимание: узлу не требуется информация обо всем дереве; он должен лишь знать, какие из его узлов-соседей по графу *G* относятся к связующему дереву.

Основная сложность, возникающая при использовании связующих деревьев, заключается в необходимости их создания и поддержки. Раз­работаны многочисленные распределенные алгоритмы для создания связующих деревьев. Здесь мы рассмотрим только один такой ал­горитм. Это **центрированный** подход, при котором в дереве определяет­ся центральный узел (также называемый **точкой рандеву** или **ядром).** Далее все узлы одноадресно передают центральному узлу сообщения для объединения дерева. Такое сообщение пересылается методом од­ноадресной маршрутизации к центру, пока не достигнет либо одного из узлов, уже относящихся к связующему дереву, либо, собственно, центра. Можно сказать, что такой путь «пересаживается» или «прививается» к имеющемуся связующему дереву



а. Пошаговое создание связующего дерева 6. Готовое связующее дерево

**Рис. 4.46. Создание связующего дерева, начиная с центрального узла**

На рис. 4.46 продемонстрировано построение связующего дерева с центральной точкой. Предположим, узел *Д* выбран в качестве центра дерева. Далее допустим, что узел *Е* первым присоединяется к дереву и от­правляет узлу *Д* объединяющее сообщение. Один канал *ДЕ* становится «ростком» связующего дерева. Затем к этому дереву подключается узел *Б*, отправляющий узлу *Д* объединяющее сообщение. Предположим, что одноадресный маршрут от *Д* до *Б* пролегает через Г. В результате имеем «прививку» пути *БГД к* связующему дереву. Следующим к группе под­ключается узел Л, направляя объединяющее сообщение узлу Д. Если одноадресный путь от *А* до *Д* пролегает через *Б,* то, поскольку узел *Б* уже относится к связующему дереву, прибытие объединяющего сооб­щения от узла *А к* узлу *Б* приведет к образованию канала *АБ,* который будет сразу же «привит» к связующему дереву Следующим к дереву присоединится узел В, который направит объединяющее сообщение не­посредственно к *Д.* Наконец, поскольку одноадресная маршрутизация от узла *Ж* к *Д* должна идти через узел Г, канал *ЖГ*«прививается» к свя­зующему дереву на узле *Г.*

**Широковещательные алгоритмы на практике**

На практике широковещательные протоколы применяются как на прикладном, так и на сетевом уровне. Протокол Gnutella использу­ет широковещательную передачу данных на прикладном уровне для рассылки запросов содержимого среди равноправных хостов Gnutella. В данном случае связь между двумя распределенными равноправными процессами прикладного уровня по протоколу Gnutella представляет собой обычное TCP-соединение. Gnutella задействует своеобразный вариант управляемой лавинообразной маршрутизации с применением порядковых номеров, где 16-битный идентификатор и 16-битный опи­сатель полезного содержимого (указывающий тип сообщения Gnutella) используются для определения того, что ранее происходило с полу­ченным широковещательным запросом: был ли он получен, копирован и переадресован. Протокол Gnutella также применяет поле TTL (пред­писанное время жизни), позволяющее ограничить количество тран­зитных узлов (переходов), которые может преодолеть лавинообразно направленный запрос. Когда процесс Gnutella получает и копирует за­прос, этот процесс сначала понижает значение в поле TTL на единицу, а только потом пересылает запрос. Соответственно, лавинообразный за­прос в Gnutella сможет достичь лишь тех узлов, количество переходов до которых на прикладном уровне не превышает исходного значения в поле TTL этого процесса. Поэтому применяемый в протоколе Gnutella механизм лавинообразной маршрутизации иногда именуется *«лавино­образная маршрутизация с ограниченным распространением»*.

Кроме того, особая разновидность управляемой лавинообразной маршрутизации с применением порядковых номеров используется для широковещательной рассылки объявлений о состоянии каналов (LSA) по протоколу OSPF, а также по протоколу IS-IS (протокол маршрутизации промежуточных систем). Для идентификации таких объяв­лений протокол OSPF использует 32-битные порядковые номера, а так­же 16-битное поле устаревания. Как вы помните, узел протокола OSPF широковещательно распространяет объявления LSA по непосредствен­но связанным с этим узлом каналам, в случае, когда стоимость пути до узла-соседа изменяется, либо когда канал включается/отключается. Порядковые номера объявлений о состоянии канала используются для выявления дублирующихся объявлений такого рода, но также выполня­ют в протоколе OSPF и другую важную функцию. При лавинообразной маршрутизации не исключено, что объявление LSA, сгенерированное исходным хостом в момент времени *t>* прибудет на узел назначения уже *после* наступления момента t+δ, в который исходный узел успеет сгене­рировать новое объявление об изменении состояния канала. Поскольку исходный узел использует порядковую нумерацию сообщений, более старое сообщение можно отличить от более нового. Поле устаревания функционально похоже на поле предписанного времени жизни (TTL). Сначала в этом поле записано значение 0, а после каждого перехода в процессе лавинообразной рассылки оно увеличивается. Это значение увеличивается даже в то время, пока пакет находится в памяти и ожи­дает, пока его направят в лавинообразную рассылку. Здесь мы не можем более подробно обсудить алгоритм лавинообразной рассылки LSA, но отметим, что разработка протоколов для такой маршрутизации являет­ся очень сложным делом. В работе Перлмана и документе RFC 789описан случай, в котором неправильно переданные двумя неисправны­ми маршрутизаторами объявления о состоянии каналов вызвали крах всей сети ARPAnet, где применялась одна из ранних версий алгоритма лавинообразной маршрутизации объявлений LSA.

**Групповая маршрутизация**

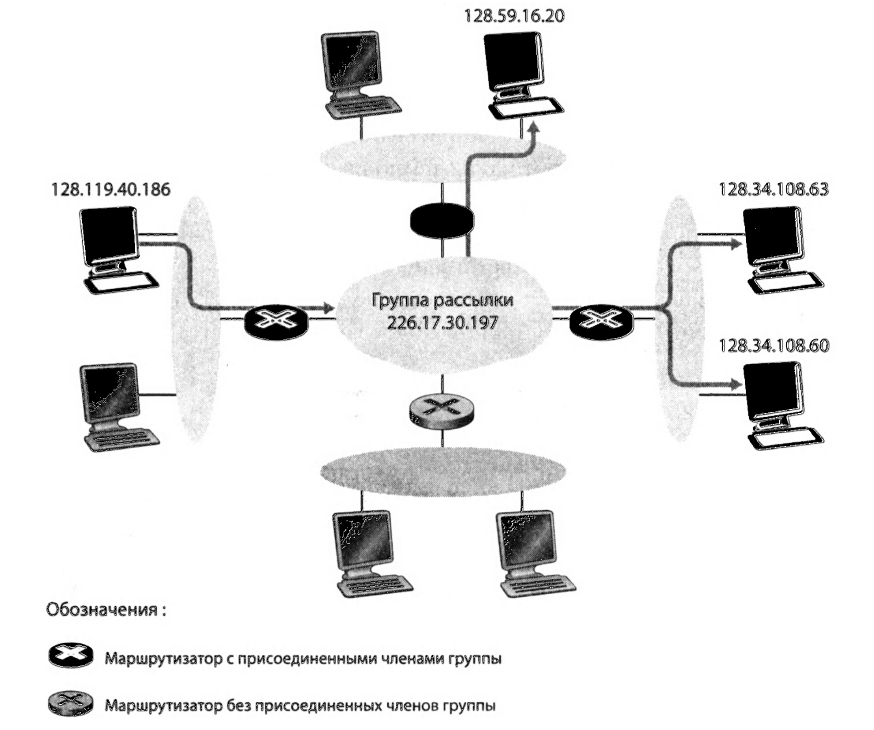
В предыдущем разделе мы убедились, что при широковещатель­ном распространении данных пакет доставляется всем без исключения узлам в сети. В этом разделе мы поговорим о **групповой** (многоадрес­ной) маршрутизации, при которой пакеты доставляются лишь *подмно­жеству* узлов в сети. Ряд недавно появившихся сетевых приложений требуют доставки пакетов от одного или более отправителей к группе получателей. Среди таких задач требуется отметить передачу массива данных — например, передачу пакета обновлений приложения пользо­вателям, которые в них нуждаются; потоковую передачу мультимедий­ной информации (например, аудио, видео и текста распределенной аудитории во время виртуальной лекции); приложения с разделяемыми данными (например, приложения для телеконференций и виртуальных лекционных досок, которые совместно используются группой распреде­ленных клиентов), потоки данных (например, для представления бир­жевых котировок), обновление веб-кэша, интерактивные игры в Ин­тернете (например, распределенные интерактивные виртуальные среды или многопользовательские игры).

При групповой рассылке мы сразу же сталкиваемся с двумя пробле­мами: как идентифицировать получателей многоадресной дейтаграммы и как адресовать эту дейтаграмму

В случае одноадресной рассылки IP-адрес получателя содержится в каждой одноадресной IP-дейтаграмме и означает единственного по­лучателя. Но в случае групповой рассылки имеется несколько получа­телей. Есть ли смысл помещать в каждую групповую дейтаграмму IP-адреса всех получателей, входящих в группу? Хотя такой подход может работать для небольшого количества получателей, он плохо подходит для случая, когда их сотни и тысячи. Адресные данные просто займут весь объем дейтаграммы, вытеснив из нее полезную информацию. Кро­ме того, чтобы явно указать *всех* получателей, отправитель должен знать все их идентификаторы и адреса. Далее будет показано, что в некоторых ситуациях такое требование может быть нежелательным.

По вышеуказанным причинам в архитектуре Интернета (а также в других архитектурах, таких, как ATM-сети) групповая дейтаграмма **адресуется косвенно.** То есть для группы получателей используется один идентификатор, а копия дейтаграммы, адресованной группе по­лучателей при помощи этого единственного идентификатора, достав­ляется всем членам этой группы. В Интернете единый идентификатор, соответствующий группе получателей, представляет собой групповой адрес класса D (см. раздел «Интернет-протокол»). Группа получателей, ассоциированная с адресом класса D, называется **группой рассылки.** На рис. 4.47 четыре хоста (с указанными IP-адресами) ассоциированы с группой рассылки с IP-адресом 226.17.30.197. Эти хосты будут по­лучать все дейтаграммы, направляемые по данному адресу. Сложность заключается в том, что у каждого хоста имеется уникальный IP-адрес, полностью независимый от адреса группы рассылки.

Хотя идея группы рассылки проста, она поднимает ряд вопросов. Как создается группа и как она прекращает свое существование? Каким образом выбирается групповой адрес? Как к группе добавляются новые хосты (в качестве отправителей или получателей)? Может ли кто угод­но присоединиться к группе (и передавать или принимать данные, по­сылаемые этой группе) или членство в группе ограничивается, и если да, то кем? Известны ли членам группы идентификаторы других членов группы? Как сетевые маршрутизаторы взаимодействуют друг с другом, чтобы доставить групповую дейтаграмму всем членам группы? В Ин­тернете на все эти вопросы отвечает протокол IGMP. Мы рассмотрим протокол IGMP, а затем вернемся к более общим вопросам.

****

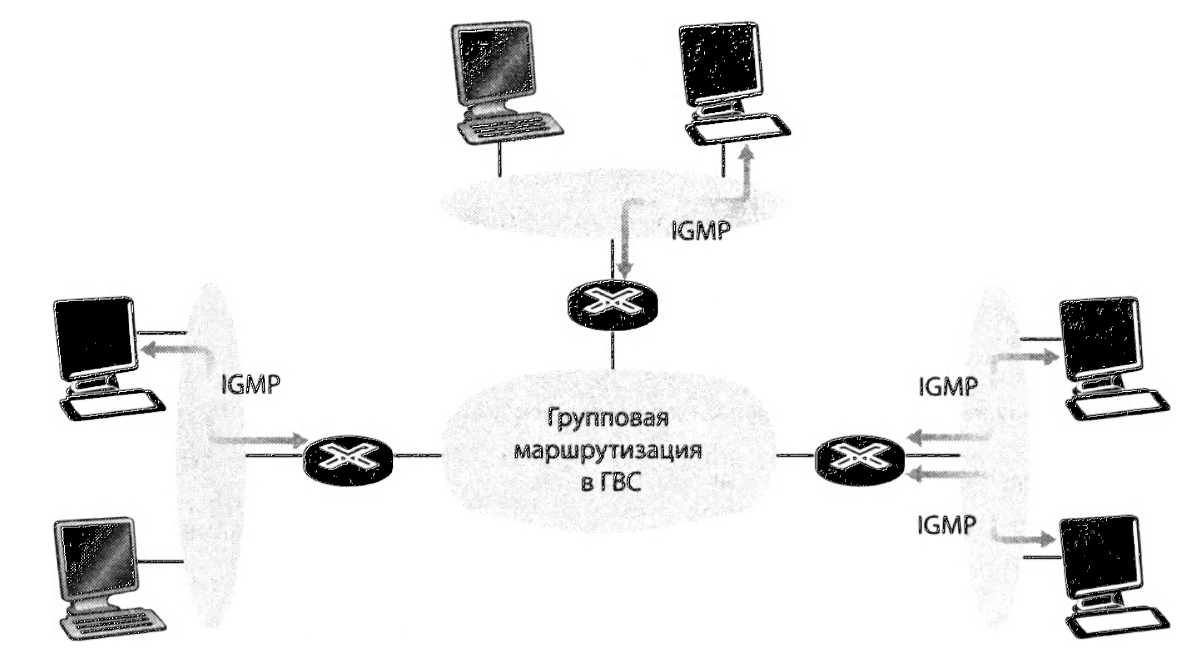
**Рис. 4.47. Адресованная в группу рассылки дейтаграмма доставляется всем**

**членам группы**

**Протокол управления группами Интернета**

Протокол IGMP (Internet Group Management Protocol — протокол управления группами Интернета) версии 3 работает между хостом и соединенным с ним напрямую маршрутизатором (который можно рас­сматривать как первый маршрутизатор на пути следования входящих дейтаграмм или последний маршрутизатор на пути следования исходя­щих дейтаграмм), как показано на рис. 4.48. На рис. 4.48 изображены три групповых маршрутизатора (расстояние между любыми двумя из них составляет один переход), каждый из которых соединен с парой хостов через локальный интерфейс. В данном примере локальный интерфейс связан с локальной сетью, и, как правило, несколько хостов локальной сети одновременно являются членами той или иной группы рассылки.

Протокол IGMP предоставляет хосту средство информировать со­единенный с ним маршрутизатор о том, что работающее на хосте при­ложение желает присоединиться к определенной группе рассылки. Учитывая ограниченность сферы действия протокола IGMP хостом и соединенным с ним маршрутизатором, для координирования группо­вых маршрутизаторов (включая присоединенные) в Интернете, очевид­но, необходимы другие протоколы, позволяющие доставлять групповые дейтаграммы к местам назначения. Это обеспечивается при помощи ал­горитмов групповой маршрутизации сетевого уровня, примеры которых мы вскоре рассмотрим. Соответственно, групповая передача данных на сетевом уровне в Интернете состоит из двух взаимодополняющих ком­понентов: протокола IGMP и протоколов групповой маршрутизации.

****

**Рис. 4.48. Два компонента групповой маршрутизации на сетевом уровне в Интернете: протокол IGMP и протоколы групповой маршрутизации**

В протоколе IGMP используются только три типа сообщений. По­добно ICMP, сообщения IGMP, вкладываются (инкапсулируются) в IP-дейтаграммы, с номером протокола 2. Общее сообщение membership\_query посылается маршрутизатором всем хостам, присоединенным к его интерфейсу (например, всем хостам локальной сети), чтобы узнать обо всех группах рассылки, членами которых стали хосты данного интерфейса.

Хосты отвечают на сообщение membership\_query IGMP-сообще-нием member shipreport. Хост может также генерировать сообщения membership\_report, не ожидая сообщения membership\_query от маршрутизатора, когда приложение впервые присоединяется к группе рассылки. Последний тип IGMP-сообщений — это leave\_group. Ин­тересно отметить, что это сообщение не является обязательным! Но как тогда маршрутизатор определяет, что в данной локальной сети не оста­лось хостов, входящих в определенную группу? Оказывается, маршру­тизатор логически заключает, что в данную группу рассылки более не входят присоединенные к нему хосты, если ни один хост не отвечает на его сообщение membership\_query с конкретным групповым адресом. Интернет-протоколы, в которых по истечении некоторого интервала времени информация об адресах удаляется, иногда называют протокола­ми с неустойчивым состоянием. Таким протоколом является протокол IGMP, в котором информация о наличии членов определенной группы рассылки среди хостов локальной сети удаляется по истечении задан­ного интервала времени (в этом случае интервал задает периодически посылаемое маршрутизатором сообщение membership\_query), если оно не обновляется явно (при помощи посылаемого хостом сообщения membership\_report).

Термин «неустойчивое состояние» был предложен Кларком, опи­савшим идею периодических сообщений об обновлении состояния, от­сылаемых конечной системой, и предположил, что при их применении состояние можно восстанавливать на конечной системе даже после ава­рийного выключения сети — на основании информации, содержащейся в последующих сообщениях об обновлении. Такой механизм представ­ляется совершенно прозрачным для конечной системы и не требует вы­зывать в ней какие-либо явные восстановительные процедуры:

*«...информация о состоянии представляется некритичной при под­держании нужного типа обслуживания в ходе работы с потоком. Действительно такой способ обслуживания будет жестко регла­ментироваться конечными системами, которые периодически об­мениваются сообщениями, чтобы гарантировать, что с потоком информации ассоциировано нужное обслуживание. Таким образом, информация о состоянии, ассоциированная с потоком, может быть потеряна при аварийном отключении, но без необратимого уничтожения тех функциональных возможностей, которые при этом ис­пользуются. Такую концепцию я называю «неустойчивым состояни­ем», и она вполне позволяет нам достичь наших приоритетных целей, связанных с жизнеспособностью и гибкостью...»*

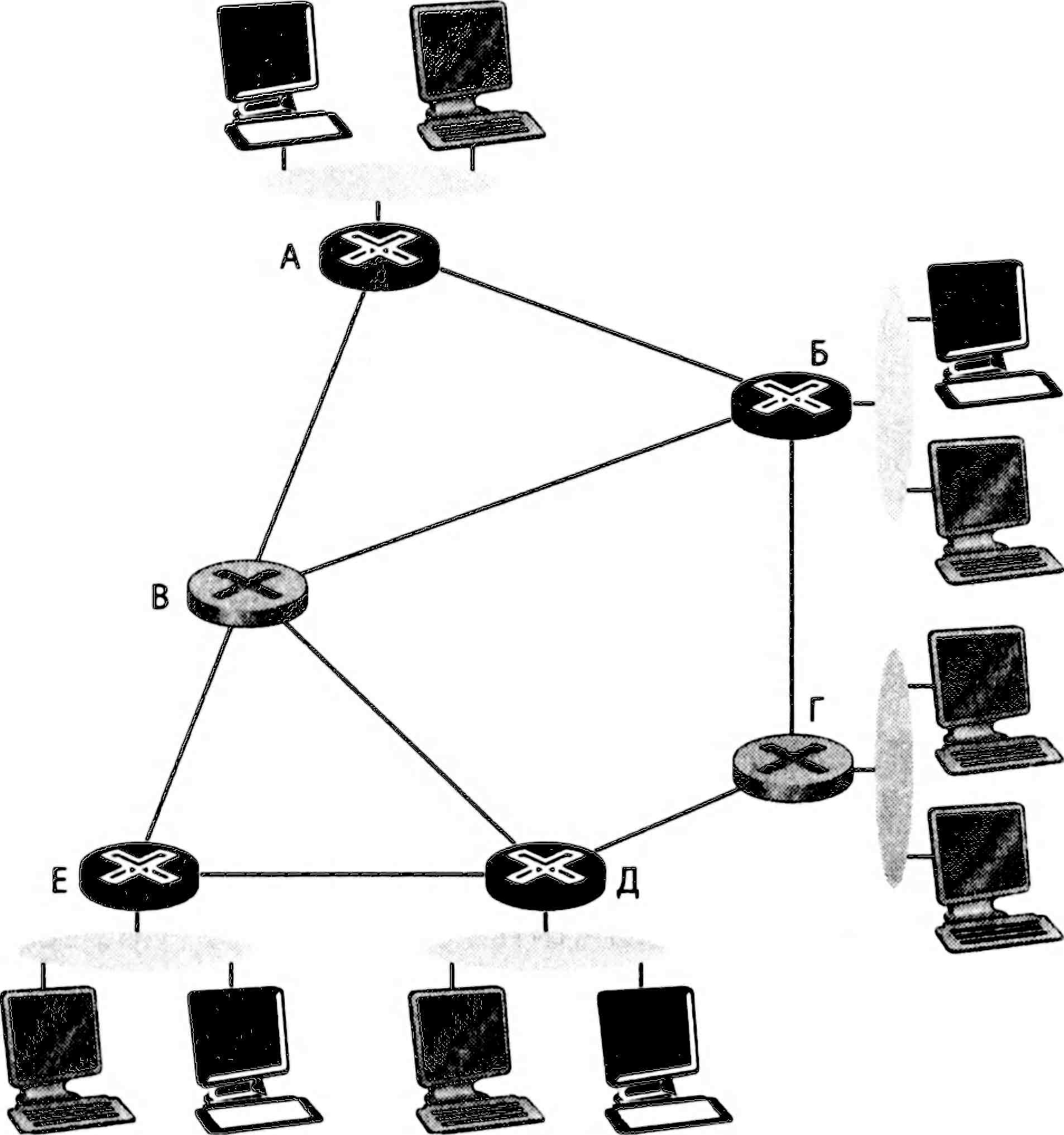
Утверждается, что протоколами с неустойчивым состоянием про­ще управлять, нежели протоколами с устойчивым состоянием. Для по­следних нужны не только механизмы явного добавления или удаления состояний (то есть информации о членстве в группах), но также какие-то средства восстановления в ситуации, когда один из этих механизмов преждевременно завершит работу или вообще выйдет из строя.

**Алгоритмы групповой маршрутизации**

Рисунок 4.49 призван проиллюстрировать **задачу групповой марш­рутизации.** Хосты, являющиеся членами группы рассылки, показаны на рисунке темными, как и непосредственно соединенные с ними маршру­тизаторы. Как видно из рисунка, принимать групповой трафик нужно не всем маршрутизаторам, а только тем, чьи хосты являются членами группы рассылки, то есть маршрутизаторам А, Б, Д и Е. Маршрутизато­рам В и Г не нужен групповой трафик, так как присоединенные к марш­рутизатору Г хосты не являются членами группы рассылки, а у маршру­тизатора В вообще нет непосредственно присоединенных хостов. Цель групповой маршрутизации заключается в том, чтобы построить дерево, связывающее все маршрутизаторы, присоединенные хосты которых от­носятся к данной группе рассылки. При этом групповые пакеты будут направляться по этому дереву от отправителя ко всем хостам, входящим в дерево группы рассылки. Разумеется, дерево может содержать марш­рутизаторы, не имеющие хостов, относящихся к данной группе рассыл­ки (например, как видно из рисунка, невозможно связать маршрутиза­торы А, Б, Д и Е в дерево, не включив в него маршрутизатор В или Г).

На практике для построения дерева групповой маршрутизации при­меняются два подхода, отличающиеся тем, используется ли общее дере­во для *нескольких* отправителей или для каждого отправителя создается специальное.

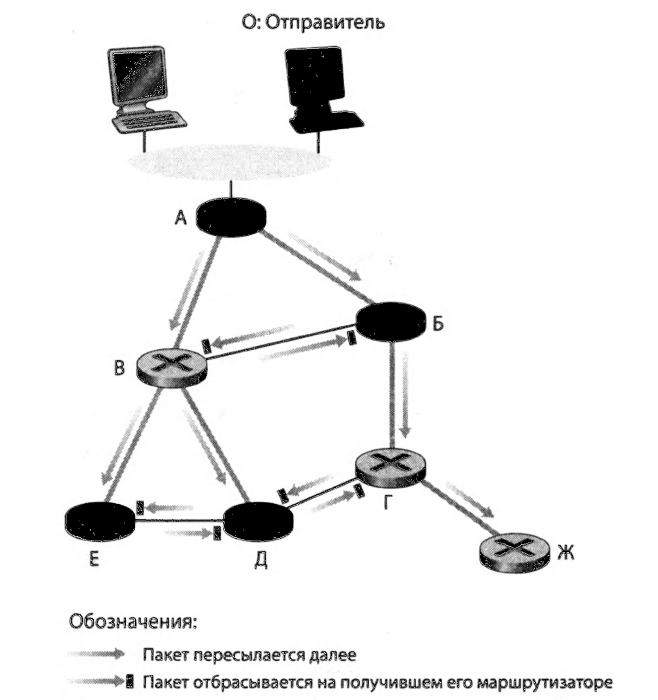
**• *Групповая маршрутизация с общим деревом.*** Как и в случае широко­вещательной передачи данных с применением связующего дерева, групповая маршрутизация с общим деревом осуществляется путем построения дерева, включающего в себя все граничные маршрутиза­торы и связанные с ними хосты, относящиеся к групповой рассылке. На практике дерево для групповой маршрутизации выстраивается начиная от центра, включает в себя все граничные маршрутизаторы и связанные с ними хосты, которые подключаются к этой группе, от­сылая центральному узлу (одно)адресные сообщения о присоедине­нии. Как и в широковещательном случае, сообщение о присоединении отправляется по направлению к центральному узлу одноадресным методом до тех пор, пока либо не попадет на маршрутизатор, отно­сящийся к группе, либо не достигнет центра. Все маршрутизаторы, расположенные по пути, преодолеваемому сообщением по направ­лению к центру, затем переадресуют полученные групповые пакеты тому маршрутизатору, который инициировал подключение к груп­повой передаче. При организации такой маршрутизации критически важен вопрос выбора центрального узла. Соответствующие алгорит­мы рассматриваются в работах Уолла, Талера и Эстрина.



**Рис. 4.49. Хосты при групповой передаче данных, примыкающие к ним**

**маршрутизаторы и другие маршрутизаторы**

**• *Групповая маршрутизация с использованием дерева для каждого ис­точника.*** В то время как при групповой передаче данных с использованием общего дерева создается единое разделяемое дерево, по кото­рому направляются пакеты от *всех* отправителей, рассматриваемый подход предполагает построение деревьев групповой маршрутиза­ции для *каждого* из источников, находящихся в группе. На практи­ке **применяется алгоритм *переадресации в обратном направлении*** (RPF) с исходным узлом *х,* при помощи которого создается дерево групповой передачи дейтаграмм, исходящих с источниках. Широко­вещательный вариант алгоритма RPF, изученный нами ранее, нужно немного усовершенствовать для применения в групповой передаче данных. Чтобы уточнить этот момент, рассмотрим маршрутизатор Г на рис. 4.50.

****

**Рис. 4.50. Переадресация в обратном направлении при групповой передаче**

**данных**

При широковещательной передаче в обратном направлении он пересылал бы пакеты маршрутизатору Ж, несмотря на то, что у Ж нет примыкающих хостов, которые относились бы к подмножеству групповой рассылки. В данном сценарии это обстоятельство не до­ставляет неудобств, так как «ниже по течению» от Г располагается всего один маршрутизатор — Ж, но можете себе представить, что случилось бы, если бы насчитывались тысячи таких маршрутиза­торов! Каждый из них получал бы ненужные пакеты, относящиеся к групповой рассылке.

Этот сценарий вовсе не так фантастичен, как это может показаться. Первая глобальная сеть групповой рассылки Mbone вначале страда­ла именно таким недостатком!)

Решение проблемы с доставкой ненужных групповых пакетов в ал­горитме RPF называют **отсечением.** Маршрутизатор, получающий групповые пакеты, у которого нет соединенных с ним напрямую хостов, принадлежащих этой группе, посылает отсекающее сообщение марш­рутизатору «выше по течению». Если маршрутизатор получает отсе­кающие сообщения от всех своих маршрутизаторов «ниже по течению», тогда он также может послать отсекающее сообщение маршрутизатору «выше по течению».

**Групповая маршрутизация в Интернете**

Первым протоколом групповой маршрутизации, использовав­шимся в Интернете, был **DVMRP** (Distance-Vector Multicast Routing Protocol) — **протокол дистанционно-векторной групповой маршру­тизации.** В протоколе DVMRP применяются деревья для каждого источника с переадресацией в обратном направлении и отсечением. В DVMRP используется алгоритм RPF с отсечением, рассмотренный выше. Вероятно, самым широко используемым **протоколом группо­вой маршрутизации** в Интернете является **PIM** (Protocol-Independent Multicast, **протоколо-независимая групповая маршрутизация),** в кото­ром явно различается два сценария распространения многоадресной ин­формации. В уплотненном режиме члены группы многоадресной пе­редачи данных расположены поблизости друг от друга; таким образом, при передаче групповых дейтаграмм задействуются многие маршрути­заторы в этой зоне или большинство из них. В уплотненном режиме ме­ханизм работы PIM напоминает лавинную рассылку и обратную пере­дачу данных, то есть, принципиально похож на функционал DVMRP.

В разреженном режиме количество маршрутизаторов, к которым подключены члены группы, сравнительно невелико по сравнению с об­щим количеством маршрутизаторов. Члены группы достаточно широко разбросаны. В разреженном режиме протокол PIM использует точки встречи для построения дерева, по которому осуществляется распреде­ленная групповая маршрутизация. При **групповой маршрутизации, зависящей от отправителя** (source-specific multicast, SSM) лишь один отправитель может отсылать трафик в дерево групповой маршрутиза­ции, что существенно упрощает построение и техническую поддержку дерева.

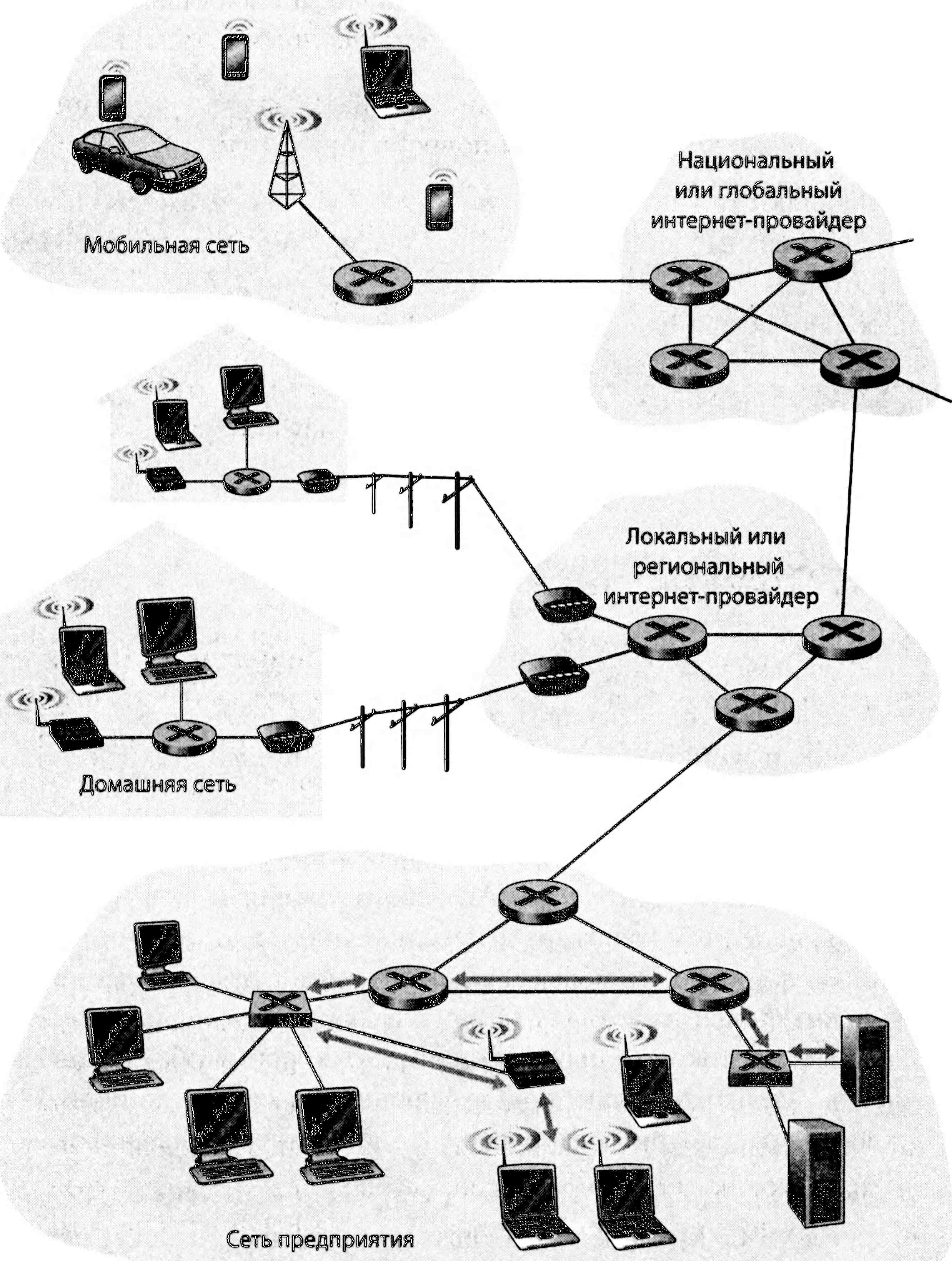
Когда в некоторой зоне одновременно используются протоколы PIM и DVMRP, администратор вычислительной сети может сконфигу­рировать IP-маршрутизаторы групповой рассылки внутри этой зоны во многом так же, как обычно конфигурируются протоколы одноадресной рассылки — RIP, IS-IS и OSPE Но что делать, если групповая маршрути­зация должна осуществляться между разными зонами? Существует ли групповой эквивалент внутреннего протокола BGP? Ответ — да. Стан­дарт RFC 4271 описывает многопротокольные расширения BGP, по­зволяющие передавать по этому нему маршрутную информацию от дру­гих протоколов, в том числе — связанную с групповой маршрутизацией. MSDP (протокол обнаружения источников группового вещания)может использоваться для соединения точек встречи в различных зо­нах, использующих разреженный режим работы с протоколом PIM. От­личный обзор актуального состояния групповой маршрутизации в Ин­тернете описан в стандарте RFC 5110.

Завершая обсуждение групповой маршрутизации в IP-сетях, следует отметить, что она пока не получила широкого распространения. Инте­ресные соображения о модели групповой маршрутизации в Интернете и о проблемах, связанных с ее развертыванием, высказываются в рабо­тах Диота и Шарма. Тем не менее, несмотря на отсутствие широкого распространения, слухи о смерти групповой маршрутизации на сетевом уровне сильно преувеличены. Уже много лет групповой трафик пере­дается в Интернете-2 и подобных сетях. Так, в Великобритании ком­пания ВВС в экспериментальном режиме распространяет контент при помощи группового вещания по IP. В то же время, групповая марш­рутизация на прикладном уровне, говоря о технологии PPLive и других одноранговых системах, напри­мер, End System Multicast обеспечивает групповое распространение содержимого среди равноправных узлов по протоколам прикладного, а не сетевого уровня. Сегодня распространение контента в одноранго­вых сетях пользуется огромной популярностью, благодаря чему можно сделать вывод, что, по крайней мере, в ближайшем будущем групповая маршрутизация будет тяготеть к прикладному уровню. Тем не менее, групповая IP-маршрутизация продолжает развиваться. Рано или позд­но эти состязания должны привести к стабильному решению.

**Обзор канального уровня**

Начнем изучение канального уровня с терминологии. Будем именовать узлом любое устройство, которое использует при работе протокол канального уровня (уровня 2). Различные виды узлов — это хосты, маршрутизаторы, коммутаторы и точки доступа Wi-Fi. Кроме того, мы будем называть каналами линии связи, которые связывают смежные узлы вдоль линии передачи данных. Чтобы дейтаграмма была передана с исходного на ко­нечный хост, ее нужно переслать по всем *отдельным каналам* полного пути от хоста к хосту. Допустим, в сети предприятия, изображенной на рис. 5.1, нужно передать дейтаграмму с одного из беспроводных хостов на один из серверов. В таком случае, данная дейтаграмма пройдет по шести каналам: канал Wi-Fi между исходным хостом и точкой доступа; канал Ethernet между точкой доступа и коммутатором канального уров­ня; канал между коммутатором канального уровня и маршрутизатором; канал между двумя маршрутизаторами; канал Ethernet между маршру­тизатором и коммутатором канального уровня; наконец, канал Ethernet между коммутатором и сервером. В каждом из этих каналов дейтаграм­ма инкапсулируется в кадр канального уровня, который и переносит кадр по каналу.

Чтобы лучше понимать, как работает канальный уровень и как он взаимодействует с сетевым, приведем аналогию с транспортом. Пред­ставьте себе агента бюро путешествий, планирующего тур из Принсто-на, штат Нью-Джерси, в Лозанну, Швейцария.



**Рис. 5.1. Шесть переходов на канальном уровне между беспроводным хостом**

**и сервером**

Предположим, турагент решает, что туристам удобнее всего отпра­виться из Принстона в аэропорт Кеннеди на такси, откуда на самолете перелететь в Женеву и, наконец, добраться до Лозанны на поезде. Тур­оператор делает три заказа, после чего за доставку туриста из Принстона в аэропорт отвечает Принстонский таксопарк, за доставку туриста из аэропорта JFK в Женеву ответственность несет авиакомпания, а за до­ставку туриста из Женевы в Лозанну — Швейцарская железнодорожная служба. Каждый из трех сегментов маршрута «напрямую» связывает два «соседних» узла. Обратите внимание, что тремя сегментами марш­рута управляют три разные компании и на этих сегментах используются совершенно разные виды транспорта (такси, самолет и поезд). Несмо­тря на это каждый сегмент предоставляет основную службу, обеспечи­вающую перемещение пассажиров в соседний узел. В данной аналогии турист соответствует дейтаграмме, каждый сегмент — линии связи, вид транспорта — протоколу канального уровня, а турагент, планирующий весь маршрут, — протоколу маршрутизации.

**Службы канального уровня**

Хотя основная услуга любого канального уровня заключается в «пе­ремещении» дейтаграммы между двумя узлами по одной линии связи, полный перечень услуг зависит от конкретного протокола канального уровня, используемого на данной линии связи. К числу таких служб от­носятся следующие:

* *Формирование кадра.* Почти все протоколы канального уровня по­мещают каждую дейтаграмму сетевого уровня в кадр канального уровня, перед тем как отправить ее по каналу. Кадр состоит из поля данных, в которое помещается дейтаграмма сетевого уровня, и не­скольких полей заголовка. Структура кадра специфицируется ка­нальным протоколом передачи данных. Во время обсуждения кон­кретных протоколов канального уровня во второй половине этой главы мы рассмотрим несколько различных форматов кадров.
* *Доступ к каналу связи.* Протокол управления доступом к среде пе­редачи (MAC, Media Access Control) определяет правила передачи кадра в канал. Для двухточечных каналов с единственным отправи­телем на одном конце и единственным получателем на другом про­токол MAC очень прост (или вообще отсутствует) — отправитель может передать кадр в любой момент, когда линия свободна. Более интересный случай представляет конфигурация, в которой несколь­ко узлов совместно используют один широковещательный канал. В этом случае возникает так называемая проблема множественного доступа, и протокол MAC призван координировать передачу кадров многих узлов.
* *Надежная доставка.* Когда протокол канального уровня предостав­ляет услугу по надежной доставке, он гарантирует перемещение каждой дейтаграммы сетевого уровня по линии связи без ошибок. Вспомним, что некоторые протоколы транспортного уровня (как, например, TCP) также обеспечивают надежную доставку. Как и на транспортном уровне, служба надежной доставки канально­го уровня поддерживается с помощью механизмов подтверждений и повторных передач (см. раздел 3.4). Служба надежной доставки транспортного уровня часто обслуживает линии связи с высокой вероятностью ошибок, например беспроводные. Таким образом, на канальном уровне ошибки исправляются локально — на том канале, где они возникают, что позволяет отказаться от повторной передачи данных протоколом транспортного или прикладного уровня. Одна­ко в линиях с низкой вероятностью ошибок надежная доставка на канальном уровне может оказаться излишней. К таким линиям от­носятся волоконно-оптические и экранированные кабели, а также различные категории линий типа «витая пара». Поэтому многие протоколы для кабельных линий не предоставляют отдельной услу­ги по надежной доставке.
* *Обнаружение и исправление ошибок.* Принимающий узел может не­вернопосчитать, что значение бита в кадре равно нулю, в то время как передавалась единица, и наоборот. Подобные битовые ошибки вызываются ослаблением сигнала и электромагнитными помехами. Поскольку нет смысла передавать дальше дейтаграмму, содержа­щую ошибки, многие протоколы канального уровня предоставля­ют услугу по обнаружению ошибок в кадре. Для этого передающий узел добавляет к кадру биты обнаружения ошибок, а получающий узел выполняет проверку контрольной суммы. Как было показано в главах 3 и 4, транспортный и сетевой уровни в Интернете также предоставляют ограниченную услугу по обнаружению ошибок. На канальном уровне обнаружение ошибок сложнее и, как правило, реализуется аппаратно. Исправление ошибок подобно их обна­ружению, за тем исключением, что получатель не только находит ошибки в кадре, но и определяет, где именно они произошли, а за­тем исправляет их.