Министерство образования и науки Российской Федерации

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Приоритетный национальный проект «Образование»
Национальный исследовательский университет

В.О. РАШИЧ В.М. ФУРТИКОВ А.В. РАШИЧ

СЕТЕВЫЕ ТЕЛЕКОММУНИКАЦИОННЫЕ ПРОТОКОЛЫ

Санкт-Петербург
Издательство политехнического университета
2011

Министерство образования и науки Российской Федерации

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Приоритетный национальный проект «Образование»
Национальный исследовательский университет

В.О. РАШИЧ В.М. ФУРТИКОВ А.В. РАШИЧ

СЕТЕВЫЕ ТЕЛЕКОММУНИКАЦИОННЫЕ ПРОТОКОЛЫ

Санкт-Петербург
Издательство политехнического университета
2011

УДК 004.7:621.39(075.8)
ББК 32.973.202:32.88я73
Р 28

Рецензент:

Кафедра военных телекоммуникационных систем Санкт-Петербургской Военной академии связи им. С.М. Буденного, д.т.н., проф. *С. Одоевский*

Рашич В.О. **Сетевые телекоммуникационные протоколы :** учеб. пособие / В. О. Рашич, В. М. Фуртиков, А.В. Рашич. — СПб.: Изд-во Политехн. ун-та, 2011. — 223 с.

ISBN

Предлагаемое учебное пособие посвящено изложению принципов построения современных телекоммуникационных сетей. Рассматриваются вопросы топологии сетей, маршрутизации и фрагментации пользовательских пакетов данных, а также особенности и характеристики наиболее популярных современных сетевых технологий. Большое внимание уделено особенностями наиболее популярного стека протоколов – TCP/IP, а также других стеков, лежащих в основе многих глобальных сетей.

Подробно изучаются принципы и протоколы маршрутизации сообщений в локальных и глобальных сетях, форматы пакетов, передаваемых на различных уровнях и в различных сетевых узлах.

Рекомендовано для студентов, обучающихся по направлениям подготовки «Инфокоммуникационные технологии и системы связи», «Радиотехника».

Печатается по решению редакционно-издательского совета Санкт-петербургского государственного политехнического университета.

© Рашич. В.О., Фуртиков В.М., Рашич А.В., 2011
© Санкт-Петербургский государственный политехнический университет, 2011

ISBN

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

В течение последнего десятилетия определяющей тенденцией в области построения телекоммуникационных сетей явился переход от технологий коммутации каналов к технологиям коммутации пакетов, на основе которых создаются современные мультисервисные сети, обеспечивающие передачу мультимедийной информации и предоставление постоянно расширяющегося набора телекоммуникационных услуг.

В настоящее время высокоскоростные пакетные технологии прочно заняли доминирующие позиции на рынке локальных и глобальных сетей. Так, в локальных сетях скорости передачи возросли до 1 Гбит/с, реализуемые при применении технологии Gigabit Ethernet. В глобальных сетях широкое распространение получил целый набор высокоскоростных пакетных технологий, таких, как 10 и 40 Gigabit Ethernet, многопротокольная коммутация по меткам (MPLS), Frame Relay, ATM и др. При этом главенствующее положение принадлежит протоколу IP, который фактически стал основным протоколом транспортных сетей, применяемым в стеке с перечисленными выше, а также другими технологиями и протоколами. Ближайшие годы будут связаны с переходом от протокола IP версии 4 к версии 6 (IPv6), что позволит существенно расширить возможности телекоммуникационных сетей по объемам, качеству и безопасности передачи мультимедийной информации.

Учебное пособие состоит из 6 глав.

В первой главе обсуждаются наиболее общие принципы построения телекоммуникационных сетей, дается их классификация и базовые понятия в области телекоммуникаций. Кратко рассмотрены вопросы эволюции телекоммуникационных сетей и тенденции их дальнейшего развития в направлении сетей следующего поколения.

Во второй главе раскрываются принципы многоуровневой организации сетей, эталонная модель взаимодействия открытых систем (ЭМВОС) и модель стека протоколов TCP/IP.

Третья глава посвящена технологиям канального уровня, используемых для построения локальных сетей, таких, как Ethernet, Token Ring, FDDI. Здесь же рассматриваются технологии беспроводных сетей и высокоскоростной 10 Gigabit Ethernet.

В четвертой главе изучаются протоколы сетевого и транспортного уровней ЭМВОС. Особое внимание уделено протоколу IPv6 и способам перехода от IPv4 к IPv6.

Пятая глава посвящена обсуждению принципов маршрутизации в IP-сетях, алгоритмов вычисления кратчайших путей и протоколов внутренней и внешней маршрутизации. Детально рассматриваются алгоритмы поиска кратчайших путей Дейкстры и Беллмана-Форда, а также протоколы маршрутизации RIP, OSPF и BGP.

В шестой главе описывается технология многопротокольной коммутации по меткам (MPLS), использующая технику виртуальных каналов для обеспечения надежной доставки сообщений.

Учебное пособие составлено преподавателями кафедры «Радиоэлектронные средства защиты информации» по материалам курсов лекций, читаемых авторами на радиофизическом факультете Санкт-Петербургского государственного политехнического университета и предназначено для студентов, обучающихся по направлениям «Инфокоммуникационные технологии и системы связи», «Радиотехника». Оно также может оказаться полезным для слушателей образовательных учреждений дополнительного профессионального образования.

1. КЛАССИФИКАЦИЯ И НАПРАВЛЕНИЯ РАЗВИТИЯ ТЕЛЕКОММУНИКАЦИОННЫХ СЕТЕЙ

Современный мир телекоммуникационных сетей очень многообразен и постоянно развивается. Несмотря на то, что телекоммуникационные сети (ТКС) обладают многими общими свойствами, они имеют также ряд отличительных черт, которые часто оказываются определяющими для понимания процессов создания, функционирования и модернизации ТКС. Поэтому в данной главе последовательно рассматриваются классификация ТКС и основные направления их развития [1, 6, 7, 8].

1.1. КЛАССИФИКАЦИЯ ТЕЛЕКОММУНИКАЦИОННЫХ СЕТЕЙ. ОСНОВНЫЕ ОПРЕДЕЛЕНИЯ

Принятая в настоящее время классификация телекоммуникационных сетей (ТКС) Единой сети электросвязи Российской Федерации [7] в соответствии с различными признаками приведена на рис. 1.1. Не останавливаясь подробно на всех уровнях данной классификации, выделим некоторые позиции, важные для понимания современных тенденций в развитии телекоммуникационных сетей.

1. Деление сетей по категориям.
2. Традиционное деление сетей на первичные и вторичные.
3. Современное деление сетей на транспортные сети и сети доступа.
4. Моносервисные и мультисервисные сети.
5. Сети с коммутацией каналов и с коммутацией пакетов.

Единая сеть электросвязи (ЕСЭ) Российской Федерации состоит из расположенных на территории Российской Федерации сетей электросвязи следующих категорий:

- сети связи общего пользования;
- выделенные сети связи;
- технологические сети связи, присоединенные к сети связи общего пользования;
- сети связи специального назначения.

Сеть связи общего пользования (ОП) предназначена для предоставления услуг электросвязи любому пользователю на территории Российской Федерации без каких либо ограничений.

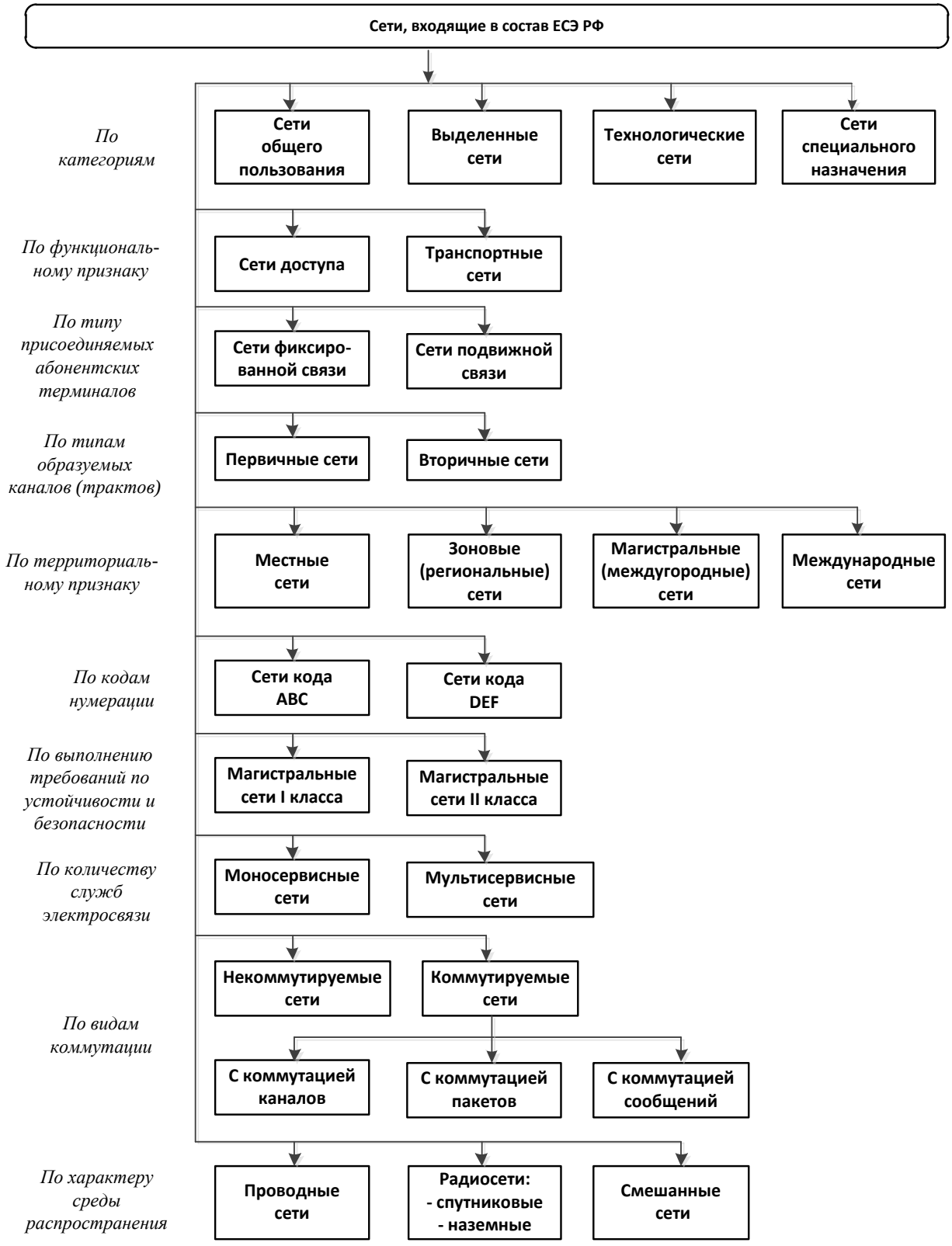


Рис. 1.1. Классификация ТКС

Сеть связи ОП представляет собой комплекс взаимодействующих сетей связи, включая сети связи для распространения программ телевизионного и радиовещания. Ответственность за функционирование и развитие сети связи общего пользования возлагается на федеральные органы исполнительной власти в области связи.

Выделенные, технологические, а также сети связи специального назначения образуют группу сетей ограниченного пользования (ОгП), так как контингент их пользователей ограничен корпоративными клиентами. Сети ограниченного пользования — составная часть ЕСЭ, которые функционируют с ограничением на предоставление услуг абонентам.

Выделенные сети связи — это сети, предназначенные для предоставления услуг ограниченному кругу пользователей. Такие сети могут взаимодействовать между собой, но не имеют присоединения к сетям общего пользования ЕСЭ, а также к сетям связи общего пользования иностранных государств. Выделенная сеть может быть присоединена к сети общего пользования ЕСЭ с переводом в категорию сети общего пользования, если она соответствует ее требованиям.

Технологические сети связи предназначены для обеспечения производственной деятельности организаций и управления технологическими процессами. При наличии свободных ресурсов эти сетевые ресурсы могут быть присоединены к сети общего пользования ЕСЭ с переводом в категорию сетей общего пользования и использованы для предоставления возмездных услуг любому пользователю.

Сети связи специального назначения предназначены для обеспечения нужд государственного управления, обороны, безопасности и охраны правопорядка в Российской Федерации. Такие сети не могут использоваться для возмездного оказания услуг связи, если иное не предусмотрено законодательством Российской Федерации.

Для традиционных ТКС характерно строгое разделение на первичные и вторичные сети.

Первичные сети составляли основу транспортных сетей и представляют собой совокупность типовых каналов и трактов передачи, образованных оборудованием сетевых узлов (станций) и соединяющих их линий передачи (или физических цепей), соединяющих эти узлы.

Сетевая станция — это комплекс технических средств (КТС), обеспечивающий образование и предоставление вторичным сетям связи типовых каналов передачи и сетевых трактов, а также их транзит.

Сетевой узел — это КТС, обеспечивающий соединение сетевых станций, образование и перераспределение сетевых трактов, типовых каналов передачи, а также предоставление их вторичным сетям.

Традиционные и современные первичные сети строятся на основе технологий PDH, SDH, NG-SDH, WDM (DWDM). Они предоставляют типовые каналы передачи во вторичные сети для образования каналов связи. Типовые каналы и тракты передачи систем плезиохронной (PDH) и синхронной цифровых иерархий представлены на рис. 1.2 и 1.3.

Наименование	Скорость передачи (Кбит/с)
ОЦК	$64 \times (1 \pm 100 \cdot 10^{-6})$
Первичный ЦТ, Е1	$2048 \times (1 \pm 50 \cdot 10^{-6})$
Вторичный ЦТ, Е2	$8448 \times (1 \pm 30 \cdot 10^{-6})$
Третичный ЦТ, Е3	$34368 \times (1 \pm 20 \cdot 10^{-6})$
Четверичный ЦТ, Е4	$139264 \times (1 \pm 15 \cdot 10^{-6})$

Рис. 1.2. Типовые каналы и тракты PDH

Наименование	Скорость передачи (Мбит/с)
STM-1	155,520
STM-4	622,080
STM-16	2487,320
STM-64	9953,280
STM-256	39813,120

Рис. 1.3. Типовые каналы и тракты SDH

Первичная сеть общего пользования строится по иерархическому принципу, охватывает всю страну и состоит из:

- магистральной первичной сети (СМП);
- внутризоновых первичных сетей (ВзПС);
- местных первичных сетей (МПС).

Магистральная первичная сеть (ПСМ) — часть первичной сети, обеспечивающая соединение между собой типовых каналов передачи и се-

тевых трактов разных внутризоновых первичных сетей на всей территории страны. Структура ПСМ — комбинированная, сочетающая решетчатую и радиально-узловую структуры. Каждый узел ПСМ должен иметь не менее трех выходов на другие сетевые узлы с тем, чтобы была возможность организации любой связи не менее, чем по трем независимым путям. Протяженность ПСМ составляет 12500 км.

Внутризоновая первичная сеть (ВзПС) — часть первичной сети, обеспечивающая соединение между собой типовых каналов передачи разных местных первичных сетей одной зоны нумерации телефонной сети. Их имеется в стране 76. ВзПС строятся таким образом, чтобы имелось бы не менее двух независимых путей составления каналов и сетевых трактов между каждым внутризоновым узлом (областным, краевым, республиканским центром) и каждой внутризоновой сетевой станцией. Первоначально ВзПС строится по радиально-узловому принципу, а в дальнейшем дополняется линиями передачи. Протяженность ВзПС составляет 600 км.

Местная первичная сеть (МПС) — часть первичной сети, ограниченная территорией города с пригородом или сельского района. Местной первичной сети присваивают названия: городская (комбинированная) или сельская первичная сеть. Протяженность МПС составляет 100 км.

Структура МПС определяется, главным образом, особенностями конкретного города или сельского района. Как правило, МПС строятся по радиально-узловому принципу.

Совокупность внутризоновой и местных первичных сетей в пределах области, края, округа на территории, совпадающей с зоной нумерации, представляет собой зоновую первичную сеть (800 км).

Вторичная сеть представляет собой совокупность каналов связи, образуемых на базе первичной сети путем их коммутации (маршрутизации) в узлах коммутации и организации связи между абонентскими устройствами пользователей.

Станция вторичной сети — это КТС, обеспечивающий соединение (коммутацию) каналов вторичной сети как между собой, так и с оконечными устройствами. Различают станции коммутации каналов, сообщений, пакетов, станции переключений (кроссы) и др.

Узел вторичной сети — это КТС, обеспечивающий соединение станций вторичной сети между собой. Различают узлы коммутации (каналов, сообщений, пакетов) и узлы переключений (кроссовые узлы).

На основе каналов и трактов передачи, образуемых первичной сетью создавались те или иные вторичные сети (как правило, моносервисные): телефонные сети, сети передачи данных и т. д.

Необходимо отметить, что в существовавших до появления и бурного развития интернета сетях связи преобладал речевой трафик, и поэтому их развитие происходило по пути совершенствования технологий, основанных на технике коммутации каналов. Таким образом, в традиционных телекоммуникационных сетях основным принципом работы был принцип коммутации каналов.

Однако в последние годы существенно изменилось соотношение между пакетным трафиком и телефонным в сторону увеличения первого. При этом существующие первичные сети с позиций эффективного использования ресурсов и обеспечения требуемых характеристик транспортной сети оказались не оптимизированы под передачу пакетного трафика.

В связи с постоянным расширением набора предоставляемых пользователям услуг существующие вторичные сети также оказались не приспособленными к их реализации без кардинальных архитектурных изменений в направлении создания интегрированных вторичных сетей. В первую очередь это касается вопросов эффективного обслуживания разнородного мультимедийного трафика. Само по себе существование различных вторичных сетей (телефонных, передачи данных и т. п.) при необходимости предоставления пользователям интегрированных услуг противоречило принципу оптимального использования ресурсов и вело к усложнению оборудования вторичных сетей и управления ими.

В соответствии с принятыми в настоящее время воззрениями сеть связи (или телекоммуникационная сеть) — это технологическая система, которая состоит из линий и каналов связи, узлов, оконечных станций, и предназначена для обеспечения пользователей электрической связью с помощью абонентских терминалов, подключаемых к оконечным станциям (рис. 1.4).

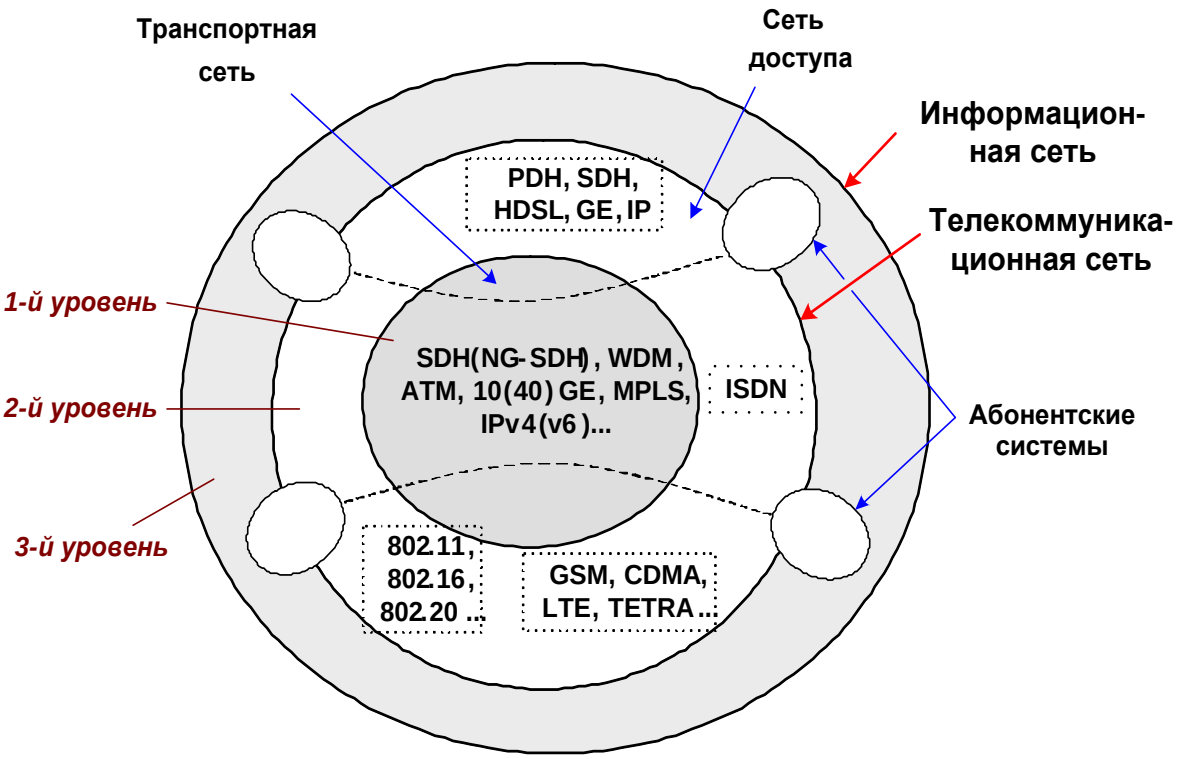


Рис. 1.4. Концептуальная модель инфокоммуникационной сети

Информационная сеть (ИС) — это технологическая система, представляющая собой функционально связанную совокупность программно-технических средств обработки и обмена информацией и состоящая из территориально распределенных информационных узлов (подсистем обработки информации) и каналов передачи информации, соединяющих данные узлы.

Современные телекоммуникационные сети (ТКС) отличаются большим многообразием с точки зрения их назначения, предоставляемых ими услуг и архитектурного построения.

Телекоммуникационная сеть охватывает два первых уровня концептуальной модели инфокоммуникационной сети. Она включает в себя *транспортное ядро* и *сети доступа* (рис. 1.4).

Сетью доступа (СД) ТКС является та ее часть, которая связывает абонентские терминалы с узлом доступа, являющимся граничным между сетью доступа и транспортной сетью. Сеть доступа обеспечивает подключение пользовательского оборудования через его абонентские линии

и / или объектовые сети к узлу доступа и предоставление ему услуг доступа к службам транспортной сети.

Таким образом, ТКС включает в себя следующие элементы:

- терминальное оборудование пользователей;
- сети доступа;
- транспортные сети различных территориальных уровней.

Применение принципа разделения сетей на транспортные сети и сети доступа при их проектировании и эксплуатации обусловлено необходимостью разделения функций транспортировки (переноса) агрегированных потоков и предоставления услуг пользователям в целях оптимизации ТКС (уменьшения стоимости создания и эксплуатации, эффективного использования ресурсов, обеспечения инвариантности к размещению пользователей, объему и номенклатуре предоставляемых им услуг, обеспечения возможности относительно независимого развития и т. п.). Указанный принцип реализуется как в традиционных сетях, так и в сетях следующего поколения (NGN — Next Generation Network), несмотря на кардинальные отличия их между собой.

1.2. ЭВОЛЮЦИЯ ТЕЛЕКОММУНИКАЦИОННЫХ СЕТЕЙ И НАПРАВЛЕНИЯ ИХ РАЗВИТИЯ

Эволюция телекоммуникационных сетей с момента их зарождения (рис. 1.5) происходит, начиная с создания специализированных для передачи каждого вида информации сетей с коммутации каналов, в направлении мультисервисных сетей, предназначенных для передачи различных видов информации, и далее — к сетям следующего поколения.

Традиционные телекоммуникационные сети базировались на технологии коммутации каналов, оптимальной для передачи преобладавшего ранее телефонного трафика. Транспортной основой их являются первичные сети, использующие технологии плезиохронной и синхронной цифровой иерархии. На узлах цифровой первичной сети размещается оборудование мультиплексирования / демультиплексирования, использующее технологию временно́го объединения каналов (TDM) и предназначенное для формирования типовых цифровых каналов и трактов передачи.

На базе типовых каналов и трактов первичной сети развертываются вторичные сети, специализированные для передачи различных видов информации: телефонные, телеграфные и др.

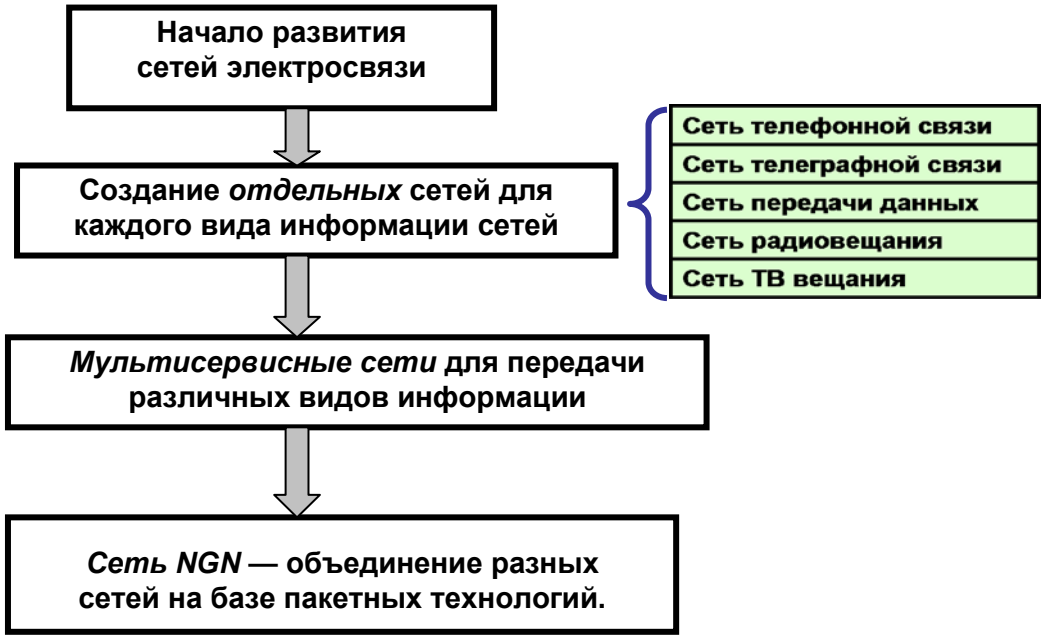


Рис. 1.5. Эволюция ТКС

Бурное развитие компьютерных технологий и тотальная «интернетизация» общества привели к росту трафика передачи данных и к расширению перечня услуг, предоставляемых пользователям. Первое (рост трафика данных) способствовало развитию пакетных технологий, так как сети с коммутацией каналов не оптимизированы под передачу трафика данных.

Второе (необходимость предоставления пользователю множества услуг помимо телефонии) привело к интеграции моносервисных вторичных сетей и породило появление концепций и технологий мультисервисных сетей, обеспечивающих передачу мультимедийного трафика.

Развитие пакетных технологий и концепций мультисервисных сетей изменило представление о классификации сетей на первичные и вторичные. Как было указано выше, в настоящее время телекоммуникационная сеть представляется в виде транспортной сети, выполняющей функции переноса, и сетей доступа, выполняющих функции предоставления услуг пользователям и доступа в транспортную сеть.

Следующим этапом развития телекоммуникаций, который происходит в настоящее время, является движение в сторону построения сетей следующего поколения. Сети NGN — это не просто объединенные пакетные сети, а сети, которые характеризуются новой архитектурой, новыми свойствами и возможностями.

Движение в сторону NGN происходило и происходит под очень сильным влиянием интернета и получивших глобальное распространение технологий беспроводной мобильной связи. При этом можно выделить следующие основные тенденции в развитии телекоммуникационных сетей:

- глобализация;
- пакетизация;
- мультисервисность;
- широкополосность;
- интеграция;
- конвергенция;
- персонализация;
- универсальная мобильность;
- гарантия качества и масштабируемость предоставляемых услуг.

Глобализация означает, что современные и перспективные ТКС обеспечивают или будут обеспечивать глобальную связь, что ранее было доступно только спутниковым системам. Это достигается за счет развертывания высокоскоростных транспортных сетей на основе волоконно-оптических систем передачи (SDH, DWDM) с дальнейшим переходом к полностью оптическим сетям при комплексном применении высокоскоростных технологий построения сетей доступа (в том числе, беспроводных сетей, сетей сотовой и спутниковой связи).

Пакетизация знаменует окончательную победу передачи всех видов трафика (речь, видео, данные) в пакетном формате и полный переход к пакетным технологиям.

Под *мультисервисностью* понимают возможность одновременного предоставления множества услуг по передаче мультимедийного трафика.

Широкополосность обусловлена необходимостью предоставления постоянно расширяющегося набора многообразных услуг с гарантированным качеством, что приводит к неуклонному возрастанию скоростей передачи информации и созданию все более высокоскоростных технологий.

Конвергенция — это сближение и взаимопроникновение сетей, технологий, услуг. Примером конвергенции может служить происходящая в настоящее время конвергенция сетей фиксированной и подвижной (мобильной) связи.

Интеграция связана с обеспечением возможности совместного функционирования различных технологий, сетей в любых сочетаниях и обеспечение любого спектра услуг.

Персонализация означает смещение акцентов в сторону обеспечения связи с конкретным человеком, независимо от места его нахождения и времени, а не с устройством, и обеспечение ему всего спектра необходимых услуг.

Наконец, *универсальная мобильность* — это обеспечение связи в движении и на месте вне зависимости от места расположения пользователя и без привязки к определенному типу телекоммуникационного оборудования.

Именно возможность реализации перечисленных тенденций отличает сети NGN от сетей предшествующего поколения.

В самом общем виде процесс эволюции телекоммуникационных сетей представлен на рис. 1.6, где можно проследить некоторые основные тенденции технологического развития ТКС, перечисленные ранее, а также обозначить особенности их архитектурного развития.

В частности, первой попыткой на пути создания мультисервисных сетей была технология ISDN — цифровых сетей с интеграцией служб, поддерживающая передачу телефонного трафика и данных, причем для управления процессом установления соединения в ней используется техника коммутации пакетов. В классической ТфОП оборудование узлов коммутации одновременно решало задачи обработки вызовов, коммутации и предоставления услуг. В сетях ISDN, базирующихся на принципах ТфОП, функции обработки сигнализации и коммутации были частично разделены, однако задачи управления коммутацией по-прежнему решались в системах коммутации. Кроме того, не было достигнуто и решение задач оптимизации ТКС под передачу разнородного трафика и обеспечения возможности неограниченного наращивания предоставляемых пользователям услуг.

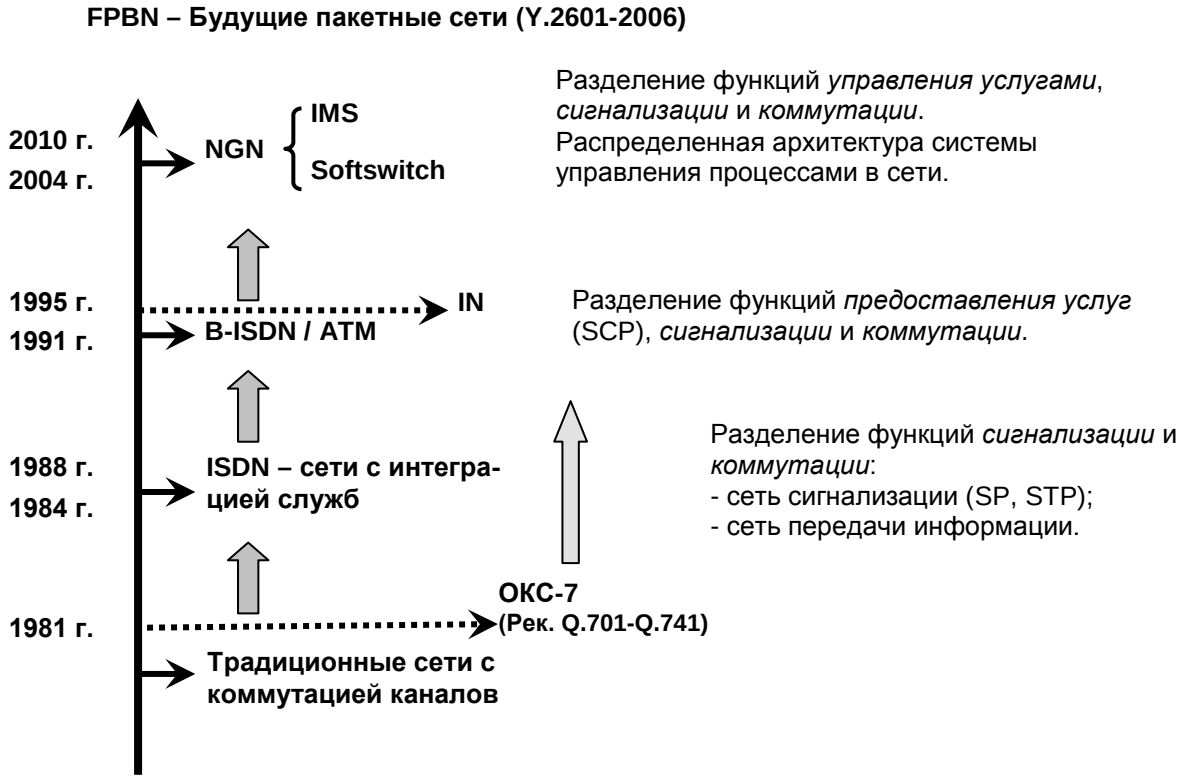


Рис. 1.6. Эволюция технологий ТКС

ISDN — это сеть с коммутацией каналов, ориентированная, прежде всего на предоставление услуг телефонии. Так, пользовательская скорость передачи данных по интерфейсу BRI не превышает 128 Кбит/с = 2×64 Кбит/с (по интерфейсу PRI — 2048 Кбит/с), что в начале 2000-х годов оказалось явно недостаточно. Кроме того, ограничен и набор услуг, предоставляемых сетью ISDN. Поэтому технология ISDN так и не стала универсальной сетевой технологией и заняла свое место в ряду технологий доступа.

Однако первый шаг в направлении перехода к пакетным сетям был сделан, и на смену ISDN пришли пакетные сети ATM или широкополосные сети с интеграцией служб (B-ISDN).

Технология асинхронного переноса (ATM) — это уже чисто пакетная технология, которая изначально создавалась и позиционировалась, как универсальная самодостаточная технология построения мультисервисных телекоммуникационных сетей. Её отличительными чертами являются:

- использование техники коммутации коротких пакетов (ячеек по 53 байта) для передачи мультимедийного трафика: речи, видео, данных;

- использование техники виртуальных каналов и путей, на базе которой появилась возможность в рамках общей сети строить виртуальные частные сети (VPN) на основе разграничения трафика;

- наличие внутренних механизмов обеспечения гарантий качества обслуживания.

Последнее достоинство технологии ATM, касающееся гарантий качества обслуживания, до сих пор является непревзойденным.

ATM предназначалась для использования и в транспортных сетях, и в сетях доступа вплоть до пользователя, т. е. претендовала на господство в мире телекоммуникационных технологий. Однако по разным причинам, среди которых называются многие и о которых до настоящего времени нет единого мнения, технология ATM оказалась вытесненной из магистральных сетей стеком технологий IP/MPLS/Ethernet.

Таким образом, телекоммуникационные сети подошли к следующему этапу развития, а именно, к сетям следующего поколения. Сети NGN должны вобрать в себя все перечисленные ранее тенденции, но, перед тем как переходить к их рассмотрению, необходимо заранее отметить одну из важнейших их особенностей: разделение функций управления услугами, сигнализацией и коммутацией, а также распределенную архитектуру управления всеми процессами в сети. Разделение функций сигнализации и коммутации впервые было реализовано в системе сигнализации № 7 (ОКС-7). В дальнейшем с появлением интеллектуальных сетей (IN), сердцевиной которых являлась система сигнализации ОКС-7, в ней появились пункты предоставления услуг (SCP). Окончательно и в полном объеме разделение функций предоставления услуг, сигнализации и коммутации предусмотрено в сетях следующего поколения.

В итоге в 2003–2005 гг. сложились условия для реализации концепции сетей следующего поколения, а именно, имелись хорошо отработанные высокоскоростные пакетные технологии, возможности которых стимулировали рост потребностей пользователей в новых услугах.

Наиболее популярными технологиями построения транспортных сетей в настоящее время являются (рис. 1.4):

- синхронная цифровая иерархия SDH (NG-SDH);

- технология спектрального мультиплексирования (DWDM);

- IP/MPLS (MPLS — многопротокольная коммутация по меткам);

- технология асинхронного переноса ATM;
- 10/40 Gigabit Ethernet.

Принципиальным вопросом в развитии транспортных сетей является вопрос выбора единого транспорта — технологии (протокола), обеспечивающего передачу любого трафика разнородных сетей доступа в едином унифицированном формате.

В IP-сетях роль такого транспортного протокола, позволяющего объединять разнородные сети (сети, построенные с применением различных технологий), выполнял протокол IP. Однако в IP-сетях отсутствовала возможность обеспечения гарантированного качества обслуживания, что допустимо для публичных сетей, но недопустимо для корпоративных сетей и тем более для сетей специального (в том числе, военного) назначения.

Важнейший недостаток IP, изначально заключающийся в невозможности обеспечить гарантированное качество обслуживания (QoS), преодолевается в настоящее время на основе разработки соответствующих механизмов QoS для IP-сетей (интегрированное и дифференцированное обслуживание), а также таких технологий, как MPLS, которые в сочетании с IP позволяют обеспечить качество обслуживания, сравнимое с ATM. В существующих ATM сетях используется в качестве транспортной технологии IP/ATM, т. е. ATM работает как протокол канального уровня.

Популярные технологии сетей доступа более многочисленны (рис. 1.4): PDH, ISDN, xDSL, VDSL, Ethernet (Fast Ethernet, Gigabit Ethernet), FDDI, FTTx/PON, Wi-Fi и WiMax и др. Отметим, что бурное развитие претерпевают как технологии проводного, так и беспроводного доступа.

Несмотря на существование, постоянное возникновение и конкуренцию множества технологий построения ТКС (как транспортных, так и сетей доступа) роль объединяющего протокола играет IP. При этом разработчики стандартов, оборудования ТКС и ведущие операторы решают важнейшие задачу обеспечения совместимости технологий, унификации, оптимизации их совместного использования в различных сочетаниях.

ВОПРОСЫ И ЗАДАНИЯ К ГЛАВЕ 1

1. Дайте классификацию сетей по различным классификационным признакам.

2. Какие основные функции реализуются транспортными сетями и сетями доступа. В чем их отличия?

3. Сравните достоинства и недостатки технологий с коммутацией каналов и коммутацией пакетов.

4. В чем отличие мультисервисных сетей от моносервисных?

5. Перечислите базовые технологии транспортных сетей.

6. Перечислите базовые технологии сетей доступа.

7. Охарактеризуйте современные тенденции развития телекоммуникационных сетей.

2. МНОГОУРОВНЕВАЯ ОРГАНИЗАЦИЯ СЕТЕЙ

Организация взаимодействия между устройствами сети является сложной задачей. Для решения сложных задач используется известный универсальный прием — *декомпозиция*, т. е. разбиение одной сложной задачи на несколько более простых задач-модулей. Декомпозиция состоит в четком определении функций каждого модуля, а также порядка их взаимодействия. При таком подходе каждый модуль можно рассматривать как «черный ящик», абстрагируясь от его внутренних механизмов и концентрируя внимание на способе взаимодействия этих модулей.

Еще более эффективной концепцией, развивающей идею декомпозиции, является *многоуровневый подход*, когда после представления исходной задачи в виде множества модулей эти модули группируют и упорядочивают по уровням, образующим иерархию. В соответствии с принципом иерархии для каждого промежуточного уровня можно указать непосредственно примыкающие к нему соседние вышележащий и нижележащий уровни.

В данной главе рассматриваются основные принципы, по которым осуществляется многоуровневая организация сетей [1, 3, 6, 8].

2.1. ПРИНЦИПЫ МНОГОУРОВНЕВОЙ ОРГАНИЗАЦИИ

Группа модулей, составляющих каждый уровень, для решения своих задач должна обращаться с запросами только к модулям соседнего нижележащего уровня. С другой стороны, результаты работы каждого из модулей, отнесенных к некоторому уровню, могут быть переданы только модулям соседнего вышележащего уровня. Такая иерархическая декомпозиция задачи (рис. 2.1) предполагает четкое определение функций и интерфейсов не только отдельных модулей, но и каждого уровня.

Межуровневый интерфейс, называемый также интерфейсом услуг, определяет набор функций, которые нижележащий уровень предоставляет вышележащему.

Многоуровневое представление средств сетевого взаимодействия имеет свою специфику, связанную с тем, что в процессе обмена сообщениями участвуют, по меньшей мере, две стороны, т. е. в данном случае необходимо организовать согласованную работу двух иерархий аппаратных и программных средств, работающих на разных компьютерах. Оба участника сетевого обмена должны при-

нять множество соглашений. Например, они должны согласовать уровни и форму электрических сигналов, способ определения размера сообщений, договориться о методах контроля достоверности и т. п. Другими словами, соглашения должны быть приняты на всех уровнях, начиная от самого низкого — уровня передачи битов — и заканчивая самым высоким, реализующим обслуживание пользователей сети.

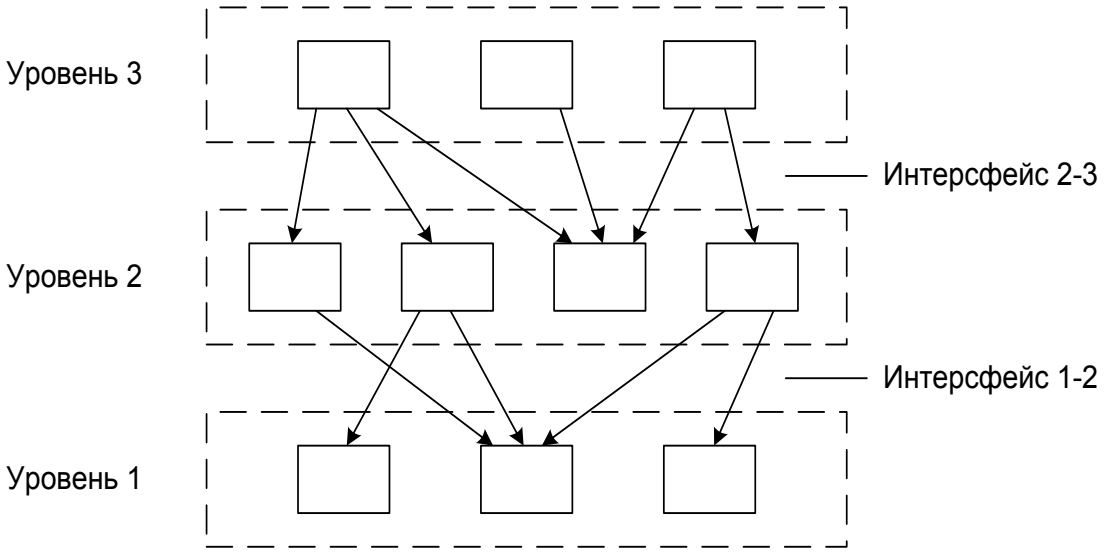


Рис. 2.1. Многоуровневый подход — создание иерархии задач

Межуровневый интерфейс**,** называемый также интерфейсом услуг**,** определяет набор функций, которые нижележащий уровень предоставляет вышележащему.

На рис. 2.2 показана модель взаимодействия двух узлов. С каждой стороны средства взаимодействия представлены четырьмя уровнями. Каждый уровень поддерживает интерфейсы двух типов. Во-первых, это интерфейсы услуг с выше- и нижележащим уровнями «своей» иерархии средств. Во-вторых, это интерфейс со средствами взаимодействия другой стороны, расположенными на том же уровне иерархии. Этот тип интерфейса называют *протоколом***.** Таким образом, протокол всегда является одноранговым интерфейсом.

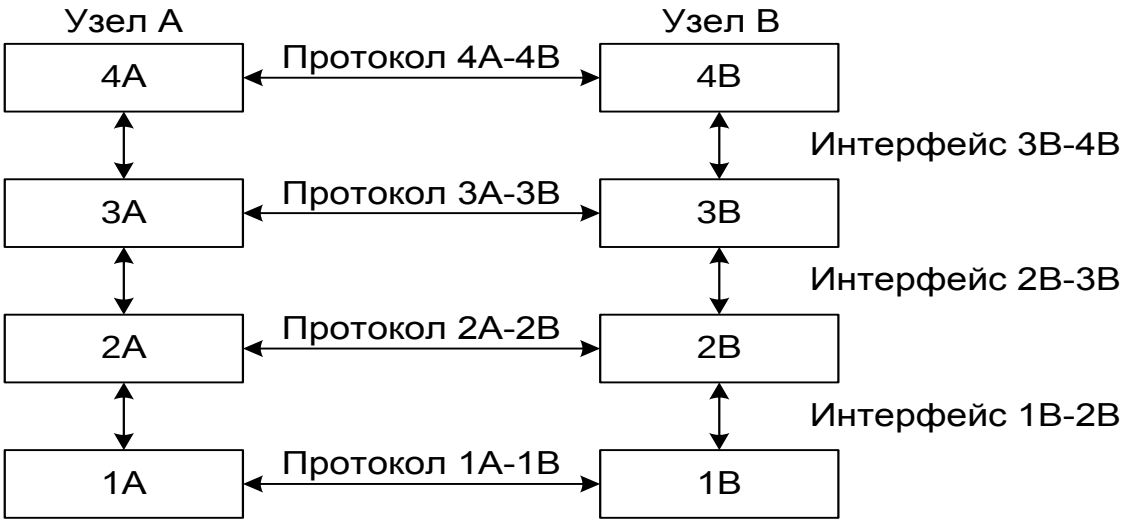


Рис. 2.2. Взаимодействие двух узлов

Иерархически организованный набор протоколов, достаточный для организации взаимодействия узлов в сети, называется *стеком протоколов*. Протоколы нижних уровней часто реализуются и программными, и аппаратными средствами, а протоколы верхних уровней, как правило, программными средствами.

2.2. МОДЕЛЬ ВЗАИМОДЕЙСТВИЯ ОТКРЫТЫХ СИСТЕМ

В начале 1980-х гг. ряд международных организаций по стандартизации, в частности International Organization for Standardization (ISO), часто называемая также International Standards Organization, а также International Telecommunications Union (ITU) и некоторые другие, разработали стандартную *модель взаимодействия открытых систем* (Open System Interconnection, OSI), которая сыграла значительную роль в развитии компьютерных сетей. Назначение модели OSI состоит в обобщенном представлении средств сетевого взаимодействия. Она разрабатывалась в качестве своего рода универсального языка сетевых специалистов, именно поэтому ее называют справочной моделью.

В модели OSI (рис. 2.3) средства взаимодействия делятся на семь уровней: прикладной, представления, сеансовый, транспортный, сетевой, канальный и физический. Каждый уровень имеет дело с совершенно определенным аспектом взаимодействия сетевых устройств.

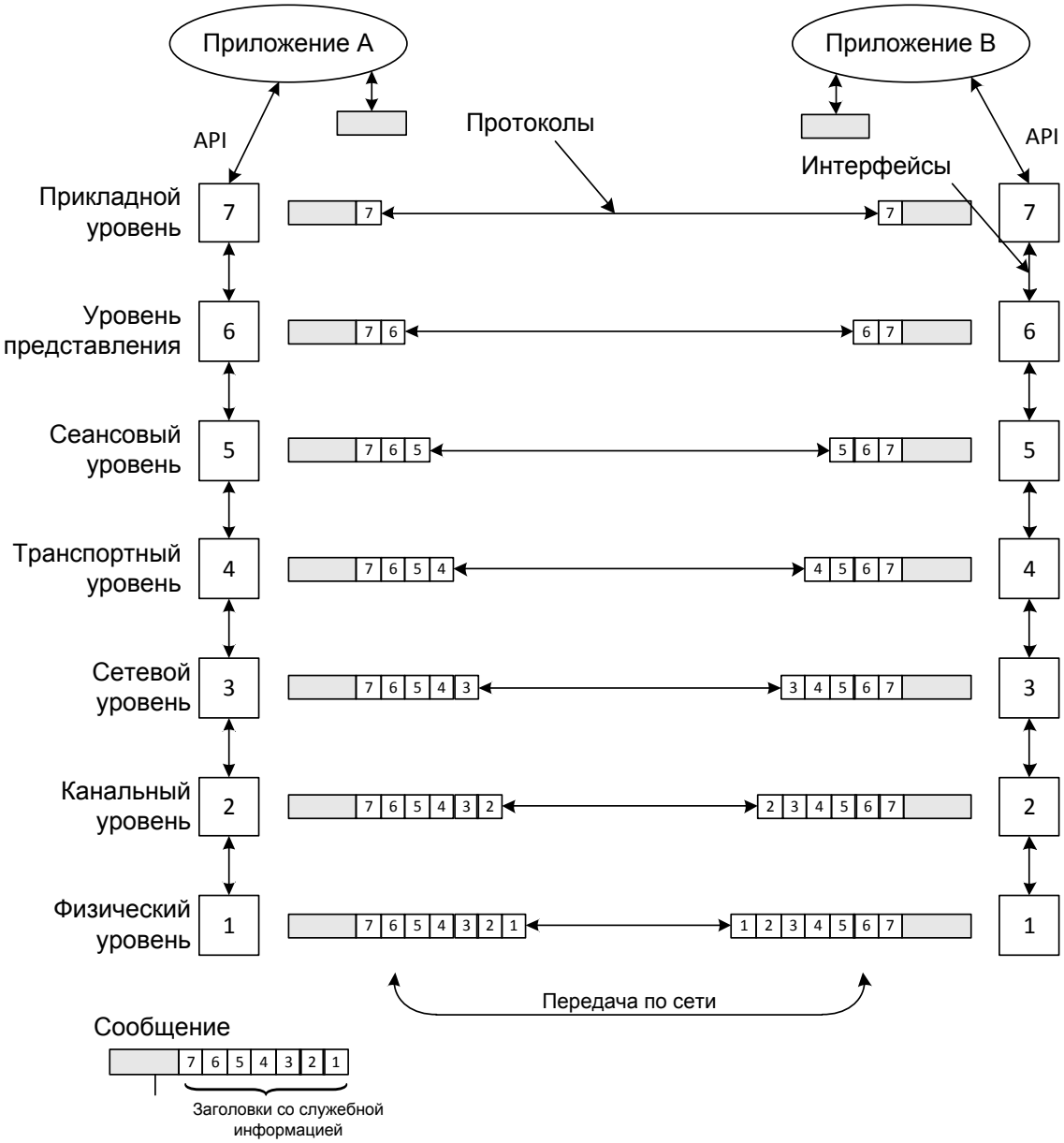


Рис. 2.3. Модель взаимодействия открытых систем OSI

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

На физическом уровне информация не разделяется по своему содержанию: представляет однородный поток битов, которые нужно доставить без искажений и в соответствии с заданными скоростью и качеством.

Канальный уровень (data link layer) является первым уровнем (если идти снизу вверх), который работает в режиме коммутации пакетов. На этом уровне протокольная единица данных (Protocol Data Unit, PDU) обычно носит название *кадр* (frame).

Функции средств канального уровня определяются по-разному для локальных и глобальных сетей. В *локальных сетях* канальный уровень должен обеспечивать доставку кадра между любыми узлами сети. При этом предполагается, что сеть имеет типовую топологию, например общую шину, кольцо, звезду или дерево (иерархическую звезду). Примерами технологий локальных сетей, применение которых ограничено типовыми топологиями, являются Ethernet, FDDI, Token Ring. В *глобальных сетях* канальный уровень должен обеспечивать доставку кадра только между двумя соседними узлами, соединенными индивидуальной линией связи. Примерами двухточечных протоколов (как часто называют такие протоколы) могут служить широко распространенные протоколы PPP и HDLC. На основе двухточечных связей могут быть построены сети произвольной топологии. Для связи локальных сетей между собой или для доставки сообщений между любыми конечными узлами глобальной сети используются средства более высокого сетевого уровня.

Одной из функций канального уровня является поддержание интерфейсов с нижележащим физическим уровнем и вышележащим сетевым уровнем. Сетевой уровень направляет канальному уровню пакет для передачи в сеть или принимает от него пакет, полученный из сети. Физический уровень используется канальным уровнем как инструмент, который принимает и передает в сеть последовательности битов.

Начнем рассмотрение работы канального уровня с момента, когда сетевой уровень отправителя передает канальному уровню пакет, а также указание на то, какому узлу его передать. Для решения этой задачи канальный уровень создает кадр, который имеет поле данных и заголовок. Канальный уровень помещает (*инкапсулирует*) пакет в поле данных кадра и заполняет соответствующей служебной информацией заголовок кадра.

Важнейшей информацией заголовка кадра является адрес назначения, на основании которого коммутаторы сети будут продвигать пакет.

Одной из задач канального уровня является *обнаружение и коррекция ошибок.* Для этого канальный уровень фиксирует границы кадра, помещая специальную последовательность битов в его начало и конец, а затем добавляет к кадру контрольную сумму, которая называется также *контрольной последовательностью кадра* (Frame Check Sequence, FCS). Контрольная сумма вычисляется по некоторому алгоритму как функция от всех байтов кадра. По значению FCS узел назначения сможет определить, были или нет искажены данные кадра в процессе передачи по сети.

Однако прежде, чем переправить кадр физическому уровню для непосредственной передачи данных в сеть, канальному уровню может потребоваться решить еще одну важную задачу. Если в сети используется разделяемая среда, то прежде, чем физический уровень начнет передавать данные, канальный уровень должен проверить *доступность* среды. Функции проверки доступности разделяемой среды иногда выделяют в отдельный подуровень управления доступом к среде (Media Access Control, MAC).

Если разделяемая среда освободилась (когда она не используется, такая проверка, конечно, пропускается), то кадр передается средствами физического уровня в сеть, проходит по каналу связи и поступает в виде последовательности битов в распоряжение физического уровня узла назначения. Этот уровень, в свою очередь, передает полученные биты «наверх» канальному уровню своего узла. Последний группирует биты в кадры, снова вычисляет контрольную сумму полученных данных и сравнивает результат с контрольной суммой, переданной в кадре. Если они совпадают, кадр считается правильным. Если же контрольные суммы не совпадают, то фиксируется ошибка. В функции канального уровня входит не только обнаружение ошибок, но и исправление их за счет повторной передачи поврежденных кадров. Однако эта функция не является обязательной, и в некоторых реализациях канального уровня она отсутствует, например в Ethernet, Token Ring, FDDI и Frame Relay.

Протоколы канального уровня реализуются компьютерами, мостами, коммутаторами и маршрутизаторами, и такие протоколы обычно работают в пределах сети, являющейся одной из составляющих более крупной составной сети, объединенной протоколами сетевого уровня. Адреса, с кото-

рыми работает протокол канального уровня, используются для доставки кадров только в пределах этой сети, а для перемещения пакетов между сетями применяются адреса следующего, сетевого уровня.

В локальных сетях канальный уровень поддерживает весьма мощный и законченный набор функций по пересылке сообщений между узлами сети. В некоторых случаях протоколы канального уровня локальных сетей оказываются самодостаточными транспортными средствами и могут допускать работу непосредственно «поверх себя» протоколов прикладного уровня или приложений без привлечения средств сетевого и транспортного уровней. Тем не менее, для качественной передачи сообщений в сетях с произвольной топологией функций канального уровня оказывается недостаточно. Это утверждение в еще большей степени справедливо для глобальных сетей, в которых протокол канального уровня реализует достаточно простую функцию передачи данных между соседними узлами.

Сетевой уровень (network layer) служит для образования единой транспортной системы, объединяющей несколько сетей, называемой *составной сетью***,** или *интернетом* (не путать с Интернетом, как самой известной реализацией составной сети, построенной на базе стека TCP/IP). Технология, позволяющая соединять в единую сеть множество сетей, в общем случае построенных на основе разных технологий, называется *технологией межсетевого взаимодействия* (internetworking). Причина необходимости такого взаимодействия очевидна и кроется в существенных отличиях одной технологии от другой. Даже наиболее близкие технологии — Ethernet, FDDI, Token Ring — имеющие одну и ту же систему адресации (адреса подуровня MAC, называемые MAC-адресами), отличаются друг от друга форматом используемых кадров и логикой работы протоколов. Еще больше отличий между технологиями LAN и WAN. Во многих технологиях WAN задействована техника предварительно устанавливаемых виртуальных каналов, идентификаторы которых применяются в качестве адресов. Все технологии имеют собственные форматы кадров (в технологии ATM кадр даже называется иначе — ячейкой) и, конечно, собственные стеки протоколов.

Таким образом, чтобы связать между собой сети, построенные на основе столь отличающихся технологий, нужны дополнительные средства, и такие средства предоставляет сетевой уровень. Функции сетевого уровня реали-

зуются соответствующей группой протоколов, а также специальными устройствами — *маршрутизаторами*.

Главной функцией маршрутизатора является физическое соединение сетей. Маршрутизатор имеет несколько сетевых интерфейсов, подобных интерфейсам компьютера, к каждому из которых может быть подключена одна сеть. Таким образом, все интерфейсы маршрутизатора можно считать узлами разных сетей. Маршрутизатор может быть реализован программно, на базе универсального компьютера (например, типовая конфигурация Unix или Windows включает программный модуль маршрутизатора). Однако чаще маршрутизаторы реализуются на базе специализированных аппаратных платформ. В состав программного обеспечения маршрутизатора входят протокольные модули сетевого уровня. Заметим, что в общем случае функции сетевого уровня шире, чем обеспечение обмена в пределах составной сети. Так, сетевой уровень решает задачу создания надежных и гибких барьеров на пути нежелательного трафика между сетями.

На сетевом уровне определяются два вида протоколов. Первый вид — *маршрутизируемые протоколы* — реализуют продвижение пакетов через сеть. Именно эти протоколы обычно имеют в виду, когда говорят о протоколах сетевого уровня. Однако часто к сетевому уровню относят и другой вид протоколов, называемых *маршрутизирующими протоколами*, или *протоколами маршрутизации*. С помощью этих протоколов маршрутизаторы собирают информацию о топологии межсетевых соединений, на основании которой осуществляется выбор маршрута продвижения пакетов.

На пути от отправителя к получателю пакеты могут быть искажены или утеряны. Хотя некоторые приложения имеют собственные средства обработки ошибок, существуют и такие, которые предпочитают сразу иметь дело с надежным соединением. *Транспортный уровень* (transport layer) обеспечивает приложениям или верхним уровням стека — прикладному, представления и сеансовому — передачу данных с той степенью надежности, которая им требуется. Модель OSI определяет 5 классов транспортных услуг: от низшего класса 0 до высшего класса 4. Эти виды услуг 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 обнару-

жению и исправлению ошибок передачи, таких, как искажение, потеря и дублирование пакетов.

Выбор класса услуг транспортного уровня определяется, с одной стороны, тем, в какой степени задача обеспечения надежности решается самими приложениями и протоколами более высоких, чем транспортный, уровней. С другой стороны, этот выбор зависит от того, насколько надежной является система транспортировки данных в сети, обеспечиваемая уровнями, расположенными ниже транспортного, — сетевым, канальным и физическим. Так, если качество каналов передачи связи достаточно высокое, и вероятность возникновения ошибок, не обнаруженных протоколами более низких уровней, невелика, то разумно воспользоваться одним из облегченных услуг транспортного уровня, не обремененных многочисленными проверками, квитированием и другими приемами повышения надежности. Если же транспортные средства нижних уровней не очень надежны, то целесообразно обратиться к наиболее развитой услуге транспортного уровня, которая работает, используя максимум средств для обнаружения и устранения ошибок, включая предварительное установление логического соединения, контроль доставки сообщений по контрольным суммам и циклической нумерации пакетов, установление тайм-аутов доставки и т. п.

Все протоколы, начиная с транспортного уровня и выше, реализуются программными средствами конечных узлов сети — компонентами их сетевых операционных систем. В качестве примера транспортных протоколов можно привести протоколы TCP и UDP стека TCP/IP и протокол SPX стека Novell.

Протоколы нижних четырех уровней обобщенно называют *сетевым транспортом*, или *транспортной подсистемой*, так как они полностью решают задачу транспортировки сообщений с заданным уровнем качества в составных сетях с произвольной топологией и различными технологиями. Оставшиеся три верхних уровня решают задачи предоставления прикладных услуг, используя нижележащую транспортную подсистему.

Сеансовый уровень (session layer) обеспечивает управление взаимодействием сторон: фиксирует, какая из сторон является активной в настоящий момент и предоставляет средства синхронизации сеанса. Эти средства позволяют в ходе длинных передач сохранять информацию о состоянии этих передач в виде контрольных точек, 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. Функции этого уровня часто объединяют с функциями прикладного уровня и реализуют в одном протоколе.

Уровень представления (presentation layer), как явствует из его названия, обеспечивает представление 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. С помощью средств данного уровня протоколы прикладных уровней могут преодолеть синтаксические различия в представлении данных или же различия в кодах символов, например кодов ASCII и EBCDIC. На этом уровне могут выполняться шифрование и дешифрирование данных, благодаря которым секретность обмена данными обеспечивается сразу для всех прикладных служб. Примером такого протокола является протокол SSL (Secure Socket Layer — слой защищенных гнезд), который обеспечивает секретный обмен сообщениями для протоколов прикладного уровня стека TCP/IP.

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.* Существует очень большое разнообразие протоколов и соответствующих служб прикладного уровня. Приведем в качестве примера несколько наиболее распространенных реализаций сетевых файловых служб: NFS и FTP в стеке TCP/IP, SMB в Microsoft Windows, NCP в операционной системе Novell NetWare.

2.3. СТАНДАРТНЫЕ СТЕКИ КОММУНИКАЦИОННЫХ ПРОТОКОЛОВ

Важно различать *модель* OSI и *стек протоколов* OSI. В то время, как модель OSI является концептуальной схемой взаимодействия открытых систем, стек OSI представляет собой набор спецификаций конкретных протоколов.

В отличие от других стеков протоколов стек OSI (рис. 2.4) полностью соответствует модели OSI, включая спецификации протоколов для всех семи уровней взаимодействия, определенных в этой модели. Это и понятно, поскольку разработчики стека OSI использовали модель OSI как прямое руководство к действию. Протоколы стека OSI отличает большая сложность и неоднозначность спецификаций. Эти свойства явились результатом общей политики разработчиков стека, стремившихся учесть в своих протоколах все многообразие уже существующих и появляющихся технологий.

<table>
<tr><td>7</td><td rowspan="2">X.400</td><td>X.500</td><td>VT</td><td>FTAM</td><td>JTM</td><td>Другие</td></tr>
<tr><td>6</td><td colspan="5">Протокол уровня представления</td></tr>
<tr><td>5</td><td colspan="6">Сеансовый протокол</td></tr>
<tr><td>4</td><td colspan="6">Транспортные протоколы (классы 0–4)</td></tr>
<tr><td>3</td><td colspan="6">ES — ES, IS — IS, CONP, CLNP</td></tr>
<tr><td>2</td><td rowspan="2">Ethernet</td><td rowspan="2">Token Bus</td><td rowspan="2">Token Ring</td><td>X.25</td><td rowspan="2">ISDN</td><td rowspan="2">FDDI</td></tr>
<tr><td>1</td><td>HDLC LAP-B</td></tr>
</table>

Рис. 2.4. Стек протоколов OSI

На физическом и канальном уровнях стек OSI поддерживает протоколы Ethernet, Token Ring, FDDI, а также протоколы LLC, X.25 и ISDN, т. е. использует все разработанные вне стека популярные протоколы нижних уровней, как и большинство других стеков.

Сетевой уровень включает сравнительно редко используемые протоколы Connection-oriented Network Protocol (CONP) и Connectionless Network Protocol (CLNP). Как следует из названий, первый из них ориентирован на соединение (connection-oriented), второй — нет (connectionless).

Более популярны протоколы маршрутизации стека OSI: ES-IS (End System-Intermediate System) между конечной и промежуточной системами и IS-IS (Intermediate System-Intermediate System) между промежуточными системами.

Транспортный уровень стека OSI в соответствии с функциями, определенными для него в модели OSI, скрывает различия между сетевыми услугами с установлением соединения и без установления соединения, так что пользователи получают требуемое качество обслуживания независимо от

нижележащего сетевого уровня. Чтобы обеспечить это, транспортный уровень требует, чтобы пользователь задал нужное качество обслуживания.

Службы прикладного уровня обеспечивают передачу файлов, эмуляцию терминала, службу каталогов и почту. Из них наиболее популярными являются служба каталогов (стандарт X.500), электронная почта (X.400), протокол виртуального терминала (VTP), протокол передачи, доступа и управления файлами (FTAM), протокол пересылки и управления работами (JTM).

Стек IPX/SPX является оригинальным стеком протоколов фирмы Novell, разработанным для сетевой операционной системы NetWare еще в начале 80-х гг. Структура стека IPX/SPX и его соответствие модели OSI иллюстрируется на рис. 2.5. Протоколы сетевого и транспортного уровней — Internetwork Packet Exchange (IPX) и Sequenced Packet Exchange (SPX) — дали название данному стеку. К сетевому уровню этого стека отнесены также протоколы маршрутизации RIP и NLSP. В качестве представителей трех верхних уровней на рисунке приведены два популярных протокола: протокол удаленного доступа к файлам NetWare Core Protocol (NCP) и протокол объявления о сервисах Service Advertising Protocol (SAP).

<table>
<tr><td>7</td><td rowspan="3">SAP</td><td colspan="2" rowspan="3">NCP</td></tr>
<tr><td>6</td></tr>
<tr><td>5</td></tr>
<tr><td>4</td><td colspan="3">SPX</td></tr>
<tr><td>3</td><td>IPX</td><td>RIP</td><td>NLSP</td></tr>
<tr><td>2</td><td colspan="3" rowspan="2">Ethernet, Token Ring, FDDI и др</td></tr>
<tr><td>1</td></tr>
</table>

Рис. 2.5. Стек протоколов IPX/SPX

Многие особенности стека IPX/SPX обусловлены ориентацией ранних версий ОС NetWare на работу в локальных сетях небольших размеров, состоящих из персональных компьютеров со скромными ресурсами. Понятно, что для таких компьютеров компании Novell нужны были протоколы, на реализацию которых требовалось бы минимальное количество оперативной памяти (ограниченной в IBM-совместимых компьютерах под управлением MS-DOS объемом 640 Кбайт) и которые бы быстро работали на процессорах небольшой

вычислительной мощности. В результате протоколы стека IPX/SPX до недавнего времени отлично справлялись с работой в локальных сетях. Однако в крупных корпоративных сетях они слишком перегружали медленные глобальные каналы связи широковещательными пакетами, интенсивно использующимися несколькими протоколами этого стека, например протоколом SAP. Это обстоятельство, а также тот факт, что стек IPX/SPX является собственностью фирмы Novell, и на его реализацию нужно получать лицензию (т. е. открытые спецификации не поддерживались), долгое время ограничивали распространенность его только сетями NetWare.

Стек TCP/IP был разработан по инициативе Министерства обороны США для связи экспериментальной сети ARPANET с другими сетями как набор общих протоколов для разнородной вычислительной среды. Большой вклад в развитие стека TCP/IP, который получил свое название по популярным протоколам IP и TCP, внес университет Беркли, реализовав протоколы стека в своей версии ОС Unix. Популярность этой операционной системы привела к широкому распространению протоколов TCP, IP и других протоколов стека. Сегодня этот стек используется для связи компьютеров в Интернете, а также в огромном числе корпоративных сетей.

Поскольку стек TCP/IP изначально создавался для Интернета, он имеет много особенностей, дающих ему преимущество перед другими протоколами, когда речь заходит о построении сетей, включающих глобальные связи. В частности, очень полезным свойством, делающим возможным применение этого протокола в больших сетях, является его способность фрагментировать пакеты. Действительно, большая составная сеть часто состоит из сетей, построенных на совершенно разных принципах. В каждой из этих сетей может быть собственная величина максимальной длины единицы передаваемых данных (кадра). В таком случае при переходе из одной сети, имеющей большую максимальную длину, в сеть с меньшей максимальной длиной может возникнуть необходимость деления передаваемого кадра на несколько частей. Протокол IP стека TCP/IP эффективно решает эту задачу.

Другой особенностью технологии TCP/IP является гибкая система адресации, позволяющая проще, чем другие протоколы аналогичного назначения включать в составную сеть сети разных технологий. Это свойство

также способствует применению стека TCP/IP для построения больших гетерогенных сетей.

В стеке TCP/IP очень экономично используются широковещательные рассылки. Это свойство совершенно необходимо при работе на медленных каналах связи, характерных для территориальных сетей.

Однако, как и всегда, за получаемые преимущества надо платить, и платой здесь оказываются высокие требования к ресурсам и сложность администрирования IP-сетей. Мощные функциональные возможности протоколов стека TCP/IP требуют для своей реализации больших вычислительных затрат. Гибкая система адресации и отказ от широковещательных рассылок приводят к наличию в IP-сети разнообразных централизованных служб типа DNS, DHCP и т. п. Каждая из этих служб направлена на облегчение администрирования сети, но, в то же время, сама требует пристального внимания со стороны администраторов.

<table>
<tr><td>7</td><td rowspan="2">HTTP</td><td rowspan="2">SNMP</td><td rowspan="2">FTP</td><td rowspan="2">telnet</td><td rowspan="2">SMTP</td><td rowspan="2">TFTP</td><td rowspan="2">I</td></tr>
<tr><td>6</td></tr>
<tr><td>5</td><td colspan="5" rowspan="2">TCP</td><td rowspan="2">UDP</td><td rowspan="2">II</td></tr>
<tr><td>4</td></tr>
<tr><td>3</td><td colspan="2">IP</td><td>ICMP</td><td>RIP</td><td colspan="2">OSPF</td><td>III</td></tr>
<tr><td>2</td><td colspan="5" rowspan="2">Не регламентируется
Ethernet, Token Ring, FDDI, X.25, SPIP, PPP</td><td rowspan="2">IV</td></tr>
<tr><td>1</td></tr>
<tr><td>Уровни модели OSI</td><td colspan="6"></td><td>Уровни стека TCP/IP</td></tr>
</table>

Рис. 2.6. Архитектура стека TCP/IP

Можно приводить и другие доводы за и против стека протоколов Интернета, однако факт остается фактом: сегодня это самый популярный стек, широко используемый как в глобальных, так и локальных сетях.

На рис. 2.6 приведена структура стека TCP/IP. Так как стек TCP/IP был разработан до появления модели ISO/OSI, соответствие в нём уровней стека TCP/IP уровням модели OSI достаточно условно.

В стеке TCP/IP определены четыре уровня.

Прикладной уровень стека TCP/IP соответствует трем верхним уровням модели OSI: прикладному, представления и сеансовому. Он объединяет службы, предоставляемые системой пользовательским приложениям. За долгие годы использования в сетях различных стран и организаций стек TCP/IP накопил большое количество протоколов и служб прикладного уровня. К ним относятся такие распространенные протоколы, как протокол передачи файлов (File Transfer Protocol, FTP), протокол эмуляции терминала (telnet), простой протокол передачи электронной почты (Simple Mail Transfer Protocol, SMTP), протокол передачи гипертекста (HyperText Transfer Protocol, HTTP) и многие другие. Протоколы прикладного уровня развертываются на хостах.

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

- гарантированную доставку, обеспечиваемую *протокол управления передачей* (Transmission Control Protocol, TCP);

- доставку по возможности, или с максимальными усилиями, обеспечиваемую *протокол пользовательских дейтаграмм* (User Datagram Protocol, UDP).

Для того чтобы обеспечить надежную доставку данных протокол TCP предусматривает установление логического соединения, что позволяет ему нумеровать пакеты, подтверждать их прием квитанциями, в случае потери организовывать повторные передачи, распознавать и уничтожать дубликаты, доставлять прикладному уровню пакеты в том порядке, в котором они были отправлены. Этот протокол позволяет объектам на компьютере-отправителе и компьютере-получателе поддерживать обмен данными в дуплексном режиме. TCP дает возможность без ошибок доставить сформированный на одном из компьютеров поток байтов в любой другой компьютер, входящий в составную сеть. TCP делит поток байтов на фрагменты и передает их нижележащему уровню межсетевого взаимодействия. После того как эти фрагменты будут доставлены средствами уровня межсетевого взаимодействия в пункт назначения, протокол TCP снова соберет их в непрерывный поток байтов.

Второй протокол этого уровня — UDP — является простейшим дейтаграммным протоколом, который используется в том случае, когда задача надежного обмена данными либо вообще не ставится, либо решается сред-

ствами более высокого уровня — прикладным уровнем или пользовательскими приложениями.

В функции протоколов транспортного уровня TCP и UDP входит также исполнение роли связующего звена между прилегающими к ним прикладным уровнем и уровнем межсетевого взаимодействия. От прикладного протокола транспортный уровень принимает задание на передачу данных с тем или иным качеством, а после выполнения рапортует ему об этом. Нижележащий уровень межсетевого взаимодействия протоколы TCP и UDP рассматривают как своего рода инструмент, не очень надежный, но способный перемещать пакет в свободном и рискованном путешествии по составной сети. Программные модули, реализующие протоколы TCP и UDP, подобно модулям протоколов прикладного уровня, устанавливаются на хостах.

Сетевой уровень, называемый также уровнем интернета, является стержнем всей архитектуры TCP/IP. Именно этот уровень, функции которого соответствуют сетевому уровню модели OSI, обеспечивает перемещение пакетов в пределах составной сети, образованной объединением множества сетей. 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, о функциях которого будет сказано далее.

Основным протоколом сетевого уровня является *межсетевой протокол* (Internet Protocol, IP). В его задачу входит продвижение пакета между сетями — от одного маршрутизатора до другого до тех пор, пока пакет не попадет в сеть назначения. В отличие от протоколов прикладного и транспортного уровней протокол IP развертывается не только на хостах, но и на всех шлюзах. Протокол IP — это дейтаграммный протокол, работающий без установления соединений по принципу доставки с максимальными усилиями.

К сетевому уровню TCP/IP часто относят протоколы, выполняющие вспомогательные функции по отношению к IP. Это, прежде всего, протоколы маршрутизации RIP и OSPF, занимающиеся изучением топологии сети, определением маршрутов и составлением таблиц маршрутизации, на основании которых протокол IP перемещает пакеты в нужном направлении. По этой же причине к сетевому уровню могут быть отнесены еще два протоко-

ла: *протокол межсетевых управляющих сообщений* (Internet Control Message Protocol, ICMP), предназначенный для передачи маршрутизатором источнику информации об ошибках, возникших при передаче пакета, и *протокол групповой адресации* (Internet Group Management Protocol, IGMP), использующийся для направления пакета сразу по нескольким адресам.

Идеологическим отличием архитектуры стека TCP/IP от многоуровневой организации других стеков является интерпретация функций самого нижнего уровня — уровня сетевых интерфейсов.

Напомним, что нижние уровни модели OSI (канальный и физический) реализуют большое количество функций доступа к среде передачи, формированию кадров и согласованию уровней электрических сигналов, кодированию и синхронизации и некоторые другие. Эти функции составляют суть таких протоколов обмена данными, как Ethernet, Token Ring, PPP, HDLC и многих других.

У нижнего уровня стека TCP/IP задача существенно проще — он отвечает только за организацию взаимодействия с технологиями сетей, входящих в составную сеть. TCP/IP рассматривает любую сеть, входящую в составную сеть, как средство транспортировки пакетов до следующего на пути маршрутизатора.

Задачу обеспечения интерфейса между технологией TCP/IP и любой другой технологией промежуточной сети упрощенно можно свести к определению способа упаковки (инкапсуляции) IP-пакета в единицу передаваемых данных промежуточной сети, а также к определению способа преобразования сетевых адресов в адреса технологии данной промежуточной сети. Такой подход делает составную сеть TCP/IP открытой для включения любой сети, какую бы внутреннюю технологию передачи данных эта сеть не использовала. Для каждой новой технологии должны быть разработаны собственные интерфейсные средства. Следовательно, функции этого уровня нельзя определить раз и навсегда.

Уровень сетевых интерфейсов в стеке TCP/IP не регламентируется. Он поддерживает все популярные технологии; для локальных сетей — это Ethernet, Token Ring, FDDI, Fast Ethernet, Gigabit Ethernet, для глобальных сетей — протоколы двухточечных соединений SLIP и PPP, технологии X.25, Frame Relay, ATM.

Каждый коммуникационный протокол оперирует некоторой единицей передаваемых данных. Названия этих единиц иногда закрепляются стандартом, а чаще просто определяются традицией. В стеке TCP/IP за многие годы его существования образовалась устоявшаяся терминология в этой области (рис. 2.7).

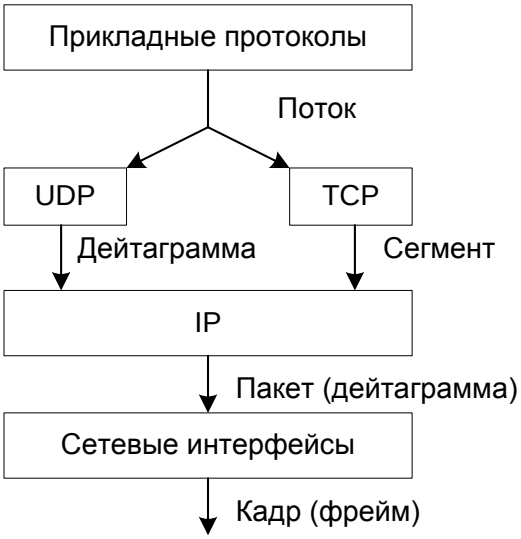


Рис. 2.7. Названия PDU в TCP/IP

Потоком данных, или просто *потоком*, называют данные, поступающие от приложений на вход протоколов транспортного уровня — TCP и UDP. Протокол TCP «нарезает» из потока данных отдельные *сегменты*. Единицу данных протокола UDP часто называют *дейтаграммой* — это общее название для единиц данных, которыми оперируют протоколы без установления соединений. К таким протоколам относится и протокол IP, поэтому его единицу данных также называют дейтаграммой. Однако очень часто используется и другой термин — *пакет*.

В стеке TCP/IP принято называть *кадрами* (*фреймами*) единицы данных любых технологий, в которые упаковываются IP-пакеты для последующей переноски их через сети составной сети. При этом не имеет значения, какое название используется для этой единицы данных в технологии составляющей сети. Для TCP/IP кадром является и кадр Ethernet, и ячейка ATM, и пакет X.25, так как

все они выступают в качестве контейнера, в котором IP-пакет переносится через составную сеть.

В реальных сетях некоторые из коммуникационных устройств поддерживают не только протоколы трех нижних уровней, но также и протоколы верхних уровней. Так, маршрутизаторы реализуют протоколы маршрутизации, позволяющие автоматически строить таблицы маршрутизации, а концентраторы и коммутаторы часто поддерживают протоколы SNMP и telnet, которые не нужны для выполнения основных функций этих устройств, но позволяют конфигурировать и управлять ими удаленно. Все эти протоколы являются протоколами прикладного уровня и выполняют некоторые вспомогательные (служебные) функции транспортной системы. Очевидно, что для работы прикладных протоколов сетевые устройства должны также поддерживать протоколы промежуточных уровней, таких как IP и TCP/UDP.

Вспомогательные протоколы можно разделить на группы, в соответствии с их функциями.

Первую группу вспомогательных протоколов представляют протоколы *маршрутизации,* такие как RIP, OSPF, BGP. Без этих протоколов маршрутизаторы не смогут продвигать пакеты, так как таблица маршрутизации будет пустой (если только администратор не заполнит ее вручную, но это не очень хорошее решение для крупной сети). Если рассматривать не только стек TCP/IP, но и стеки протоколов сетей с виртуальными каналами, то в эту группу попадают служебные протоколы, которые используются для установления виртуальных каналов.

Другая группа вспомогательных протоколов выполняет *преобразование адресов.* Здесь работает протокол DNS, который преобразует символьные имена узлов в IP-адреса. Протокол DHCP позволяет назначать IP-адреса узлам динамически, а не статически, что облегчает работу администратора сети.

Третью группу образуют протоколы, которые используются для *управления сетью.* В стеке TCP/IP здесь находится протокол SNMP (Simple Network Management Protocol — простой протокол управления сетью), который позволяет автоматически собирать информацию об ошибках и отказах устройств, а также протокол telnet, с помощью которого администратор может удаленно конфигурировать коммутатор или маршрутизатор.

ВОПРОСЫ И ЗАДАНИЯ К ГЛАВЕ 2

1. Обоснуйте причины применения многоуровневого подхода в задачах описания, синтеза и анализа телекоммуникационных сетей.

2. В чем состоят отличия понятий «интерфейс» и «протокол»?

3. Перечислите функции физического и канального уровней. Сравните их между собой.

4. Перечислите основные функции сетевого и транспортного уровня. В чем их основные отличия? Как они дополняют друг друга?

5. Каким телекоммуникационным оборудованием реализуются функции физического, канального и сетевого уровня?

6. Перечислите отличия эталонной модели взаимодействия открытых систем и модели стека протоколов TCP/IP.

7. Какими средствами реализуются функции уровней с транспортного по прикладной?

8. Перечислите технологии и протоколы, которые могут работать на канальном уровне под IP?

9. В чем отличие терминов «протокол транспортного уровня» и «технология построения транспортной сети»?

3. ТЕХНОЛОГИИ ЛОКАЛЬНЫХ СЕТЕЙ

Локальные сети являются неотъемлемой частью любой современной компьютерной сети. Если рассматривать структуру глобальной сети, например Интернета или крупной корпоративной сети, то можно обнаружить, что практически все информационные ресурсы этой сети сосредоточены в локальных сетях, а глобальная сеть является транспортом, который соединяет многочисленные локальные сети.

Одним из основных назначений локальной сети является объединение компьютеров в пределах одного здания или нескольких близко стоящих зданий для предоставления пользователям сети доступа к информационным услугам локальных серверов. Локальные сети также являются удобным средством группирования компьютеров для объединения их в глобальную сеть, так как глобальной сети проще маршрутизировать данные между сетями, а не отдельными компьютерами.

В данной главе рассмотрим некоторые наиболее распространенные технологии построения локальных сетей [2 — 6].

3.1. ТЕХНОЛОГИЯ ETHERNET

Ethernet — это самый распространенный на сегодня стандарт локальных сетей. Общее количество сетей, работающих по протоколу Ethernet в настоящее время оценивается в несколько миллионов.

Когда говорят Ethernet, под этим обычно понимают любой из вариантов этой технологии, в которую входят сегодня также Fast Ethernet, Gigabit Ethernet и 10G Ethernet. В более узком смысле Ethernet — это сетевой стандарт передачи данных со скоростью 10 Мбит/с, который появился в конце 1970-х гг. как стандарт трех компаний — Digital, Intel и Xerox. В начале 1980-х гг. Ethernet был стандартизован рабочей группой IEEE 802.3, и с тех пор он является международным стандартом. Технология Ethernet была первой технологией, которая предложила использовать разделяемую среду для доступа к сети.

Локальные сети, являясь пакетными сетями, используют принцип временн*о*го мультиплексирования, т. е. разделяют передающую среду во времени. Алгоритм управления доступом к среде является одной из важнейших характеристик любой технологии LAN, в значительно большей

степени определяющей ее облик, чем метод кодирования сигналов или формат кадра. В технологии Ethernet в качестве алгоритма разделения среды применяется метод случайного доступа. И хотя его трудно назвать совершенным — при росте нагрузки полезная пропускная способность сети резко падает — он благодаря своей простоте послужил основной причиной успеха технологии Ethernet.

Популярность стандарта Ethernet 10 Мбит/с послужила мощным стимулом его развития. В 1995 г. был принят стандарт Fast Ethernet, в 1998 г. — Gigabit Ethernet, а в 2002 г. — 10G Ethernet. Каждый из новых стандартов превышал скорость своего предшественника в 10 раз, образуя впечатляющую иерархию скоростей 10 Мбит/с — 100 Мбит/с — 1000 Мбит/с — 10 Гбит/с.

Технологии локальных сетей реализуют, как правило, функции только двух нижних уровней модели OSI — физического и канального (рис. 3.1). Функциональности этих уровней достаточно для доставки кадров в пределах стандартных топологий, которые поддерживают LAN — звезда (общая шина), кольцо и дерево.

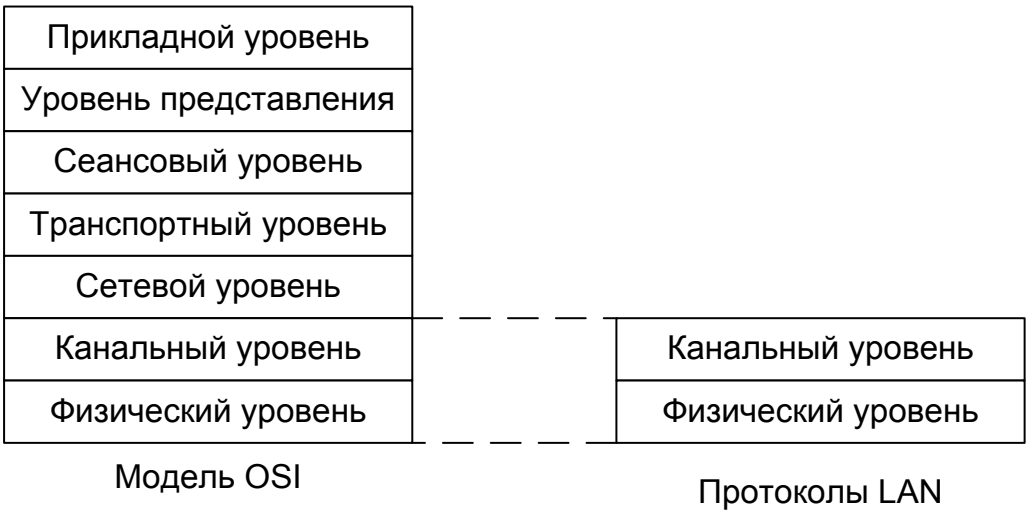


Рис. 3.1. Соответствие протоколов LAN уровням модели OSI

Канальный уровень локальных сетей делится на два подуровня, которые часто также называют просто уровнями: *уровень управления логическим каналом* (Logical Link Control, LLC) и *уровень управления доступом к среде* (Media Access Control, MAC).

Функции уровня LLC обычно реализуются программно соответствующим модулем операционной системы, а функции уровня MAC — программно-аппаратно: сетевым адаптером и его драйвером.

Основными функциями уровня MAC являются обеспечение доступа к разделяемой среде, а также передача кадров между конечными узлами, используя функции и устройства физического уровня.

Одним из основных методов захвата разделяемой среды является *метод случайного доступа*. Он основан на том, что узел, у которого есть кадр для передачи, пытается его отправить без какой бы то ни было предварительной процедуры согласования времени использования разделяемой среды с другими узлами сети.

Метод случайного доступа является децентрализованным, он не требует наличия в сети специального узла, который играл бы роль арбитра, регулирующего доступ к среде. Результатом этого является высокая вероятность *коллизий,* т. е. случаев одновременной передачи кадра несколькими станциями. Во время коллизии происходит наложение сигналов нескольких передатчиков, из-за чего информация всех передаваемых на периоде коллизии кадров искажается. Поскольку в локальных сетях применяются достаточно простые методы кодирования, они не позволяют выделить нужный сигнал из суммарного, как это, например, может делать технология CDMA.

Одним способом улучшения случайного доступа является введение *процедуры прослушивания среды* перед передачей. Узел не имеет права передавать кадр, если он обнаруживает, что среда уже занята передачей другого кадра. Это снижает вероятность коллизий (хотя и не исключает их).

Алгоритмы случайного доступа не гарантируют узлу, что он получит доступ к разделяемой среде в течение определенного времени. Какое бы большое время ожидания ни выбиралось, всегда есть ненулевая вероятность, что реальное время ожидания превысит этот предел. Алгоритмы случайного доступа также не предоставляют никаких возможностей для дифференцированной поддержки характеристик QoS для разных типов трафика — все кадры получают одинаковый уровень доступа к среде.

Транспортировка кадров осуществляется уровнем MAC в несколько этапов, которые в общем случае не зависят от выбранного метода доступа.

1. *Формирование кадра.* На этом этапе осуществляется заполнение полей кадра на основании информации, получаемой от протокола верхнего уровня, такой, как адреса источника и назначения, пользовательские данные, признак протокола верхнего уровня, отсылающего эти данные. После того как кадр сформирован, уровень MAC подсчитывает контрольную сумму кадра и помещает ее в соответствующее поле.

2. *Передача кадра через среду.* Когда кадр сформирован, и доступ к разделяемой среде получен, уровень MAC передает кадр на физический уровень, который побитно передает все поля кадра в среду. Функции физического уровня выполняет передатчик сетевого адаптера, который преобразует байты кадра в последовательность битов и кодирует их соответствующими электрическими или оптическими сигналами. После прохождения сигналов по среде они поступают в приемники сетевых адаптеров, подключенных к разделяемой среде, которые выполняют обратное преобразование сигналов в байты кадра.

3. *Прием кадра.* Уровень MAC каждого узла сети, подключенного к разделяемой среде, проверяет адрес назначения поступившего кадра, и если он совпадает с его собственным адресом, то продолжает его обработку; в противном случае кадр отбрасывается. Продолжение обработки заключается в проверке корректности контрольной суммы кадра. Кадр с корректной контрольной суммой передается уровнем MAC вверх по стеку, на чем функции уровня MAC заканчиваются. Если же контрольная сумма кадра говорит о том, что информация при передаче через среду была искажена, то кадр отбрасывается.

Из приведенного описания следует, что технология Ethernet реализует дейтаграммный полудуплексный режим передачи данных.

Уровень LLC выполняет две функции: организует интерфейс с прилегающим к нему сетевым уровнем и обеспечивает доставку кадров с заданной степенью надежности.

Интерфейсные функции LLC заключаются в передаче пользовательских и служебных данных между уровнем MAC и сетевым уровнем. При передаче данных сверху вниз уровень LLC принимает от протокола сетевого уровня пакет (например, IP- или IPX-пакет), в котором уже находятся пользовательские данные. Помимо пакета сверху также передается адрес узла назначения в формате той технологии LAN, которая будет использо-

вана для доставки кадра в пределах данной локальной сети. Напомним, что в терминах стека TCP/IP такой адрес называется аппаратным. Полученные от сетевого уровня пакет и аппаратный адрес уровень LLC передает далее вниз — уровню MAC. Кроме того, LLC при необходимости решает задачу мультиплексирования, передавая данные от нескольких протоколов сетевого уровня единственному протоколу уровня MAC.

При передаче данных снизу вверх LLC принимает от уровня MAC пакет сетевого уровня, пришедший из сети. Теперь ему нужно выполнить еще одну интерфейсную функцию — демультиплексирование, т. е. решить, какому из сетевых протоколов передать полученные от MAC данные (рис. 3.2).

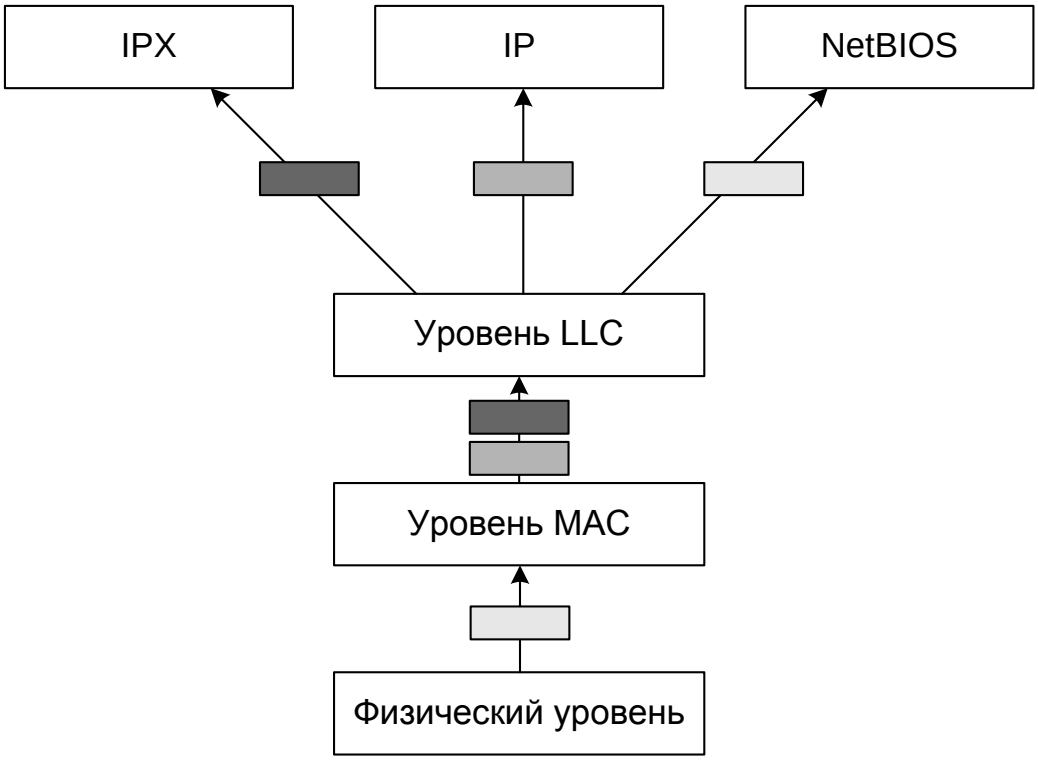


Рис. 3.2. Демультиплексирование кадров протоколом LLC

Задачи мультиплексирования и демультиплексирования свойственны не только LLC, но и любому протоколу, над которым может работать несколько протоколов. Для демультиплексирования данных LLC использует в своем заголовке специальные поля (рис. 3.3).

Адрес точки входа службы приемника (DSAP)	Адрес точки входа службы источника (SSAP)	Управляющее поле	Поле данных

Рис. 3.3. Формат LLC-кадра

Поле DSAP (Destination Service Access Point — точка входа службы приемника) используется для хранения кода протокола, которому адресовано содержимое поля данных. Соответственно, поле SSAP (Source Service Access Point — точка входа доступа к услуге источника) используется для указания кода протокола, от которого посылаются данные. Применение двух полей для целей демультиплексирования является нетипичным, обычно протоколы обходятся одним полем, например, протокол IP всегда отсылает свои пакеты протоколу IP, а протокол IPX — протоколу IPX. Два поля полезны в тех случаях, когда вышележащий протокол поддерживает несколько режимов работы, так что протокол на узле-отправителе может использовать различные значения DSAP и SSAP для уведомления узла-получателя о переходе в новый режим работы. Этим свойством протокола LLC часто пользуется протокол NetBEUI.

Обеспечение доставки кадров с заданной степенью надежности — вторая основная функция уровня LLC. Протокол LLC поддерживает несколько режимов работы, отличающихся наличием или отсутствием процедур восстановления кадров в случае их потери или искажения, т. е. отличающихся надежностью доставки. Уровень LLC, непосредственно прилегающий к сетевому уровню, принимает от него запрос на выполнение транспортной операции канального уровня с тем или иным качеством.

Уровень LLC предоставляет верхним уровням три типа транспортных услуг.

1. *Услуга LLC1 без установления соединения и без подтверждения получения данных.* Данная услуга дает пользователю средства для передачи данных с минимумом издержек. В этом случае LLC поддерживает дейтаграммный режим работы, как и MAC, так что технология LAN в целом работает в дейтаграммном режиме. Обычно эта процедура используется, когда такие функции, как восстановление данных после ошибок и упорядочивание данных выполняются протоколами вышележащих уровней, по-

этому нет нужды дублировать их на уровне LLC.

2. *Услуга LLC2 с логическим соединением.* Данная услуга дает пользователю возможность установить логическое соединение перед началом передачи любого блока данных и, если это требуется, выполнить процедуры восстановления после ошибок и упорядочивание потока блоков в рамках установленного соединения. Для надежной доставки данных протокол LLC2 использует алгоритм скользящего окна.

3. *Услуга LLC3 без установления соединения, но с подтверждением получения данных.* В некоторых случаях (например, при использовании сетей в системах реального времени, управляющих промышленными объектами), с одной стороны, временные издержки установления логического соединения перед отправкой данных неприемлемы, а с другой — необходимо подтверждение о корректности приема переданных данных. Для такого рода ситуаций и предусмотрена дополнительная услуга LLC3, которая является компромиссом между LLC1 и LLC2, так как она не предусматривает установление логического соединения, но обеспечивает подтверждение получения данных.

Какой из трех режимов работы уровня LLC будет использован, зависит от требований протокола верхнего уровня. Информация о требуемой от LLC транспортной услуге передается через межуровневый интерфейс уровню LLC вместе с аппаратным адресом и пакетом с пользовательскими данными. Например, когда поверх LLC работает протокол IP, он всегда запрашивает режим LLC1, поскольку в стеке TCP/IP задачу обеспечения надежной доставки решает протокол TCP.

Технология Ethernet использует метод доступа CSMA/CD — *коллективный доступ с опознаванием несущей и обнаружением коллизий* (Carrier Sense Multiple Access with Collision Detection). Предполагая для простоты изложения, что каждый узел (станция) имеет только один сетевой интерфейс, рассмотрим, как на основе алгоритма CSMA/CD происходит передача данных в сети Ethernet.

На уровне MAC, который обеспечивает доступ к среде и передачу кадра, для идентификации сетевых интерфейсов узлов сети используются уникальные шестибайтовые адреса, называемые MAC-адресами. Обычно MAC-адрес записывают в виде шести пар шестнадцатеричных цифр, разделенных тире или двоеточиями, например 11:A0:17:3D:BC:01. Каждый

сетевой адаптер имеет, по крайней мере, один MAC-адрес.

Все компьютеры в сети с разделяемой средой имеют возможность немедленно (с учетом задержки распространения сигнала по физической среде) получить данные, которые любой из компьютеров начал передавать в общую среду. Говорят, что среда, к которой подключены все станции, работает в режиме коллективного доступа (Multiply Access, MA).

Чтобы получить возможность передавать кадр, интерфейс-отправитель должен убедиться, что разделяемая среда свободна. Это достигается прослушиванием основной гармоники сигнала, которая также называется несущей частотой (Carrier Sense, CS). Признаком «незанятости» среды является отсутствие в ней несущей частоты, которая при манчестерском способе кодирования, принятом для всех вариантов Ethernet 10 Мбит/с, равна 5…10 МГц в зависимости от последовательности единиц и нулей, передаваемых в данный момент.

Если среда свободна, то узел имеет право начать передачу кадра. В примере, показанном на рис. 3.4, узел 1 обнаружил, что среда свободна и начал передавать свой кадр. В классической сети Ethernet, реализованной на коаксиальном кабеле, сигналы передатчика узла 1 распространяются в обе стороны, так что их получают все узлы сети. Кадр данных всегда сопровождается преамбулой, которая состоит из 7 байт, каждый из которых имеет значение 10101010, и 8-го байта, равного 10101011. Последний байт носит название *ограничителя начала кадра*. Преамбула нужна для вхождения приемника в побитовую и побайтовую синхронизацию с передатчиком. Наличие двух единиц, идущих подряд, говорит приемнику о том, что преамбула закончилась и следующий бит является началом кадра.

Все станции, подключенные к кабелю, начинают записывать байты передаваемого кадра в свои внутренние буферы. Первые 6 байт кадра содержат адрес назначения. Та станция, которая узнает собственный адрес в заголовке кадра, продолжает записывать его содержимое в свой внутренний буфер, а остальные станции на этом прием кадра прекращают. Станция назначения обрабатывает полученные данные, передает их вверх по своему стеку. Кадр Ethernet содержит не только адрес назначения, но и адрес источника данных, поэтому станция-получатель знает, кому нужно послать ответ.

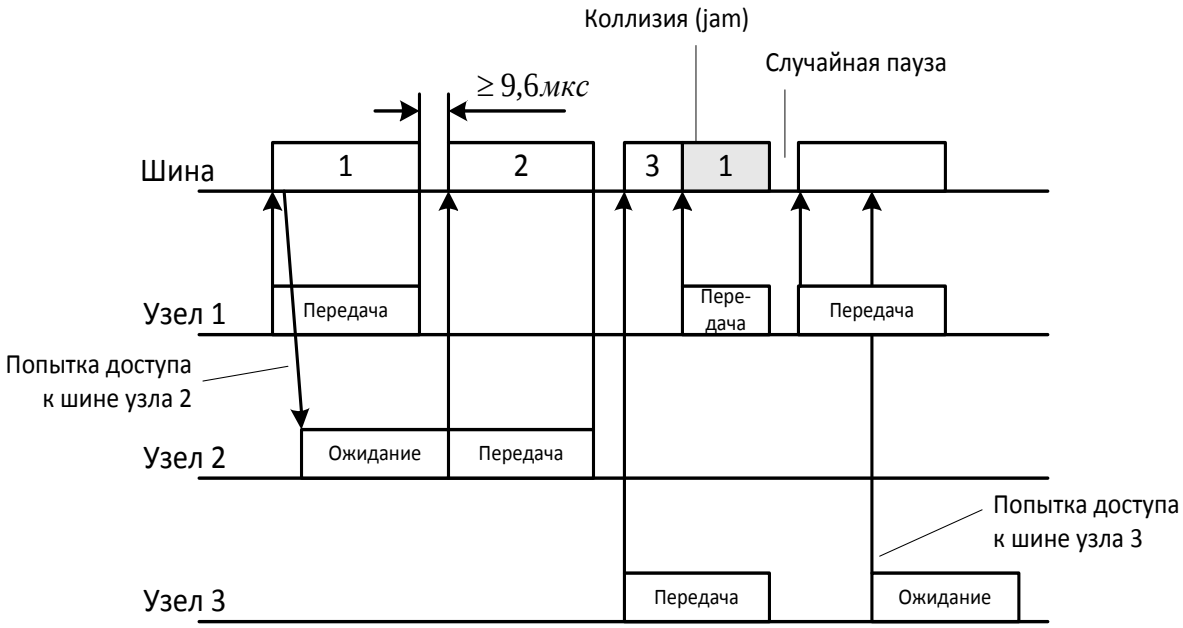


Рис. 3.4. Метод случайного доступа CSMA/CD

Узел 2 во время передачи кадра узлом 1 также пытается начать передачу своего кадра, однако обнаруживает, что среда занята — на ней присутствует несущая частота, поэтому узел 2 вынужден ждать, пока узел 1 не прекратит передачу кадра.

После окончания передачи кадра все узлы сети обязаны выдержать технологическую паузу, равную межпакетному интервалу (Inter Packet Gap, IPG), равному 9,6 мкс. Эта пауза нужна для приведения сетевых адаптеров в исходное состояние, а также для предотвращения монопольного захвата среды одной станцией. После окончания технологической паузы узлы имеют право начать передачу своего кадра, так как среда свободна. В приведенном примере узел 2 дождался окончания передачи кадра узлом 1, сделал паузу в 9,6 мкс и начал передачу своего кадра.

Механизм прослушивания среды и пауза между кадрами не гарантируют исключения такой ситуации, когда две или более станции одновременно решают, что среда свободна и начинают передавать свои кадры. Говорят, что при этом происходит *коллизия*, так как содержимое обоих кадров сталкивается на общем кабеле и происходит искажение информации. Следует подчеркнуть, что наличие коллизий — это нормальная ситуация в работе сетей Ethernet, и для возникновения коллизии не обязательно, чтобы несколько станций начали передачу абсолютно одновременно, что ма-

ловероятно. Более близка к реальности ситуация, когда один узел начинает передачу, а через некоторое небольшое время другой узел, проверив среду и не обнаружив несущую (сигналы первого узла еще не успели до него дойти), начинает передачу своего кадра. Таким образом, возникновение коллизии является следствием распределения узлов локальной сети в пространстве.

Чтобы корректно обработать коллизию все станции одновременно наблюдают за распространяющимися в кабеле сигналами. Если передаваемые и наблюдаемые сигналы отличаются, то фиксируется факт обнаружения коллизии (Collision Detection, CD). Для увеличения вероятности скорейшего обнаружения коллизии та станция, которая обнаружила коллизию, прерывает передачу своего кадра (в произвольном месте, возможно, и не на границе байта) и усиливает ситуацию коллизии посылкой в сеть специальной последовательности (jam-последовательности) из 32 бит. После этого обнаружившая коллизию передающая станция обязана прекратить передачу и сделать паузу в течение короткого случайного интервала времени. Затем она может снова предпринять попытку захвата среды и передачи кадра. При этом случайная пауза выбирается, исходя из простой формулы:

$$\text{пауза} = L \times \text{интервал отсрочки},$$

где L представляет собой целое число, выбранное с равной вероятностью из диапазона $[0; 2N]$, а N — номер повторной попытки передачи данного кадра: 1, 2,..., 10. После 10-й попытки интервал, из которого выбирается пауза, не увеличивается.

В сетях Ethernet интервал отсрочки выбран равным значению 512 битовых интервалов. Битовый интервал соответствует времени между появлением двух последовательных битов данных в кабеле; для скорости 10 Мбит/с величина битового интервала равна 0,1 мкс. Таким образом, случайная пауза в технологии Ethernet может принимать значения от 0 до 52,4 мс.

Надежное распознавание коллизий всеми станциями сети является необходимым условием корректной работы сети Ethernet. Если какая-либо передающая станция не распознает коллизию и решит, что кадр данных передан ею верно, этот кадр будет утерян. Из-за наложения сигналов при коллизии содержащаяся в кадре информация исказится, и он будет отбракован принимающей станцией из-за несовпадения контрольной суммы.

Скорее всего, недошедшие до получателя данные будут повторно переданы каким-либо протоколом верхнего уровня, например транспортным или прикладным, работающим с установлением соединения, или протоколом LLC, если он работает в режиме LLC2. Однако повторная передача сообщения протоколами верхних уровней произойдет гораздо позже (иногда по прошествии нескольких секунд), чем повторная передача средствами сети Ethernet, работающей с микросекундными интервалами. Поэтому если коллизии не будут надежно распознаваться узлами сети Ethernet, то это приведет к заметному снижению полезной пропускной способности сети.

Для надежного распознавания коллизий должно выполняться следующее соотношение:

$$T_{\min} \geq \text{PDV}.$$

Здесь $T_{\min}$ — время передачи кадра минимальной длины, а PDV (Path Delay Value) — *время оборота*, т. е. время, за которое сигнал коллизии успевает распространиться до самого дальнего узла сети. В худшем случае сигнал должен пройти дважды между наиболее удаленными друг от друга станциями сети (в одну сторону проходит неискаженный сигнал, а в обратном направлении распространяется уже искаженный коллизией сигнал). При выполнении этого условия передающая станция должна успеть обнаружить коллизию, которую вызвал переданный ее кадр, еще до того, как она закончит передачу этого кадра.

Очевидно, что выполнение этого условия зависит, с одной стороны, от длины минимального кадра и скорости передачи данных, а с другой — от длины кабельной системы сети и скорости распространения сигнала в кабеле (для разных типов кабеля эта скорость несколько отличается). Стандарт Ethernet определяет минимальную длину поля данных кадра в 46 байт, что вместе со служебными полями дает минимальную длину кадра 64 байт, а вместе с преамбулой — 72 байт, или 576 бит. Отсюда может быть вычислено ограничение на расстояние между станциями. В стандарте Ethernet 10 Мбит/с время передачи кадра минимальной длины равно 575 битовых интервалов, следовательно, время оборота должно быть меньше 57,5 мкс. Расстояние, которое сигнал может пройти за это время, зависит от типа кабеля: для толстого коаксиального кабеля (см. далее) оно составляет приблизительно 13280 м. Учитывая, что за время 57,5 мкс сигнал должен пройти по линии связи дважды, расстояние между двумя узлами не

должно быть больше 6635 м. В стандарте величина этого расстояния выбрана существенно меньше с учетом других, более строгих ограничений.

Одно из таких ограничений связано с предельно допустимым затуханием сигнала. Для обеспечения необходимой мощности сигнала при его прохождении между наиболее удаленными друг от друга станциями максимальная длина непрерывного сегмента толстого коаксиального кабеля с учетом вносимого им затухания выбрана в 500 м. Очевидно, что в кабеле длиной 500 м условия распознавания коллизий будут выполняться с большим запасом для кадров любой стандартной длины, в том числе и 72 байт (время оборота по кабелю 500 м составляет всего 43,3 битовых интервала). Поэтому минимальная длина кадра могла бы быть установлена еще меньше. Однако разработчики технологии не стали уменьшать минимальную длину кадра, имея в виду сети, которые строятся из нескольких сегментов, соединенных повторителями.

Повторители увеличивают мощность передаваемых с сегмента на сегмент сигналов, что позволяет использовать сеть гораздо большей длины. В коаксиальных реализациях Ethernet разработчики ограничили максимальное количество сегментов в сети пятью, что, в свою очередь, ограничивает общую длину сети 2500 м. Даже в такой многосегментной сети условие обнаружения коллизий по-прежнему выполняется с большим запасом (сравним полученное из условия допустимого затухания расстояние в 2500 м с вычисленным выше максимально возможным по времени распространения сигнала расстоянием 6635 м). Однако в действительности временной запас существенно меньше, поскольку в многосегментных сетях сами повторители вносят в распространение сигнала дополнительную задержку в несколько десятков битовых интервалов. Небольшой запас был сделан также для компенсации отклонений параметров кабеля и повторителей.

В табл. 3.1 представлены основные характеристики, налагаемые на параметры сети Ethernet и используемых сигналов.

Исторически первые сети Ethernet были созданы на коаксиальном кабеле диаметром 0,5 дюйма. В дальнейшем были определены и другие спецификации физического уровня для стандарта Ethernet, позволяющие задействовать различные среды передачи данных. Метод доступа

CSMA/CD и все временные параметры остаются одними и теми же для любой спецификации физической среды технологии Ethernet 10 Мбит/с.

Таблица 3.1

Основные параметры сети Ethernet

Параметры	Значения
Битовая скорость	10 Мбит/с
Интервал отсрочки	512 битовых интервала
Межкадровый интервал (IPG)	9,6 мкс
Максимальное число попыток передачи	16
Максимальное число возрастания диапазона паузы	10
Длина jam-последовательности	32 бит
Максимальная длина кадра (без преамбулы)	1518 байт
Минимальная длина кадра (без преамбулы)	64 байт (512 бит)
Длина преамбулы	64 бит
Минимальная длина случайной паузы после коллизии	0 битовых интервалов
Максимальная длина случайной паузы после коллизии	524 000 битовых интервала
Максимальное расстояние между станциями сети	2500 м
Максимальное число станций в сети	1024

Физические спецификации технологии Ethernet на сегодня включают следующие среды передачи данных:

- 10Base-5 — коаксиальный кабель диаметром 0,5 дюйма, называемый «толстым коаксиалом». Имеет волновое сопротивление 50 Ом. Максимальная длина сегмента 500 м (без повторителей).

- 10Base -2 — коаксиальный кабель диаметром 0,25 дюйма, называемый «тонким коаксиалом». Имеет волновое сопротивление 50 Ом. Максимальная длина сегмента 185 м (без повторителей).

- 10Base-T — кабель на основе неэкранированной витой пары (UTP). Образует звездообразную топологию на основе концентратора. Расстояние между концентратором и конечным узлом не более 100 м.

- 10Base-F — волоконно-оптический кабель. Топология аналогична топологии стандарта 10Base-T. Имеется несколько вариантов этой спецификации: FOIRL (расстояние до 1000 м), 10Base-FL (расстояние до 2000 м), 10Base-FB (расстояние до 2000 м).

Число 10 в указанных выше названиях обозначает номинальную битовую скорость передачи данных этих стандартов 10 Мбит/с, а слово «Base» — метод передачи на одной базовой частоте 10 МГц — в отличие от методов, использующих несколько несущих частот. Последний символ в названии стандарта физического уровня обозначает тип кабеля.

В табл. 3.2 представлены основные параметры физической среды для сетей, реализованных по технологии Ethernet.

Таблица 3.2

Параметры физической среды для сетей Ethernet

Параметр	10Base-5	10Base-2	10Base-T	10Base-F
Кабель	Толстый коаксиальный кабель RG-8 или RG-11	Тонкий коаксиальный кабель RG-58	Неэкранированная витая пара категорий 3, 4, 5	Многомодовый волоконно-оптический кабель
Максимальная длина сегмента	500 м	185 м	100 м	2000 м
Максимальное расстояние между узлами сети (с повторителями)	2500 м	925 м	500 м	2500 м (2740 м для 10Base-FB)
Максимальное число станций в сегменте	100	30	1024	1024
Максимальное число повторителей между станциями	4	4	4	4 (5 для 10 Base-FB)

Домен коллизий — это часть сети Ethernet, все узлы которой конкурируют за общую разделяемую среду передачи и, следовательно, каждый узел которой может создать коллизию с любым другим узлом этой части сети. Сеть Ethernet, построенная на повторителях, всегда образует один домен коллизий. Мосты, коммутаторы и маршрутизаторы разделяют сеть Ethernet на несколько доменов коллизий.

3.2. ТЕХНОЛОГИЯ FAST ETHERNET

В 1992 г. группа производителей сетевого оборудования, включая таких лидеров технологии Ethernet, как SynOptics, 3Com и ряд других, образовала некоммерческое объединение Fast Ethernet Alliance для разработки стандарта новой технологии, которая должна была обеспечить резкое повышение производительности при максимально возможном сохранении особенностей технологии Ethernet.

В комитете 802 института IEEE в это же время была сформирована исследовательская группа для изучения технического потенциала новых высокоскоростных технологий. За период с конца 1992 г. по конец 1993 г. группа IEEE изучила 100-мегабитные решения, предложенные различными производителями. Наряду с предложениями Fast Ethernet Alliance группа рассмотрела также и высокоскоростную технологию, предложенную компаниями Hewlett-Packard и AT&T.

Обсуждалась проблема сохранения метода случайного доступа CSMA/CD. Предложение Fast Ethernet Alliance сохраняло этот метод и, тем самым, обеспечивало преемственность и согласованность сетей 10 Мбит/с и 100 Мбит/с. Коалиция HP и AT&T, которая имела поддержку значительно меньшего числа производителей в сетевой индустрии, чем Fast Ethernet Alliance, предложила совершенно новый метод доступа, названный *приоритетным доступом по требованию* (demand priority). Он существенно менял картину поведения узлов в сети и, поэтому, не смог вписаться в технологию Ethernet и стандарт 802.3; для его стандартизации был организован новый комитет IEEE 802.12.

Осенью 1995 г. обе технологии стали стандартами IEEE. Комитет IEEE 802.3 принял спецификацию Fast Ethernet в качестве стандарта 802.3u, который не является самостоятельным стандартом, а представляет собой дополнение к существующему стандарту 802.3 в виде глав с 21 по

30. Комитет 802.12 принял технологию 100VG-AnyLAN, которая использовала принципиально новый метод доступа Demand Priority и поддерживала кадры двух форматов — Ethernet и Token Ring.

Все отличия технологий Fast Ethernet и Ethernet сосредоточены на физическом уровне (рис. 3.5). Уровни MAC и LLC в Fast Ethernet остались абсолютно теми же, и их описывают прежние главы стандартов 802.3 и 802.2. Физические варианты Fast Ethernet отличаются друг от друга в большей степени, чем варианты физической реализации Ethernet. Здесь меняется как количество проводников, так и методы кодирования. А так как физические варианты Fast Ethernet создавались одновременно, а не эволюционно, как для сетей Ethernet, имелась возможность детально определить и те подуровни физического уровня, которые не изменяются от варианта к варианту, и те подуровни, которые специфичны для каждого варианта физической среды.

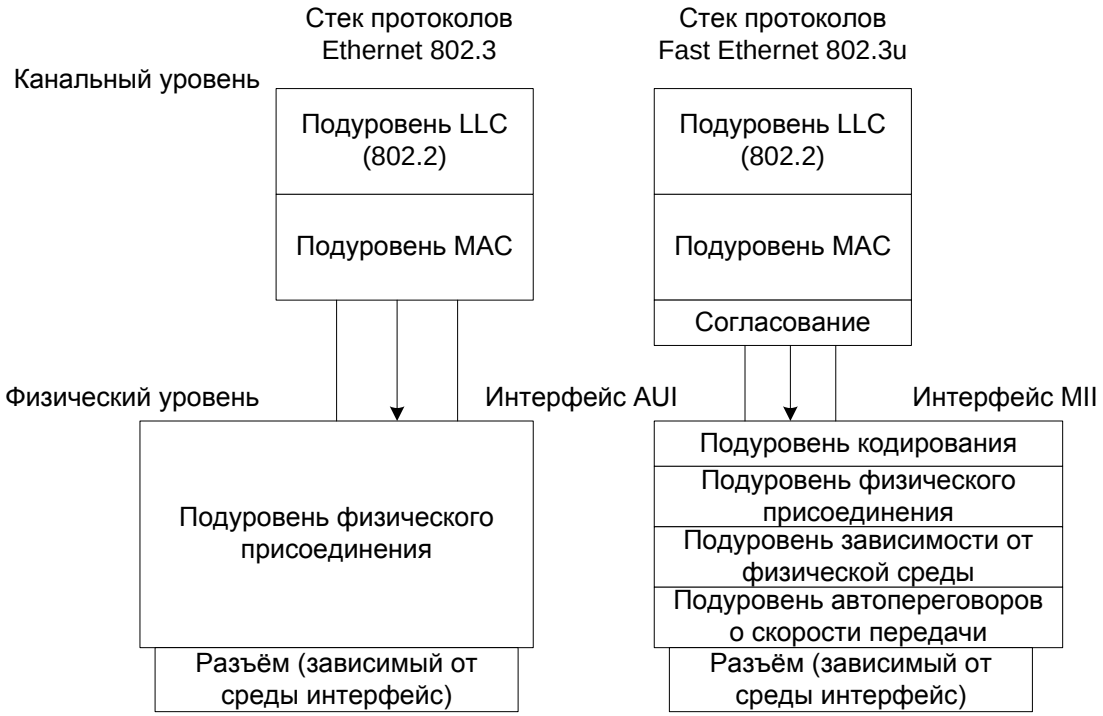


Рис. 3.5. Отличия технологий Fast Ethernet и Ethernet

Официальный стандарт 802.3 установил три различных спецификации для физического уровня Fast Ethernet и дал им следующие названия:

- 100Base-TX для двухпарного кабеля на неэкранированной витой па-

ре UTP категории 5 или экранированной витой паре STP типа 1;

- 100Base-T4 для четырехпарного кабеля на неэкранированной витой паре UTP категории 3, 4 или 5;

- 100Base-FX для многомодового оптоволоконного кабеля с двумя волокнами.

Для всех трех стандартов справедливы перечисленные ниже утверждения и характеристики.

Форматы кадров технологии Fast Ethernet не отличаются от форматов кадров технологий 10-мегабитного Ethernet.

Межкадровый интервал IPG равен 0,96 мкс, а битовый интервал равен 10 нс. Все временные параметры алгоритма доступа (интервал отсрочки, время передачи кадра минимальной длины и т. п.), измеренные в битовых интервалах, остались прежними, поэтому изменения в разделы стандарта, касающиеся уровня MAC, не вносились.

Признаком свободного состояния среды является передача по ней символа Idle соответствующего избыточного кода (а не отсутствие сигналов, как в стандартах Ethernet 10 Мбит/с).

Физический уровень технологии Fast Ethernet содержит три основных элемента.

1. Независимый от среды интерфейс (Media Independent Interface, МП).

2. Уровень согласования, необходимый для того, чтобы уровень MAC мог работать с физическим уровнем через интерфейс МП.

3. Устройство физического уровня (Physical Layer Device, PHY), состоящее, в свою очередь, из нескольких подуровней (рис. 3.5):

- подуровня логического кодирования данных, преобразующего поступающие от уровня MAC байты в символы кода 4B/5B или 8B/6T (оба кода используются в технологии Fast Ethernet);

- подуровней физического присоединения и зависимости от физической среды (PMD), которые обеспечивают формирование сигналов в соответствии с методом физического кодирования, например NRZI или MLT-3;

- подуровня автопереговоров, который позволяет двум взаимодействующим портам автоматически выбрать наиболее эффективный режим работы, например, полудуплексный или дуплексный (этот подуровень является опциональным).

Между спецификациями 100Base-FX, 100Base-TX и 100Base-T4 есть много общего, поэтому одинаковые для спецификаций свойства будут даваться под обобщенным названием, например 100Base-FX/TX или 100Base-TX/T4.

Спецификация 100Base-FX определяет работу протокола Fast Ethernet по многомодовому оптоволокну в полудуплексном и дуплексном режимах. В то время как в Ethernet со скоростью передачи 10 Мбит/с используется манчестерское кодирование для представления данных, в стандарте Fast Ethernet определен другой метод кодирования — 4B/5B, который к моменту разработки технологии Fast Ethernet уже показал свою эффективность в сетях FDDI, поэтому он без изменений был перенесен в спецификацию 100Base-FX/TX. В этом методе каждые 4 бита данных подуровня MAC (называемых символами) представляются 5 битами. Избыточный бит позволяет применить потенциальные коды при представлении каждого из пяти битов в виде электрических или оптических импульсов.

Существование запрещенных комбинаций символов позволяет отбраковывать ошибочные символы, что повышает устойчивость работы сетей с 100Base-FX/TX. Так, в Fast Ethernet признаком того, что среда свободна, стала повторяющаяся передача одного из запрещенных для кодирования пользовательских данных символа, а именно символа простоя источника Idle (11111). Такой способ позволяет приемнику всегда находиться в синхронизме с передатчиком.

Для отделения кадра Ethernet от символов простоя источника используется комбинация символов начального ограничителя кадра — пара символов J (11000) и K (10001) кода 4B/5B, а после завершения кадра перед первым символом простоя источника вставляется символ T (рис. 3.6).

После преобразования 4-битовых порций кодов MAC в 5-битовые порции физического уровня их необходимо представить в виде оптических или электрических сигналов в кабеле, соединяющем узлы сети. В спецификациях 100Base-FX и 100Base-TX для этого используются различные методы физического кодирования — NRZI и MLT-3 соответственно.

В спецификации 100Base-TX качестве среды передачи данных используется витая пара UTP категории 5 или STP типа 1. Основным отличием от спецификации 100Base-FX — наряду с использованием метода кодирования MLT-3 — является наличие функции автопереговоров для выбора ре-

жима работы порта. Такая схема позволяет двум физически соединенным устройствам, которые поддерживают несколько стандартов физического уровня, отличающихся битовой скоростью и количеством витых пар, согласовать наиболее выгодный режим работы. Обычно процедура автопереговоров происходит при подсоединении сетевого адаптера, который может работать на скоростях 10 и 100 Мбит/с, к концентратору или коммутатору.

Преамбула Idle	JK	Преамбула	SFD	DA	SA	L	Данные	CRC	T	Преамбула Idle
Первый байт							JK — ограничитель начала потока данных T — ограничитель конца потока данных			

Рис. 3.6. Непрерывный поток данных спецификаций 100Base-FX/TX

Спецификация 100Base-T4 (витая пара UTP категории 3, четыре пары) появилась позже других спецификаций физического уровня Fast Ethernet. Разработчики технологий 100Base-TX/FX, появившихся ранее, в первую очередь хотели создать физические спецификации, наиболее близкие к спецификациям 10Base-T и 10Base-F, которые работали на двух линиях передачи данных: двух парах или двух волокнах. Для реализации работы по двум витым парам пришлось перейти на более качественный кабель категории 5. Вместо кодирования 4B/5B в этом методе используется кодирование 8B/6T, которое обладает более узким спектром сигнала и при скорости 33 Мбит/с укладывается в полосу 16 МГц витой пары категории 3 (при кодировании 4B/5B спектр сигнала в эту полосу не укладывается). Каждые 8 бит информации уровня MAC кодируются шестью троичными цифрами, т. е. цифрами, имеющими три состояния. Каждая троичная цифра имеет длительность 40 нс. Затем такие группы из шести троичных цифр передается на одну из трех передающих витых пар независимо и последовательно. Четвертая пара всегда используется для прослушивания несущей частоты в целях обнаружения коллизий. Скорость передачи данных по каждой из трех передающих пар равна 33,3 Мбит/с, поэтому общая скорость протокола 100Base-T4 составляет 100 Мбит/с. В то же время, из-за принятого способа кодирования скорость изменения сигнала на каждой паре равна

всего 25 МБод, что, собственно, и позволяет использовать витую пару категории 3.

3.3. ТЕХНОЛОГИЯ GIGABIT ETHERNET

Достаточно быстро после появления на рынке продуктов Fast Ethernet сетевые интеграторы и администраторы при построении корпоративных сетей почувствовали определенные ограничения. Во многих случаях серверы, подключенные по 100-мегабитному каналу, перегружали магистрали сетей, работающих также на скорости 100 Мбит/с — магистрали FDDI и Fast Ethernet. Ощущалась потребность в следующем уровне иерархии скоростей. В 1995 г. более высокий уровень скорости могли предоставить только коммутаторы ATM, которые из-за высокой стоимости, а также больших отличий от классических технологий применялись в локальных сетях достаточно редко. Летом 1996 г. было объявлено о создании группы 802.3z для разработки протокола, в максимальной степени подобного Ethernet, но с битовой скоростью 1000 Мбит/с. Стандарт 802.3z был окончательно принят в 1998 г. Работы по реализации Gigabit Ethernet на витой паре категории 5 были переданы проблемной группе 802.3ab ввиду сложности обеспечения гигабитной скорости на этом типе кабеля, который был создан для поддержки скоростей 100 Мбит/с. Проблемная группа 802.3ab успешно справилась со своей задачей, и версия Gigabit Ethernet для витой пары категории 5 была также принята.

В стандарте 802.3z определены следующие типы физической среды:

- одномодовый волоконно-оптический кабель;
- многомодовый волоконно-оптический кабель 62,5/125;
- многомодовый волоконно-оптический кабель 50/125;
- экранированный сбалансированный медный кабель.

Для передачи данных по традиционному для компьютерных сетей многомодовому волоконно-оптическому кабелю стандарт определяет применение излучателей, работающих на двух длинах волн: 1300 и 850 нм. Применение светодиодов с длиной волны 850 нм объясняется тем, что они намного дешевле, чем светодиоды, работающие на волне 1300 нм, хотя при этом максимальная длина кабеля уменьшается, так как затухание многомодового оптоволокна на волне 850 нм более чем в два раза выше, чем на

волне 1300 нм. Однако возможность удешевления чрезвычайно важна для такой в целом дорогой технологии, как Gigabit Ethernet.

Для многомодового оптоволокна стандарт 802.3z определяет спецификации 1000Base-SX и 1000Base-LX. В первом случае используется длина волны 850 нм (символ S означает Short Wavelength), а во втором — 1300 нм (L — Long Wavelength). Спецификация 1000Base-SX может использовать только многомодовый кабель, при этом его максимальная длина составляет около 500 м. Для спецификации 1000Base-LX в качестве источника излучения всегда применяется полупроводниковый лазерный диод с длиной волны 1300 нм. Спецификация 1000Base-LX может работать как с многомодовым (максимальное расстояние до 500 м), так и с одномодовым кабелем (максимальное расстояние зависит от мощности передатчика и качества кабеля и может доходить до нескольких десятков километров).

Как известно, каждая пара кабеля категории 5 имеет гарантированную полосу пропускания до 100 МГц. Для передачи по такому кабелю данных со скоростью 1000 Мбит/с было решено организовать параллельную передачу одновременно по всем 4 парам кабеля. Это сразу снизило скорость передачи данных по каждой паре до 250 Мбит/с. Однако и для такой скорости необходим метод кодирования, при котором ширина спектра не превышает 100 МГц. Например, применение кода 4B/5B не может решить поставленную задачу, так как основной вклад в спектр сигнала на такой скорости вносит частота 155 МГц. Кроме того, не нужно забывать, что каждая новая технология должна поддерживать не только классический полудуплексный режим, но также и полноценный дуплексный режим. На первый взгляд кажется, что одновременное использование четырех пар лишает сеть возможность работы в дуплексном режиме, так как не остается свободных пар для одновременной передачи данных в двух направлениях. На эти вопроса проблемная группа 802.3ab нашла следующие приемлемые решения.

Для кодирования данных был применен код PAM5, имеющий пять потенциальных уровней: –2, –1, 0, +1, +2, так что за один тактовый интервал по одной паре передается $\log_2 5 = 2{,}322$ бит информации. Следовательно, для достижения скорости 250 Мбит/с тактовую частоту 250 МГц можно уменьшить в 2,322 раза. Разработчики стандарта решили использовать несколько более высокую частоту, а именно 125 МГц, при которой код

PAM5 имеет спектр уже, чем 100 МГц, т. е. сигналы могут быть переданы без искажений по кабелю категории 5. Таким образом, в каждом такте передается не 2,322 × 4 = 9,288 бит информации, а 8, что и дает искомую суммарную скорость 1000 Мбит/с. Передача ровно 8 бит в каждом такте достигается за счет того, что при кодировании информации используются не все $5^4 = 625$ комбинаций кода PAM5, а только $2^8 = 256$. Оставшиеся комбинации приемник использует для контроля принимаемой информации и выделения правильных комбинаций на фоне шума.

Для организации дуплексного режима разработчики спецификации 802.3ab применили технику выделения принимаемого сигнала из суммарного. Два передатчика работают навстречу друг другу по каждой из 4-х пар в одном и том же диапазоне частот (рис. 3.7). Н-образная схема гибридной развязки позволяет приемнику и передатчику одного и того же узла использовать одновременно витую пару и для приема, и для передачи (также, как и в приёмопередатчиках Ethernet на коаксиале).

Для отделения принимаемого сигнала от собственного приемник вычитает из результирующего сигнала известный ему свой сигнал, для чего используются специальные процессоры цифровой обработки сигналов (Digital Signal Processor, DSP).

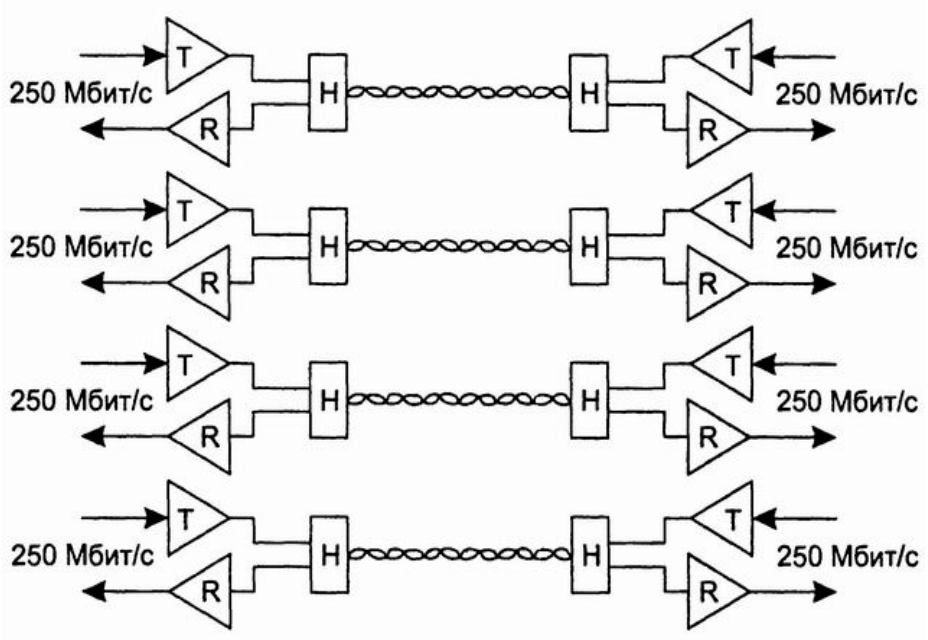


Рис. 3.7. Двунаправленная передача по четырем парам UTP категории 5

3.4. ТЕХНОЛОГИЯ TOKEN RING

Технология Token Ring была разработана компанией IBM в 1984 г., а затем передана в качестве проекта стандарта в комитет IEEE 802, который на ее основе принял в 1985 г. стандарт 802.5. Компания IBM в течение долгого времени использовала технологию Token Ring как свою основную сетевую технологию построения локальных сетей на основе компьютеров различных классов — мэйнфреймов, миникомпьютеров и персональных компьютеров. Однако в последнее время даже в продукции компании IBM доминируют представители семейства Ethernet.

Сети Token Ring работают с двумя битовыми скоростями: 4 и 16 Мбит/с. Смешение в одном кольце станций, работающих на разных скоростях, не допускается. Сети Token Ring, работающие со скоростью 16 Мбит/с, имеют некоторые усовершенствования в алгоритме доступа по сравнению со стандартом 4 Мбит/с.

Технология Token Ring сложнее, чем Ethernet. Она обладает некоторыми начальными свойствами отказоустойчивости. В сети Token Ring определены процедуры контроля работы сети, которые опираются на свойство обратной связи, изначально присущее кольцеобразной структуре — посланный кадр всегда возвращается к станции-отправителю.

Для контроля сети одна из станций исполняет роль так называемого активного монитора. Активный монитор выбирается во время инициализации кольца, критерием выбора служит максимальное значение MAC-адреса. Если активный монитор выходит из строя, то процедура инициализации повторяется, и выбирается новый активный монитор. Чтобы сеть могла обнаружить отказ активного монитора, последний в работоспособном состоянии каждые 3 с генерирует специальный кадр, обозначающий его присутствие. Если этот кадр не появляется в сети более 7 с, то остальные станции сети начинают процедуру выборов нового активного монитора.

Работа станций начинается с передачей *токена* — специального опознавательного сигнала. Передача токена обычно реализуется децентрализовано. Каждый компьютер, получивший токен, имеет право на использование разделяемой среды в течение фиксированного промежутка времени — времени удержания токена, в течение которого станция передает свои кадры. После истечения этого промежутка станция обязана передать

токен другой станции. Таким образом, если известно количество компьютеров в сети, то максимальное время ожидания доступа равно произведению времени удержания токена на это число. Заметим, что реальное время ожидания может быть и меньше, поскольку, если компьютер, получивший токен, не имеет кадров для передачи, то он передает его следующему компьютеру, не дожидаясь истечения времени удержания. Последовательность передачи токена от компьютера к компьютеру может определяться разными способами. В сетях Token Ring и FDDI она определяется топологией связей. Компьютер в кольце получает токен от предыдущего соседа, а передает токен следующему.

Алгоритмы детерминированного доступа отличаются от алгоритмов случайного доступа тем, что они более эффективно работают при большой загрузке сети, когда коэффициент использования приближается к единице. В то же время, при небольшой загрузке сети более эффективными являются алгоритмы случайного доступа, так как они позволяют передать кадр немедленно, не тратя время на процедуры определения права доступа к среде. Достоинство детерминированных методов доступа также заключается в том, что они могут приоритезировать трафик, а значит, поддерживать требования QoS.

Получив токен, станция анализирует его и при отсутствии у нее данных для передачи продвигает токен к следующей станции. Станция, которая имеет данные для передачи, при получении токена изымает его из кольца, что дает ей право доступа к физической среде для передачи своих данных. Затем эта станция выдает в кольцо кадр данных установленного формата последовательно по битам. Кадр снабжается адресами приемника и источника.

Переданные данные проходят по кольцу всегда в одном направлении от одной станции к другой. Все станции кольца ретранслируют кадр побитно как повторители. Если кадр проходит через станцию назначения, то, распознав свой адрес, эта станция копирует кадр в свой внутренний буфер и вставляет в кадр признак подтверждения приема. Станция, выдавшая кадр данных в кольцо, получив его с подтверждением приема, изымает свой кадр из кольца и передает в сеть новый токен, давая другим станциям сети возможность передавать данные.

Обычно время удержания токена по умолчанию равно 10 мс, а максимальный размер кадра в стандарте 802.5 не определен. Для сетей 4 Мбит/с он, как правило, равен 4 Кбайт, а для сетей 16 Мбит/с — 16 Кбайт. Это связано с тем, что за время удержания токена станция должна успеть передать хотя бы один кадр. При скорости 4 Мбит/с за время 10 мс можно передать 5000 байт, а при скорости 16 Мбит/с — 20 000 байт. Максимальные размеры кадра выбраны с некоторым запасом.

В сетях Token Ring 16 Мбит/с используется модернизированный вариант алгоритма доступа к кольцу, называемый *алгоритмом раннего освобождения токена*. В соответствии с ним станция передает токен доступа следующей станции сразу же после окончания передачи последнего бита кадра, не дожидаясь возвращения по кольцу этого кадра с установленными битами A и C. В этом случае пропускная способность кольца используется более эффективно, так как по кольцу одновременно продвигаются кадры нескольких станций. Тем не менее, свои кадры в каждый момент времени может генерировать только одна станция — та, которая в данный момент владеет токеном. Остальные станции в это время только повторяют чужие кадры, так что принцип разделения кольца во времени сохраняется, ускоряется только процедура передачи владения кольцом. Описанный алгоритм доступа к среде иллюстрируется временной диаграммой на рис. 3.8.

Здесь показана передача пакета A в кольце, состоящем из 6 станций, от станции 1 к станции 3. После прохождения станции назначения 3 в пакете A устанавливаются два признака: признак *A* распознавания адреса и признак *C* копирования пакета в буфер (что на рисунке отмечено звездочкой внутри пакета). После возвращения пакета в станцию 1 отправитель распознает свой пакет по адресу источника и удаляет пакет из кольца. Установленные станцией 3 признаки говорят станции-отправителю о том, что пакет дошел до адресата и был успешно скопирован в его буфер.

Для различных видов сообщений, передаваемых кадрами, могут назначаться различные приоритеты, от 0 (низший) до 7 (высший). Решение о приоритете конкретного кадра принимает передающая станция (протокол Token Ring получает этот параметр через межуровневые интерфейсы от протоколов верхнего уровня, например, прикладного). Токен также всегда имеет некоторый уровень текущего приоритета. Станция имеет право захватить переданный ей токен только в том случае, если приоритет кадра,

который она хочет передать, выше приоритета токена (или равен ему). В противном случае станция обязана передать токен следующей по кольцу станции.

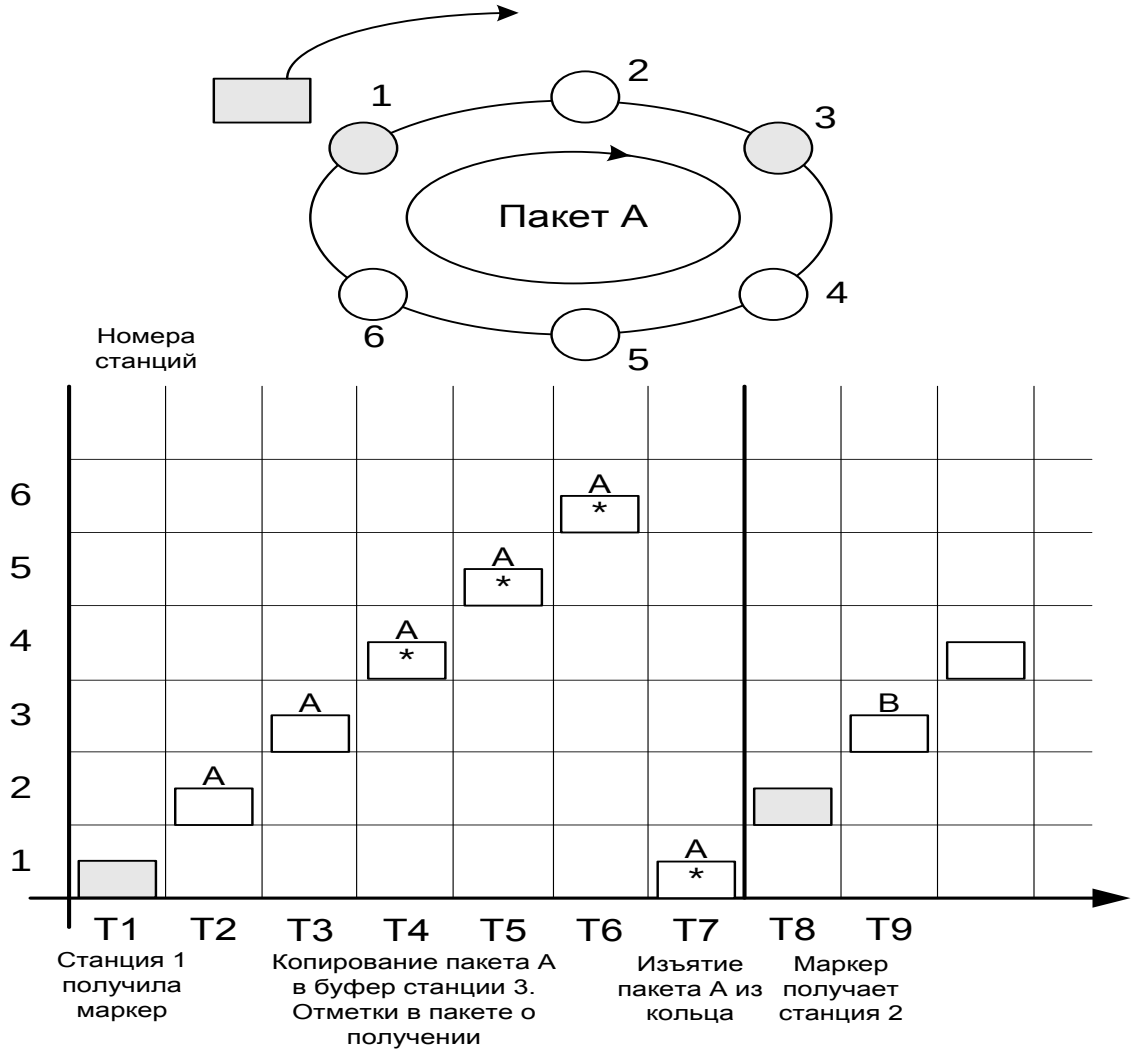


Рис. 3.8. Доступ с передачей токена

За наличие в сети токена, причем единственной его копии, отвечает активный монитор. Если активный монитор не получает токен в течение длительного времени (например, 2,6 с), то он порождает новый токен.

Стандарт Token Ring фирмы IBM изначально предусматривал построение связей в сети с помощью концентраторов (рис. 3.9), называемых *устройствами многостанционного доступа* (Multi-station Access Unit, MAU, или MSAU). Сеть Token Ring может включать до 260 узлов. Использование концентраторов приводит к тому, что сети Token Ring имеют физическую топологию звезда, а логическую — кольцо.

Концентратор сети Token Ring может быть активным или пассивным.

Пассивный концентратор просто соединяет порты внутренними связями так, чтобы станции, подключаемые к этим портам, образовали кольцо. Ни усиление сигналов, ни их ресинхронизацию пассивный концентратор не выполняет. Такое MSAU-устройство можно считать простым кроссовым блоком за одним исключением: MSAU обеспечивает обход какого-либо порта, когда присоединенный к этому порту компьютер выключается. Такая функция необходима для обеспечения связности кольца вне зависимости от состояния подключенных компьютеров. Обычно обход порта выполняется за счет релейных схем, которые питаются постоянным током от сетевого адаптера, а при выключении сетевого адаптера нормально замкнутые контакты реле соединяют вход порта с его выходом.

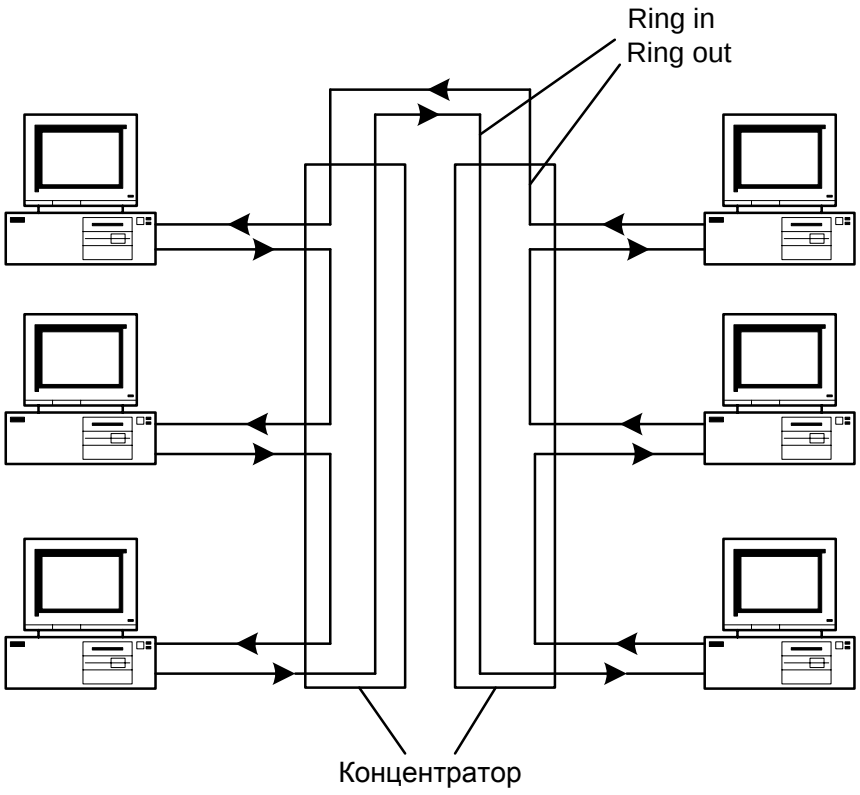


Рис. 3.9 Физическая конфигурация сети Token Ring

Активный концентратор выполняет функции регенерации сигналов и поэтому его можно назвать повторителем.

Технология Token Ring позволяет использовать для соединения конечных станций и концентраторов различные типы кабеля: STP типа 1,

UTP типа 3, UTP типа 6, а также волоконно-оптический кабель. При использовании экранированной витой пары STP типа 1 из номенклатуры кабельной системы IBM в кольцо допускается объединять до 260 станций при длине ответвительных кабелей до 100 м, а при использовании неэкранированной витой пары максимальное количество станций сокращается до 72 при длине ответвительных кабелей до 45 м. Расстояние между пассивными концентраторами может достигать 100 м при использовании кабеля STP типа 1 и 45 м при использовании кабеля UTP типа 3. Между активными концентраторами максимальное расстояние увеличивается соответственно до 730 или 365 м в зависимости от типа кабеля.

3.5. ТЕХНОЛОГИЯ FDDI

Технология FDDI (Fiber Distributed Data Interface, распределенный интерфейс передачи данных по оптоволокну) — это первая технология локальных сетей, в которой в качестве среды передачи данных стал применяться волоконно-оптический кабель. Работы по созданию технологий и устройств локальных сетей, использующих волоконно-оптические каналы, начались в 1980-е гг. вскоре после начала промышленной эксплуатации подобных каналов в территориальных сетях. Проблемная группа X3T9.5 института ANSI разработала в период 1986–1988 гг. начальные версии стандарта FDDI, который описывает передачу со скоростью 100 Мбит/с по двойному волоконно-оптическому кольцу длиной до 100 км.

Технология FDDI во многом основывается на технологии Token Ring, развивая и совершенствуя ее основные идеи. Разработчики технологии FDDI ставили перед собой следующие цели:

- повысить битовую скорость передачи данных до 100 Мбит/с;

- повысить отказоустойчивость сети за счет стандартных процедур восстановления ее после отказов различного рода: повреждения кабеля, некорректной работы узла, концентратора, возникновения высокого уровня помех на линии и т. п.;

- максимально эффективно использовать потенциальную пропускную способность сети как для асинхронного, так и для синхронного (чувствительного к задержкам) трафиков.

Сеть FDDI строится на основе двух оптоволоконных колец, которые образуют основной и резервный пути передачи данных между узлами сети.

Наличие двух колец — это основное средство повышения отказоустойчивости в сети FDDI.

Узлы, которые хотят воспользоваться этим повышенным потенциалом надежности, должны быть подключены к обоим кольцам. В технологии FDDI для передачи световых сигналов по оптическим волокнам реализовано кодирование 4B/5B в сочетании с кодированием NRZI, что приводит к передаче по линии связи сигналов с тактовой частотой 125 МГц.

В нормальном режиме работы сети данные проходят через все узлы и все участки кабеля только первичного кольца, этот режим назван *сквозным*, или *транзитным*. Вторичное кольцо в этом режиме не используется. В случае какого-либо вида отказа, когда часть первичного кольца не может передавать данные (например, обрыв кабеля или отказ узла), первичное кольцо объединяется со вторичным (рис. 3.10), вновь образуя единое кольцо. Этот режим работы сети называется *режимом свертывания колец*.

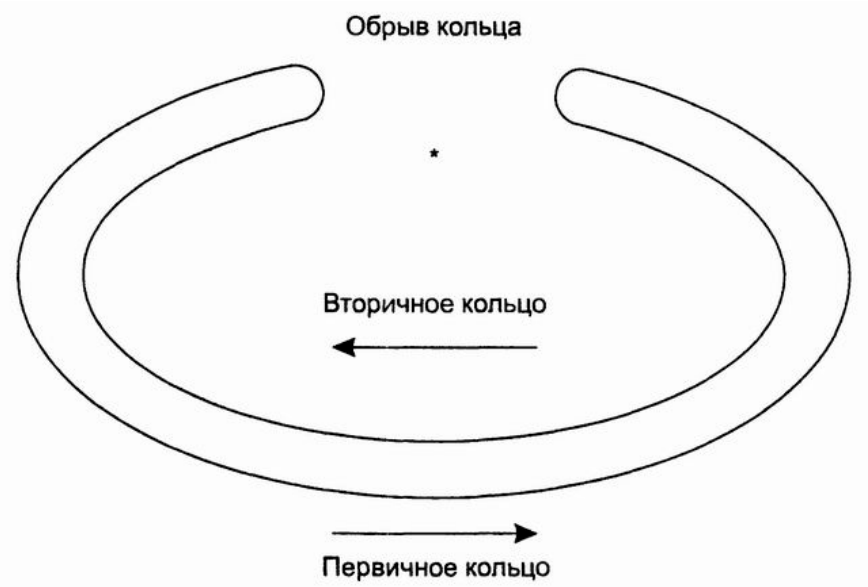


Рис. 3.10. Реконфигурация колец FDDI при отказе

Операция свертывания производится средствами концентраторов и/или сетевых адаптеров FDDI. Для упрощения этой процедуры данные по первичному кольцу всегда передаются в одном направлении (на рисунке это направление изображается против часовой стрелки), а по вторичному — в обратном. Поэтому при образовании общего кольца из двух колец передатчики станций по-прежнему остаются подключенными к приемни-

кам соседних станций, что позволяет правильно передавать и принимать информацию соседними станциями.

В стандартах FDDI много внимания отводится различным процедурам, которые позволяют определить наличие отказа в сети, а затем произвести необходимое реконфигурацию. Технология FDDI дополняет механизмы обнаружения отказов технологии Token Ring механизмами реконфигурирования пути передачи данных в сети, основанными на наличии резервных связей, которые предоставляет второе кольцо.

Кольца в сетях FDDI рассматриваются как общая разделяемая среда передачи данных. Метод доступа к этой среде очень близок к методу доступа сетей Token Ring. Станции FDDI применяют алгоритм раннего освобождения токена, как и сети Token Ring 16 Мбит/с.

Отличия в методах доступа сетях Token Ring и FDDI заключаются в том, что время удержания токена в сети FDDI не является постоянной величиной, как в сети Token Ring, а зависит от загрузки кольца: при небольшой загрузке оно растет, а при перегрузках может снижаться до нуля. Однако эти изменения касаются только асинхронного трафика, который не критичен к небольшим задержкам передачи кадров. Для синхронного трафика время удержания токена по-прежнему остается фиксированной величиной.

Механизм приоритетов кадров, принятый в Token Ring, в технологии FDDI отсутствует. Разработчики технологии решили, что деление трафика на 8 уровней приоритетов избыточно, достаточно разделить трафик на два класса: асинхронный и синхронный, последний из которых обслуживается всегда, даже при перегрузках кольца.

В остальном пересылка кадров между станциями кольца FDDI на уровне MAC полностью соответствует технологии Token Ring.

Как уже отмечалось, для обеспечения отказоустойчивости в стандарте FDDI предусмотрено создание двух оптоволоконных колец — первичного и вторичного. В стандарте FDDI определены два типа конечных узлов: станции и концентраторы. Для подключения станций и концентраторов к сети может быть использован один из двух возможных способов.

Двойное подключение (Dual Attachment, DA) — одновременное подключение к первичному и вторичному кольцам; станция и концентратор, подключенные таким способом, называются соответственно станцией

двойного подключения (Dual Attachment Station, DAS) и концентратором двойного подключения (Dual Attachment Concentrator, DAC).

Одиночное подключение (Single Attachment, SA) — подключение только к первичному кольцу; станция и концентратор, подключенные данным способом, называются соответственно станцией одиночного подключения (Single Attachment Station, SAS) и концентратором одиночного подключения (Single Attachment Concentrator, SAC).

В случае однократного обрыва кабеля между устройствами с двойным подключением сеть FDDI сможет продолжить нормальную работу за счет автоматической реконфигурации внутренних путей передачи кадров между портами концентратора. Двукратный обрыв кабеля приведет к образованию двух изолированных сетей FDDI.

3.6. БЕСПРОВОДНЫЕ ЛОКАЛЬНЫЕ СЕТИ. ОБЩИЙ ВЗГЛЯД

Беспроводные локальные сети сегодня рассматриваются как дополнение к проводным сетям, а не как конкурентное решение. Отношение к беспроводным локальным сетям по мере их развития менялось; в середине 1990-х гг. было популярно мнение, в соответствии с которым все большее число локальных сетей будет переходить на беспроводные технологии. Преимущество беспроводных локальных сетей очевидно: их проще и дешевле разворачивать и модифицировать, так как вся громоздкая кабельная и инфраструктура оказывается излишней; кроме того, обеспечивается связь между мобильными пользователями. Однако за эти преимущества беспроводных сетей приходится расплачиваться большим количеством проблем, которые несет с собой неустойчивая и непредсказуемая беспроводная среда, поскольку помехи от разнообразных бытовых приборов и других телекоммуникационных систем, атмосферные помехи и отражения сигнала создают большие трудности для надежного приема информации.

Методы расширения спектра помогают снизить влияние помех на полезный сигнал, кроме того, в беспроводных сетях широко используются прямая коррекция ошибок (Forward Error Control, FEC) и протоколы с повторной передачей потерянных кадров. Тем не менее, практика показала, что в тех случаях, когда ничего не мешает применению проводной локаль-

ной сети, организации предпочитают именно этот вид LAN, несмотря на то, что при этом нельзя обойтись без кабельной системы.

В примере на рис. 3.11, *а* показана фрагментированная локальная сеть. Неполносвязность беспроводной сети порождает проблему доступа к разделяемой среде, известную под названием «скрытого терминала». Проблема возникает в том случае, когда два узла находятся вне зон досягаемости друг друга (узлы A и C на рис. 3.11, *а*), но существует третий узел B, который принимает сигналы как от A, так и от C. Предположим, что в радиосети используется традиционный метод доступа, основанный на прослушивании несущей, например CSMA/CD. В данном случае коллизии будут возникать значительно чаще, чем в проводных сетях. Пусть, например, узел B занят обменом с узлом A. Узлу C сложно определить, что среда занята, он может посчитать ее свободной и начать передавать свой кадр. В результате сигналы в районе узла B будут искажены, т. е. произойдет коллизия, вероятность возникновения которой в проводной сети была бы неизмеримо ниже.

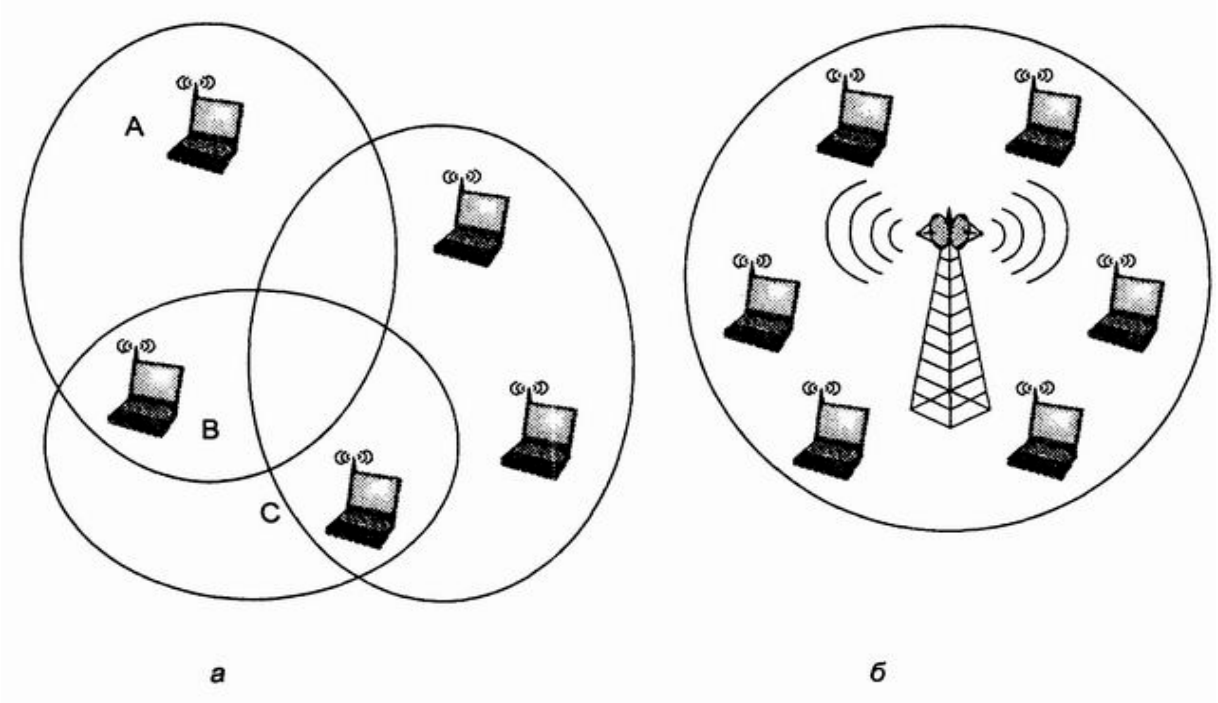


Рис. 3.11. Связность беспроводной локальной сети: *а* — специализированная беспроводная сеть; *б* — беспроводная сеть с базовой станцией

Как следствие подобных проблем, в методах доступа в беспроводных сетях не только отказываются от прослушивания несущей частоты, но и от распознавания коллизий, а вместо этого используются методы предотвращения коллизий, в том числе и методы опроса.

Применение базовой станции может улучшить связность сети (рис. 3.11, *б*). Базовая станция обычно обладает большей мощностью, а ее антенна устанавливается так, чтобы более равномерно и беспрепятственно покрывать нужную территорию. В результате все узлы беспроводной локальной сети получают возможность обмениваться данными с базовой станцией, которая транзитом передает данные между узлами.

Беспроводные локальные сети считаются перспективными для таких применений, в которых сложно или невозможно использовать проводные сети. Ниже перечислены основные области применения беспроводных локальных сетей.

- Резидентный доступ альтернативных операторов связи, у которых нет проводного доступа к клиентам, проживающим в многоквартирных домах.

- Так называемый «кочевой» доступ в аэропортах, железнодорожных вокзалах и т. п.

- Организация локальных сетей в зданиях, где нет возможности установить современную кабельную систему, например, в исторических зданиях с оригинальным интерьером.

- Организация временных локальных сетей, например, при проведении конференций.

- Расширения локальных сетей. Иногда одно здание предприятия, например испытательная лаборатория или цех, может быть расположено изолированно от других. Небольшое число рабочих мест в таком здании делает крайне невыгодным прокладку к нему отдельного кабеля, поэтому беспроводная связь оказывается более рациональным вариантом.

- Мобильные локальные сети. Если пользователь хочет пользоваться услугами сети, перемещаясь из помещения в помещение или из здания в здание, то здесь конкурентов у беспроводной локальной сети просто нет.

Первым общим стандартом, описывающим принципы построения и функционирования беспроводных локальных сетей, оказался стандарт

IEEE 802.11, на базе которого в дальнейшем были созданы многочисленные дополнения и развития.

Естественно, что стек протоколов стандарта IEEE 802.11 соответствует общей структуре стандартов комитета 802, т. е. состоит из физического уровня и уровня MAC, над которыми работает уровень LLC. Как и у всех технологий семейства 802, технология 802.11 определяется нижними двумя уровнями (физическим уровнем и уровнем MAC), а уровень LLC выполняет свои стандартные общие для всех технологий LAN функции. Так как искажения кадров в беспроводной среде более вероятны, чем в проводной, уровень LLC должен, скорее всего, использоваться в режиме LLC2, но это уже не зависит от конкретной технологии 802.11. Структура стека протоколов IEEE 802.11 показана на рис. 3.12.

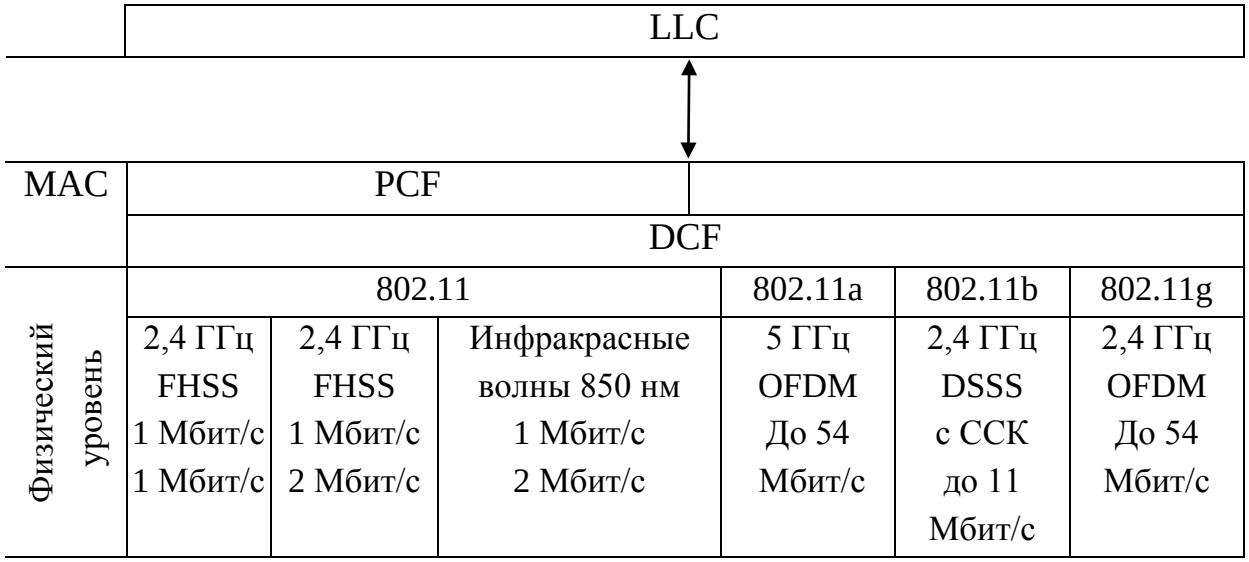


Рис. 3.12. Стек протоколов IEEE 802.11

Уровень MAC выполняет в беспроводных сетях больше функций, чем в проводных сетях. Функции уровня MAC в стандарте 802.11 включают:

- доступ к разделяемой среде;

- обеспечение мобильности станций при наличии нескольких базовых станций;

- обеспечение безопасности, эквивалентной безопасности проводных локальных сетей.

Рассмотрим возможные топологии, используемые при построении беспроводных локальных сетей семейства 802.11.

Станции могут использовать разделяемую среду для того, чтобы передавать данные следующим образом:

- непосредственно друг другу в пределах одной BSS-сети;
- в пределах одной BSS-сети транзитом через точку доступа;
- между разными BSS-сетями через две точки доступа и распределенную систему;
- между BSS-сетью и проводной локальной сетью через точку доступа, распределенную систему и портал.

Стандарт 802.11 поддерживает два типа топологий локальных сетей: с базовым и с расширенным наборами услуг.

Сеть с базовым набором услуг (Basic Service Set, BSS) образуется отдельными станциями, базовая станция отсутствует, узлы взаимодействуют друг с другом непосредственно (рис. 3.13). Для того чтобы войти в BSS-сеть, станция должна выполнить процедуру присоединения.

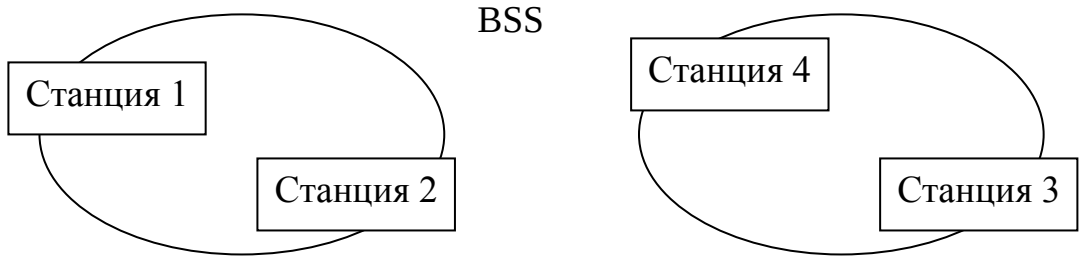


Рис. 3.13. Сети с базовым набором услуг

BSS-сети не являются традиционными сотами в отношении зон покрытия, они могут находиться друг от друга на значительном расстоянии, а могут частично или полностью перекрываться — стандарт 802.11 оставляет здесь свободу для проектировщика сети.

В сетях, обладающих заметно разветвленной инфраструктурой, некоторые станции сети являются базовыми, или, по терминологии 802.11, *точками доступа* (Access Point, AP). Все базовые станции сети связаны между собой с помощью распределенной системы (Distribution System, DS), в качестве которой может использоваться та же среда (т. е. радио- или

инфракрасные волны), что и для взаимодействия между станциями, или же отличная от нее, например проводная.

Точки доступа вместе с распределенной системой (рис. 3.14) поддерживают службу распределенной системы (Distribution System Service, DSS). Задачей DSS является передача пакетов между станциями, которые по каким-то причинам не могут или не хотят взаимодействовать между собой непосредственно. Наиболее очевидной причиной использования DSS является принадлежность станций разным BSS-сетям. В этом случае они передают кадр своей точке доступа, которая через DS передает его точке доступа, обслуживающей BSS-сеть со станцией назначения.

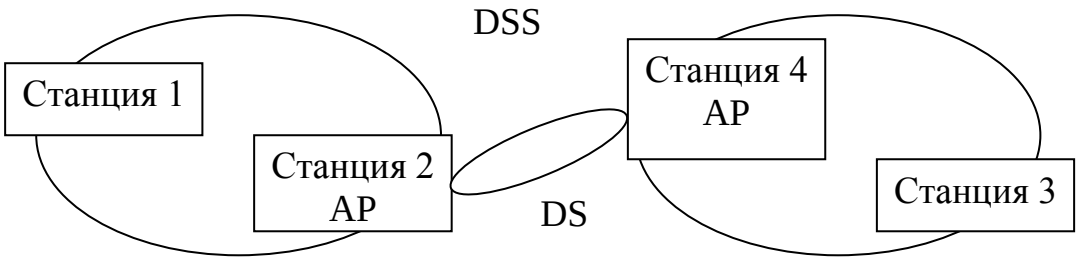


Рис. 3.14. Сети, поддерживающие службу распределенной системы

Сеть с расширенным набором услуг (Extended Service Set, ESS) состоит из нескольких BSS-сетей, объединённых распределённой средой. Каждая ESS-сеть обеспечивает станциям мобильность — они могут переходить из одной BSS-сети в другую. Эти перемещения обеспечиваются функциями уровня MAC рабочих и базовых станций, поэтому они являются прозрачными для уровня LLC. ESS-сеть может также взаимодействовать с проводной локальной сетью, для чего в распределенной системе должен присутствовать портал.

3.7. ТЕХНОЛОГИЯ 10G ETHERNET

Формально этот стандарт имеет обозначение IEEE 802.3ae и является поправкой к основному тексту стандарта 802.3, описывающей семь новых спецификаций физического уровня, которые взаимодействуют с уровнем MAC с помощью нового варианта подуровня согласования (рис. 3.17).

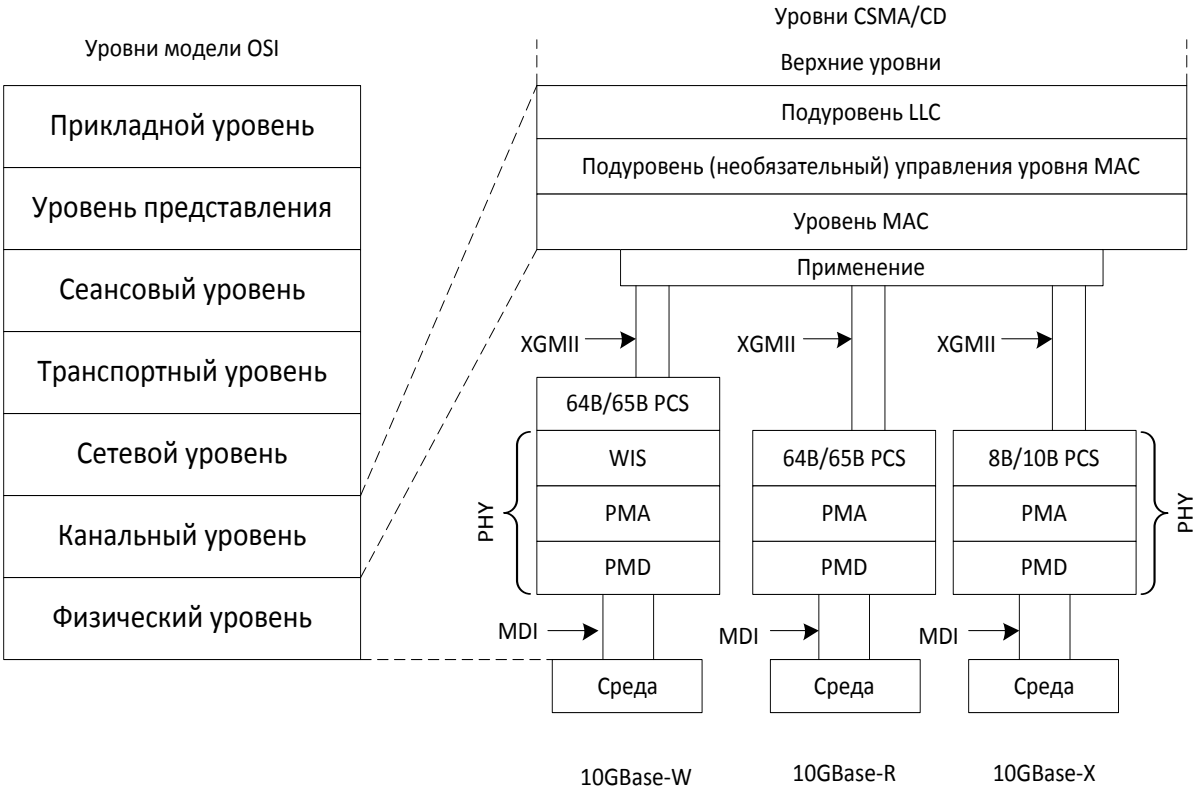


Рис. 3.17. Три группы физических интерфейсов 10G

Этот подуровень обеспечивает для всех вариантов физического уровня 10G Ethernet единый интерфейс XGMII (extended Gigabit Medium Independent Interface, расширенный интерфейс независимого доступа к гигабитной среде), который предусматривает параллельный обмен четырьмя байтами, образующими четыре потока данных.

Как видно из рис. 3.17, существуют три группы физических интерфейсов стандарта 10G Ethernet: 10GBase-X, 10GBase-R и 10GBase-W. Они отличаются способом кодирования данных: в варианте 10GBase-X используется код 8B/10B, а в остальных двух — код 64B/65B. Все они задействуют оптическую среду для передачи данных.

Группа 10GBase-X в настоящее время состоит из одного интерфейса подуровня PMD — 10GBase-LX4. Символ L говорит о том, что информация передается с помощью волн второго диапазона прозрачности, то есть 1310 нм. Информация в каждом направлении передается одновременно с помощью четырех волн (что отражает цифра 4 в названии интерфейса), ко-

торые мультиплексируются на основе техники WDM (рис. 3.18). Каждый из четырех потоков интерфейса XGMII передается в оптическом волокне со скоростью 2,5 Гбит/с.

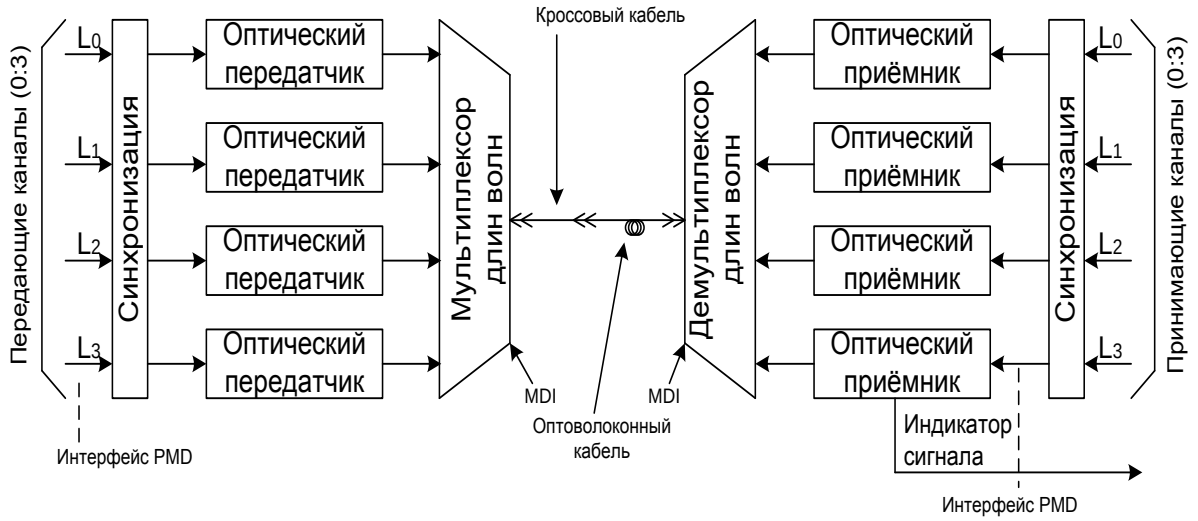


Рис. 3.18. Интерфейс 10GBase-LX4 использует технику WDM

Максимальное расстояние между передатчиком и приемником стандарта 10GBase-LX4 на многомодовом волокне равно 200–300 м (в зависимости от полосы пропускания волокна), а на одномодовом — 10 км.

В каждой из групп 10GBase-W и 10GBase-R может быть три варианта подуровня PMD: S, L и E в зависимости от используемого для передачи информации диапазона волн — 850, 1310 или 1550 нм соответственно. Таким образом, существуют интерфейсы 10GBase-WS, 10GBase-WL, 10GBase-WE и 10GBase-RS, lOGBase-RL и lOGBase-RE. Каждый из них передает информацию с помощью одной волны соответствующего диапазона.

В отличие от 10GBase-R физические интерфейсы группы 10GBase-W обеспечивают скорость передачи и формат данных, совместимые с интерфейсом SONET STS-192/SDH STM-64. Пропускная способность интерфейсов группы W равна 9,95328 Гбит/с, а эффективная скорость передачи данных — 9,58464 Гбит/с (часть пропускной способности тратится на заголовки кадров STS/STM). Из-за того, что скорость передачи информации у этой группы интерфейсов ниже, чем 10 Гбит/с, они могут взаимодейство-

вать только между собой, т. е. соединение, например, интерфейсов 10GBase-RL и 10Base-WL невозможно.

Интерфейсы группы W не являются полностью совместимыми по электрическим характеристикам с интерфейсами SONET STS-192/SDH STM-64. Поэтому для соединения сетей 10G Ethernet через первичную сеть SONET/SDH у мультиплексоров первичной сети должны быть специальные интерфейсы 10G, совместимые со спецификациями 10GBase-W. Поддержка оборудованием 10GBase-W скорости 9,95328 Гбит/с обеспечивает принципиальную возможность передачи трафика 10G Ethernet через сети SONET/SDH в кадрах STS-192/STM-64.

Физические интерфейсы, работающие в окне прозрачности E, обеспечивают передачу данных на расстояния до 40 км. Это позволяет строить не только локальные сети, но и сети мегаполисов, что нашло отражение в поправках к исходному тексту стандарта 802.3.

3.8. ОБОРУДОВАНИЕ ДЛЯ ЛОКАЛЬНЫХ СЕТЕЙ

Сетевой адаптер, или сетевая интерфейсная карта (Network Interface Card, NIC), вместе со своим драйвером реализует канальный уровень модели OSI в конечном узле сети — компьютере. Точнее, в сетевой операционной системе пара «адаптер — драйвер» выполняет только функции физического уровня и уровня MAC, в то время как уровень LLC обычно реализуется модулем операционной системы, единым для всех драйверов и сетевых адаптеров. Например, в ОС Windows XP уровень LLC реализуется в модуле NDIS, общем для всех драйверов сетевых адаптеров независимо от того, какую технологию поддерживает драйвер. Сетевой адаптер совместно с драйвером выполняют две операции: передачу и прием кадра.

Передача кадра из компьютера в кабель требует выполнения перечисленных ниже этапов.

- Прием кадра данных уровня LLC через межуровневый интерфейс вместе с адресной информацией уровня MAC. Обычно взаимодействие между протоколами внутри компьютера происходит через буферы, расположенные в оперативной памяти. Данные для передачи в сеть помещаются в эти буферы протоколами верхних уровней, которые извлекают их из дисковой памяти либо из файлового кэша с помощью подсистемы ввода-вывода операционной системы.

- Оформление кадра данных уровня MAC, в который инкапсулируется кадр уровня LLC. Заполнение адресов приемника и источника, вычисление контрольной суммы.

- Формирование символов кодов при использовании избыточных кодов типа 4B/5B. Скремблирование кодов для получения более равномерного спектра сигналов. Этот этап выполняется не во всех протоколах, например, технология Ethernet 10 Мбит/с обходится без него.

- Выдача сигналов в кабель в соответствии с принятым линейным кодом: манчестерским, NRZI, MLT-3 и т. п.

- Прием кадра из кабеля в компьютер включает следующие действия.

- Прием из кабеля сигналов, кодирующих битовый поток.

- Выделение сигналов на фоне шума. Эту операцию могут выполнять различные специализированные микросхемы или процессоры DSP. В результате в приемнике адаптера образуется некоторая битовая последовательность, с большой степенью вероятности совпадающая с той, которая была послана передатчиком. Если данные перед отправкой в кабель подвергались скремблированию, то они пропускаются через дескремблер, после чего в адаптере восстанавливаются символы кода, посланные передатчиком.

- Проверка контрольной суммы кадра. Если контрольная сумма неверна, то кадр отбрасывается, а через межуровневый интерфейс наверх, протоколу LLC передается соответствующий код ошибки. Если контрольная сумма верна, то из MAC-кадра извлекается LLC-кадр и передается через межуровневый интерфейс наверх, протоколу LLC.

Распределение обязанностей между сетевым адаптером и его драйвером стандартами не определяется, поэтому каждый производитель решает этот вопрос самостоятельно. Обычно сетевые адаптеры делятся на адаптеры для клиентских компьютеров и адаптеры для серверов.

Рассмотрим более подробно особенности функционирования некоторых наиболее важных сетевых устройств

Практически во всех современных технологиях локальных сетей определено устройство, которое имеет несколько равноправных названий: *концентратор, хаб, повторитель*. В зависимости от области применения этого устройства в значительной степени изменяется состав его функций и конструктивное исполнение. Неизменной остается только основная функ-

ция — повторение кадра либо на всех портах (как определено в стандарте Ethernet), либо только на некоторых портах, в соответствии с конкретным алгоритмом, определенным тем или иным стандартом.

Концентратор обычно имеет несколько портов, к которым с помощью отдельных физических сегментов кабеля подключаются конечные узлы сети — компьютеры. Концентратор объединяет отдельные физические сегменты сети в единую разделяемую среду, доступ к которой осуществляется в соответствии с одним из рассмотренных протоколов локальных сетей — Ethernet, Token Ring и т. п. Так как логика доступа к разделяемой среде существенно зависит от технологии, то для каждого типа технологии выпускаются свои концентраторы: Ethernet, Token Ring, FDDI.

Каждый концентратор выполняет некоторую основную функцию, определенную в соответствующем стандарте той технологии, которую он поддерживает. Помимо основной функции концентратор может выполнять некоторое количество дополнительных функций, которые либо вообще не определены в стандарте, либо являются опциональными. Например, концентратор Token Ring может выполнять функцию отключения некорректно работающих портов и перехода на резервное кольцо, хотя в стандарте такие его возможности не описаны. Концентратор оказался удобным устройством для выполнения дополнительных функций, облегчающих контроль и эксплуатацию сети.

У концентратора есть многочисленные функции, включая автосегментацию — способность отключать некорректно работающие порты, изолируя тем самым остальную часть сети от возникших в узле проблем. Например, основной причиной отключения порта в стандартах Ethernet и Fast Ethernet является отсутствие ответа на последовательность импульсов теста связности, посылаемых во все порты каждые 16 мс. В этом случае неисправный порт отключается, но импульсы теста связности будут продолжать посылаться в порт с тем, чтобы при восстановлении устройства работа с ним была продолжена автоматически.

Рассмотрим ситуации, в которых концентраторы Ethernet и Fast Ethernet выполняют отключение порта.

Ошибки на уровне кадра. Если интенсивность прохождения через порт кадров, имеющих ошибки, превышает заданный порог, то порт отключается, а затем, при отсутствии ошибок в течение заданного времени, включа-

ется снова. Такими ошибками могут быть: неверная контрольная сумма, неверная длина кадра (больше 1518 байт или меньше 64 байт), неоформленный заголовок кадра.

Множественные коллизии. Если концентратор фиксирует, что источником коллизии был один и тот же порт 60 раз подряд, то порт отключается. Через некоторое время порт снова будет включен.

Затянувшаяся передача. Как и сетевой адаптер, концентратор контролирует время прохождения одного кадра через порт. Если это время превышает время передачи кадра максимальной длины в 3 раза, то порт отключается.

Поддержка резервных связей. Так как использование резервных связей в концентраторах определено только в стандарте FDDI, для остальных стандартов разработчики концентраторов поддерживают такую функцию с помощью своих частных решений. Например, концентраторы Ethernet могут образовывать только иерархические связи без петель. Поэтому резервные связи всегда должны соединять отключенные порты, чтобы не нарушать логику работы сети.

Защита от несанкционированного доступа. Разделяемая среда предоставляет очень удобную возможность для несанкционированного прослушивания сети и получения доступа к передаваемым данным. Разработчики концентраторов предоставляют определенные средства защиты данных в разделяемых средах. Наиболее простое средство — назначение разрешенных MAC-адресов портам концентратора.

Обратимся к другому, не менее важному сетевому элементу беспроводных локальных сетей — *коммутатору*. По сути, коммутатор — это мультипроцессорный мост, способный параллельно продвигать кадры сразу между всеми парами своих портов. Однако производительность коммутаторов на несколько порядков выше, чем мостов — коммутаторы могут передавать до нескольких миллионов кадров в секунду, а мосты обычно 3–5 тысяч кадров в секунду. Структурная схема коммутатора EtherSwitch представлена на рис. 3.19.

Каждый из 8 портов 10Base-T обслуживается одним процессором пакетов Ethernet (Ethernet Packet Processor, EPP). Кроме того, коммутатор имеет системный модуль, который координирует работу всех процессоров EPP, в частности ведет общую адресную таблицу коммутатора. Для пере-

дачи кадров между портами используется коммутационная матрица. Она функционирует по принципу коммутации каналов, соединяя порты коммутатора. Для 8 портов матрица может одновременно обеспечить 8 внутренних каналов при полудуплексном режиме работы портов и 16 — при дуплексном, когда передатчик и приемник каждого порта работают независимо друг от друга.

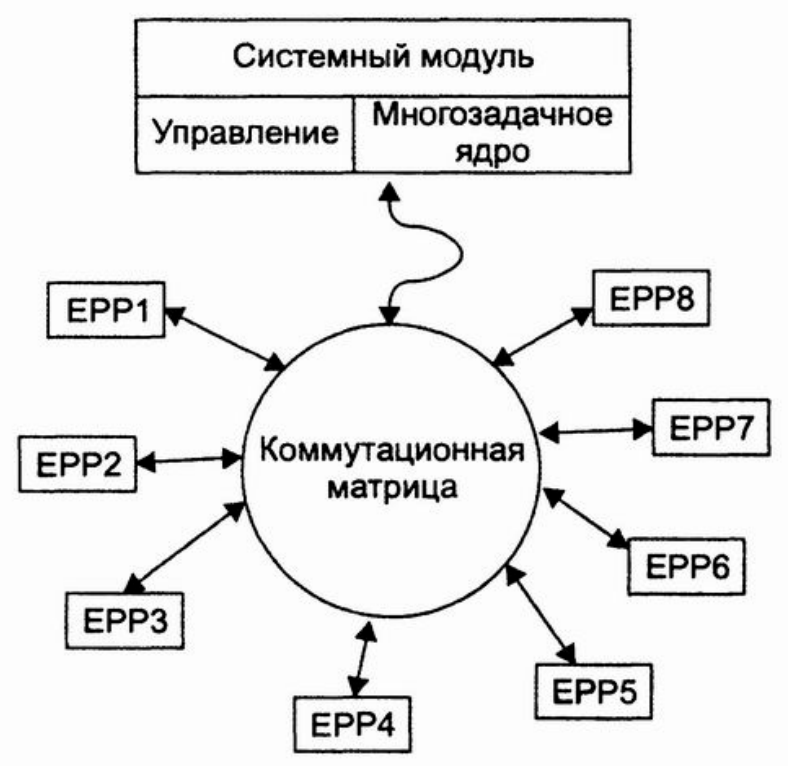


Рис. 3.19. Структура коммутатора EtherSwitch

При поступлении кадра в какой-либо порт соответствующий процессор EPP буферизует несколько первых байтов кадра, чтобы прочитать адрес назначения. После получения адреса назначения процессор сразу же приступает к обработке кадра, не дожидаясь прихода остальных его байтов.

Процессор EPP просматривает свой кэш адресной таблицы, и если он не находит там нужного адреса, то обращается к системному модулю, который работает в многозадачном режиме, параллельно обслуживая запросы всех процессоров EPP. Системный модуль производит просмотр общей адресной таблицы и возвращает процессору найденную строку, которую тот буферизует в своем кэше для последующего использования.

Если адрес назначения найден в адресной таблице, и кадр нужно отфильтровать, процессор просто прекращает записывать в буфер байты кадра, очищает буфер и ждет поступления нового кадра. Если же адрес найден, но кадр нужно передать на другой порт, то процессор, продолжая прием кадра в буфер, обращается к коммутационной матрице, пытаясь установить в ней путь, связывающий его порт с портом, через который идет маршрут к адресу назначения.

Коммутационная матрица может это сделать только в том случае, когда порт адреса назначения в этот момент свободен, т. е. не соединен с другим портом данного коммутатора. Если же порт занят, то, как и в любом устройстве с коммутацией каналов, матрица в соединении отказывает. В этом случае кадр полностью буферизуется процессором входного порта, после чего процессор ожидает освобождения выходного порта и образования коммутационной матрицей нужного пути.

После того как нужный путь установлен, в него направляются буферизованные байты кадра, которые принимаются процессором выходного порта. Как только процессор выходного порта получает доступ к подключенному к нему сегменту Ethernet по алгоритму CSMA/CD, байты кадра сразу же начинают передаваться в сеть. Процессор входного порта постоянно хранит несколько байтов принимаемого кадра в своем буфере, что позволяет ему независимо и асинхронно принимать и передавать байты кадра.

Описанный выше способ передачи кадра без его полной буферизации получил название коммутации «на лету» (on-the-fly), или «напролет» (cut-through). Этот способ представляет собой, по сути, конвейерную обработку кадра, когда частично совмещаются во времени несколько этапов его передачи.

Когда говорят, что коммутатор может поддерживать устойчивый неблокирующий режим работы коммутатора, то имеют в виду, что коммутатор передает кадры со скоростью их поступления в течение произвольного промежутка времени. Для обеспечения подобного режима нужно таким образом распределить потоки кадров по выходным портам, чтобы, во-первых, порты справлялись с нагрузкой, а во-вторых, коммутатор мог всегда в среднем передать на выходы столько кадров, сколько их поступило на входы. Если же входной поток кадров (просуммированный по всем портам) в среднем будет превышать выходной поток кадров (также просум-

мированный по всем портам), то кадры будут накапливаться в буферной памяти коммутатора, и при переполнении — просто отбрасываться.

Иногда говорят, что коммутатор поддерживает мгновенный неблокирующий режим. Это означает, что он может принимать и обрабатывать кадры от всех своих портов на максимальной скорости протокола, независимо от того, обеспечиваются ли условия устойчивого равновесия между входным и выходным трафиком. Правда, обработка некоторых кадров при этом может быть неполной — при занятости выходного порта кадр помещается в буфер коммутатора.

Одной из главных проблем, решаемых коммутатором, является борьба с перегрузками.

Борьба с перегрузками методом *обратного давления* (backpressure) состоит в создании искусственных коллизий в сегменте, который чересчур интенсивно посылает кадры в коммутатор. Для этого коммутатор обычно использует jam-последовательность, отправляемую на выход порта, к которому подключен сегмент (или узел), чтобы приостановить его активность.

Другой метод — *метод торможения* — обычно применяется в том случае, когда соседом является конечный узел. Метод основан на так называемом агрессивном захвате среды либо после окончания передачи очередного кадра, либо после коллизии. Эти два случая иллюстрируются на рис. 3.18.

В первом случае (рис. 3.20, *а*) коммутатор окончил передачу очередного кадра и вместо технологической паузы в 9,6 мкс сделал паузу в 9,1 мкс, после чего начал передачу нового кадра. Компьютер не смог захватить среду, так как он выдержал стандартную паузу в 9,6 мкс и обнаружил после этого, что среда уже занята.

Во втором случае (рис. 3.20, *б*) кадры коммутатора и компьютера столкнулись, т. е. была зафиксирована коллизия. Так как компьютер сделал паузу после коллизии в 51,2 мкс, как это положено по стандарту (интервал отсрочки равен 512 битовых интервалов), а коммутатор — 50 мкс, в этом случае компьютеру не удалось передать свой кадр.

Коммутаторы в соответствии со спецификациями IEEE 802.1H и RFC 1042 могут выполнять трансляцию одного протокола канального уровня в другой, например Ethernet в FDDI, Fast Ethernet в Token Ring и т. п. Трансляцию протоколов локальных сетей облегчает тот факт, что наиболее

сложную работу, а именно, работу по трансляции адресов, которую при объединении гетерогенных сетей выполняют маршрутизаторы и шлюзы, в данном случае выполнять не нужно.

Рис. 3.20. Агрессивное поведение коммутатора при перегрузках буферов

Все конечные узлы локальных сетей имеют уникальные адреса одного и того же формата (MAC-адреса) независимо от поддерживаемого протокола. Поэтому адрес сетевого адаптера Ethernet понятен сетевому адаптеру FDDl, и они оба могут использовать эти адреса в полях своих кадров, не задумываясь о том, что узел, с которым они взаимодействуют, принадлежит сети, работающей по другой технологии. Поэтому при согласовании протоколов локальных сетей коммутаторы просто переносят адреса приемника и источника из кадра одного протокола в кадр другого.

Многие модели коммутаторов позволяют администраторам задавать дополнительные условия фильтрации кадров наряду со стандартными условиями их фильтрации в соответствии с информацией адресной таблицы. Пользовательские фильтры предназначены для создания дополнительных барьеров на пути кадров, позволяющих ограничить доступ определённых групп пользователей к отдельным службам сети.

Наиболее простыми являются пользовательские фильтры на основе MAC-адресов станций. Так как MAC-адреса — это та информация, с кото-

рой работает коммутатор, он позволяет создавать такие фильтры в удобной для администратора форме, возможно, проставляя некоторые условия в дополнительном поле адресной таблицы (например, условие отбрасывать кадры с определенным адресом). Таким способом пользователю, работающему на компьютере с данным MAC-адресом, полностью запрещается доступ к ресурсам другого сегмента сети.

Скорости фильтрации и продвижения кадров — две основные характеристики производительности коммутатора. Эти характеристики являются интегральными, они не зависят от того, каким образом технически реализован коммутатор.

Скорость фильтрации — это скорость, с которой коммутатор выполняет перечисленные ниже этапы обработки кадров:

- прием кадра в свой буфер.

- просмотр адресной таблицы с целью нахождения порта для адреса назначения кадра.

- уничтожение кадра, так как его порт назначения и порт источника принадлежат одному логическому сегменту.

Скорость фильтрации практически у всех коммутаторов является неблокирующей — коммутатор успевает отбрасывать кадры в темпе их поступления.

Скорость продвижения — это скорость, с которой коммутатор выполняет следующие этапы обработки кадров:

- прием кадра в свой буфер.

- просмотр адресной таблицы с целью нахождения порта для адреса назначения кадра.

- передача кадра в сеть через найденный по адресной таблице порт назначения.

Как скорость фильтрации, так и скорость продвижения измеряются обычно в кадрах в секунду. Если в характеристиках коммутатора не уточняется, для какого протокола и для какого размера кадра приведены значения скоростей фильтрации и продвижения, то по умолчанию считается, что эти показатели даются для протокола Ethernet и кадров минимального размера, т. е. кадров длиной 64 байт. Режим передачи кадров минимальной длины используется как наиболее сложный тест, который должен подтвер-

дить способность коммутатора работать при наихудшем сочетании параметров трафика.

Задержка передачи кадра измеряется как время, прошедшее с момента прихода первого байта кадра на входной порт коммутатора до момента появления этого байта на его выходном порту. Задержка складывается из времени, затрачиваемого на буферизацию байтов кадра, а также времени, затрачиваемого на обработку кадра коммутатором, — просмотр адресной таблицы, принятие решения о фильтрации или продвижении, получение доступа к среде выходного порта. Величина вносимой коммутатором задержки зависит от режима его работы. Если коммутация осуществляется «на лету», то задержки обычно невелики и составляют от 5 до 40 мкс, а при полной буферизации кадров — от 50 до 200 мкс (для кадров минимальной длины).

Производительность коммутатора определяется количеством пользовательских данных, переданных в единицу времени через его порты (измеряется в мегабитах в секунду). Так как коммутатор работает на канальном уровне, для него пользовательскими данными являются те данные, которые переносятся в поле данных кадров протоколов канального уровня — Ethernet, Token Ring, FDDI и т. п. Максимальное значение производительности коммутатора всегда достигается на кадрах максимальной длины, так как при этом минимальна доля накладных расходов на служебную информацию кадра. Поскольку коммутатор — это многопортовое устройство, для него в качестве характеристики принято давать максимальную суммарную производительность при одновременной передаче трафика по всем его портам.

ВОПРОСЫ И ЗАДАНИЯ К ГЛАВЕ 3

1. Поясните смысл понятий «коллизия» и «домен коллизий».
2. Поясните алгоритм работы станций в сети с множественным доступом CSMA/CD.
3. Чем определяется реальная скорость передачи данных в сетях Ethernet?
4. Какие функции выполняются уровнем LLC?
5. Какие топологические структуры сетей Ethernet Вам известны? Дайте их сравнительную характеристику.
6. Что такое время двойного оборота?
7. Из каких соображений выбирается максимальная длина физического сегмента в стандартах Ethernet?
8. Перечислите сходства и отличия технологий Fast Ethernet и Gigabit Ethernet?
9. Сравните между собой методы множественного доступа, реализованные в технологиях Ethernet и Token Ring.
10. В чем состоит сходство и различие технологий FDDI и Token Ring?
11. За счет чего в сети FDDI обеспечивается отказоустойчивость?

4. ПРОТОКОЛЫ IP-СЕТЕЙ

В предыдущей главе был проведен краткий анализ существующих на сегодняшний день основных сетевых технологий, рассмотрено назначение и характеристики базовых сетевых устройств.

Цель данной главы — более детальное изучение «интеллектуального ядра» большинства сетевых технологий: стека протоколов TCP/IP. Будут рассмотрены принципы адресации в сетях, маршрутизации и фрагментации пакетов пользовательской информации, а также конкретные сетевые протоколы, реализующие эти принципы [1, 3, 6, 8].

4.1. АДРЕСАЦИЯ В IP-СЕТЯХ. ПРОТОКОЛ IPv4

В большинстве технологий LAN (Ethernet, FDDI, Token Ring) для однозначной адресации интерфейсов используются MAC-адреса. Существует немало технологий (X.25, ATM, Frame Relay), в которых применяются другие схемы адресации. Будучи автономными, такие сети используют свою схему адресации исключительно для обеспечения связи собственных узлов. Однако как только некоторая сеть объединяется с другими сетями, функциональность этих адресов расширяется, они становятся необходимым элементом вышележащей объединяющей технологии — в данном случае технологии TCP/IP. Роль, которую играют эти адреса в TCP/IP, не зависит от того, какая именно технология используется в подсети, поэтому они имеют общее название — локальные (аппаратные) адреса.

Слово «локальный» в контексте TCP/IP означает «действующий не во всей составной сети, а лишь в пределах подсети». Именно в таком смысле понимаются здесь термины: «локальная технология» (технология, на основе которой построена подсеть), «локальный адрес» (адрес, который используется некоторой локальной технологией для адресации узлов в пределах подсети). Напомним, что в качестве подсети («локальной сети») может выступать сеть, построенная как на основе локальной технологии, например Ethernet или FDDI, так и на основе глобальной технологии, например X.25 или Frame Relay. Следовательно, использование термина «локальный» указывает не характеристику технологии, на которой построена эта подсеть, а как роль, которую играет эта подсеть в архитектуре составной сети.

Сложности могут возникнуть и при интерпретации определения «аппаратный». В данном случае этот термин подчеркивает концептуальное представление разработчиков стека TCP/IP о подсети как о некотором вспомогательном аппаратном средстве, единственной функцией которого является перемещение IP-пакета через подсеть до ближайшего шлюза (маршрутизатора). При этом не важно, что реально локальная нижележащая технология может быть достаточно сложной, ибо все ее сложности игнорируются технологией TCP/IP.

Чтобы использование технологи TCP/IP могло решить задачу объединения сетей, необходима глобальная система адресации, не зависящая от способов адресации узлов в отдельных сетях. Такая система должна позволять универсальным и однозначным способом идентифицировать любой интерфейс составной сети. Очевидным решением является уникальная нумерация всех сетей (подсетей) составной сети, а затем — нумерация всех узлов в пределах каждой из подсетей. Пара, состоящая из номера сети и номера узла, отвечает поставленным условиям и может служить в качестве сетевого адреса.

В качестве номера узла может выступать либо локальный адрес этого узла (такая схема принята в стеке IPX/SPX), либо некоторое число, никак не связанное с локальной технологией и однозначно идентифицирующее узел в пределах данной подсети. В первом случае сетевой адрес становится зависимым от локальных технологий, что ограничивает его применение. Например, сетевые адреса IPX/ SPX рассчитаны на работу в составных сетях, объединяющих сети, в которых используются только MAC-адреса или адреса аналогичного формата. Второй подход более универсален, он характерен для стека TCP/IP.

Каждый раз, когда пакет направляется адресату через составную сеть, в его заголовке указывается IP-адрес узла назначения. По номеру сети назначения каждый очередной маршрутизатор находит IP-адрес следующего маршрутизатора. Перед тем как отправить пакет в следующую сеть, маршрутизатор должен определить на основании найденного IP-адреса следующего маршрутизатора его локальный адрес. Для этой цели протокол IP, как показано на рис. 4.1, обращается к протоколу разрешения адресов (см. далее).

Для идентификации компьютеров аппаратное и программное обеспечение в сетях TCP/IP полагается на IP-адреса. Например, команда ftp://192.45.66.17 будет устанавливать сеанс связи с нужным ftp-сервером, а команда http://203.23.106.33 откроет начальную страницу на корпоративном веб-сервере. Однако пользователи обычно предпочитают работать с более удобными символьными именами компьютеров.

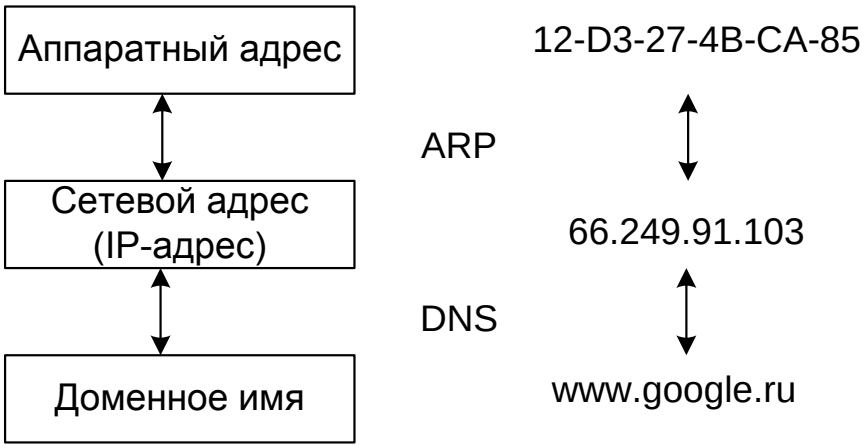


Рис. 4.1. Преобразование адресов

Символьные идентификаторы сетевых интерфейсов в пределах составной сети строятся по иерархическому признаку. Составляющие полного символьного (или доменного) имени в IP-сетях разделяются точкой и перечисляются в следующем порядке: сначала простое имя хоста, затем имя группы хостов (например, имя организации), затем имя более крупной группы (домена) и так до имени домена самого высокого уровня (например, домена, объединяющего организации по географическому принципу: RU — Россия, UK — Великобритания, US — США). Примером доменного имени может служить имя cee.spbstu.ru.

Между доменным именем и IP-адресом узла нет никакой функциональной зависимости, поэтому единственный способ установления соответствия — это таблица. В сетях TCP/IP используется специальная система доменных имен (Domain Name System, DNS), которая устанавливает это соответствие на основании создаваемых администраторами сети таблиц соответствия. Поэтому доменные имена называют также DNS-именами.

В общем случае сетевой интерфейс может иметь несколько локальных адресов, сетевых адресов, доменных имен. Рассмотрим формат сетевого адреса.

В заголовке IP-пакета для хранения IP-адресов отправителя и получателя отводятся два поля, каждое имеет фиксированную длину 4 байт (32 бит). IP-адрес состоит из двух логических частей: номера сети и номера узла в сети.

Наиболее распространенной формой представления IP-адреса является запись в виде четырех чисел, представляющих значения каждого байта в десятичной форме и разделенных точками, например:

128.10.2.30

Этот же адрес может быть представлен в двоичном формате:

10000000 00001010 00000010 00011110

или в шестнадцатеричном:

80.0A.02.1D

Заметим, что запись адреса не предусматривает специального разграничительного знака между номером сети и номером узла. Вместе с тем, при передаче пакета по сети часто возникает необходимость разделить адрес на эти две части. Например, маршрутизация, как правило, осуществляется на основании номера сети, поэтому каждый маршрутизатор, получая пакет, должен прочитать из соответствующего поля заголовка адрес назначения и выделить из него номер сети. Возникает вопрос, каким образом маршрутизаторы определяют, какая часть из 32 бит, отведенных под IP-адрес, относится к номеру сети, а какая — к номеру узла?

Можно предложить несколько вариантов решения этой проблемы. Простейший из них состоит в использовании фиксированной границы. При этом все 32-битовое поле адреса заранее делится на две части не обязательно равной, но фиксированной длины, в одной из которых всегда будет размещаться номер сети, а в другой — номер узла.

Однако такое простое решение имеет, по меньшей мере, один существенный изъян. Поскольку поле, которое отводится для хранения номера узла, имеет фиксированную длину, все сети будут иметь одинаковое максимальное число узлов. Так, если, например, под номер сети отвести один первый байт, то все адресное пространство распадется на сравнительно небольшое (28) число сетей огромного размера (2 узлов). Если границу пере-

двинуть дальше вправо, то сетей станет больше, но все равно они будут одинакового размера. Очевидно, что такой жесткий подход не позволяет дифференцированно удовлетворять потребности отдельных предприятий и организаций. Именно поэтому он не нашел применения, хотя и использовался на начальном этапе существования технологии TCP/IP (RFC 760).

Второй подход (RFC 950, RFC 1518) основан на использовании *маски*, которая позволяет максимально гибко устанавливать границу между номером сети и номером узла. При таком подходе адресное пространство можно использовать для создания множества сетей разного размера. Маска — это число, применяемое в паре с IP-адресом, причем двоичная запись маски содержит непрерывную последовательность единиц в тех разрядах, которые должны в IP-адресе интерпретироваться как номер сети. Граница между последовательностями единиц и нулей в маске соответствует границе между номером сети и номером узла в IP-адресе.

Третий и до недавнего времени наиболее распространенный способ решения данной проблемы заключается в использовании *классов адресов* (RFC 791). Этот способ представляет собой компромисс по отношению к двум предыдущим: размеры сетей хотя и не могут быть произвольными, как при использовании масок, но и не должны быть одинаковыми, как при установлении фиксированных границ. При этом вводится пять классов адресов: *A*, *B*, *C*, *D* и *E*. Три из них — *A*, *B* и *C* — используются для адресации сетей, а два — *D* и *E* — имеют специальное назначение. Для каждого класса сетевых адресов определено собственное положение границы между номером сети и номером узла.

Признаком, на основании которого IP-адрес относится к тому или иному классу, являются значения нескольких первых битов адреса. В табл. 4.1 иллюстрируется структура IP-адресов разных классов.

К классу *A* относится адрес, в котором старший бит имеет значение 0. Во адресах класса *A* под идентификатор сети отводится 1 байт, а остальные 3 байта интерпретируются как номер узла в сети. Те сети, в которых все IP-адреса имеют значение первого байта в диапазоне от 1 (00000001) до 126 (01111110), называются сетями класса *A*. Значение 0 (00000000) первого байта не используется, а значение 127 (01111111) зарезервировано для специальных целей. Сетей класса *A* сравнительно немного, зато количество узлов в них может достигать 2^{24} = 16 777 216 узлов.

Таблица 4.1

Структура IP-адресов разных классов

Класс	Первые биты	Наименьший номер сети	Наибольший номер сети	Максимальное число узлов
A	0	1.0.0.0 (0 — не используется)	126.0.0.0 (127 — зарезервирован)	2^{24}, поле 3 байта
B	10	128.0.0.0	191.255.0.0	2^{16}, поле 2 байта
C	110	192.0.0.0	223.255.255.0	2^{8}, поле 1 байт
D	1110	224.0.0.0	239.255.255.255	Групповые адреса
E	11110	240.0.0.0	247.255.255.255	Зарезервировано

К классу *A* относится адрес, в котором старший бит имеет значение 0. В адресах класса *A* под идентификатор сети отводится 1 байт, а остальные 3 байта интерпретируются как номер узла в сети. Те сети, в которых все IP-адреса имеют значение первого байта в диапазоне от 1 (00000001) до 126 (01111110), называются сетями класса *A*. Значение 0 (00000000) первого байта не используется, а значение 127 (01111111) зарезервировано для специальных целей. Сетей класса *A* сравнительно немного, зато количество узлов в них может достигать 2^{24} = 16 777 216 узлов.

К классу *B* относятся все адреса, старшие два бита которых имеют значение 10. В адресах класса *B* под номер сети и под номер узла отводится по два байта. Сети, значения первых двух байтов адресов которых находятся в диапазоне от 128.0. (10000000 00000000) до 191.255 (10111111 11111111), называются сетями класса *B*. Ясно, что сетей класса *B* больше, чем сетей класса *A*, а размеры их меньше. Максимальное количество узлов в сетях класса *B* составляет 2^{16} = 65 536.

К классу *C* относятся все адреса, старшие три бита которых имеют значение 110. В адресах класса *C* под номер сети отводится 3 байта, а под номер узла — 1 байт. Сети, старшие три байта которых находятся в диапа-

зоне от 192.0.0 (11000000 00000000 00000000) до 223.255 (11011111 11111111 11111111), называются сетями класса *C*. Сети класса C наиболее распространены и имеют наименьшее максимальное число узлов $2^8 = 256$.

Если адрес начинается с последовательности 1110, то он является адресом класса *D* и обозначает особый, групповой адрес (multicast address). В то время как адреса классов *A*, *B* и *C* используются для идентификации отдельных сетевых интерфейсов, т. е. являются индивидуальными адресами (unicast address), групповой адрес идентифицирует группу сетевых интерфейсов, которые в общем случае могут принадлежать разным сетям. Интерфейс, входящий в группу, получает наряду с обычным индивидуальным IP-адресом еще один групповой адрес. Если при отправке пакета в качестве адреса назначения указан адрес класса *D*, то такой пакет должен быть доставлен всем узлам, которые входят в группу.

Если адрес начинается с последовательности 11110, то это значит, что данный адрес относится к классу *E*. Адреса этого класса зарезервированы для будущих применений.

Чтобы получить из IP-адреса номер сети и номер узла, требуется не только разделить адрес на две соответствующие части, но и дополнить каждую из них нулями до полных 4 байт. Возьмем, например, адрес класса *B* 129.64.134.5. Первые два байта идентифицируют сеть, а последующие два — узел. Таким образом, номером сети является адрес 129.64.0.0, а номером узла — адрес 0.0.134.5.

В стеке TCP/IP существуют ограничения при назначении IP-адресов, а именно, номера сетей и номера узлов не могут состоять из одних двоичных нулей или единиц. Отсюда следует, что максимальное количество узлов, приведенное в табл. 4.1 для сетей каждого класса, должно быть уменьшено на 2. Например, в адресах класса *C* под номер узла отводится 8 бит, которые позволяют задавать 256 номеров: от 0 до 255. Однако в действительности максимальное число узлов в сети класса *C* не может превышать 254, так как адреса 0 и 255 запрещены для адресации сетевых интерфейсов. Из этих же соображений следует, что конечный узел не может иметь адрес типа 98.255.255.255, поскольку номер узла в этом адресе класса *A* состоит из одних двоичных единиц.

Из этого можно сделать вывод о том, что некоторые IP-адреса интерпретируются особым образом.

Если IP-адрес состоит только из двоичных нулей, то он называется неопределенным адресом и обозначает адрес того узла, который сгенерировал этот пакет. Адрес такого вида в особых случаях помещается в заголовок IP-пакета в поле адреса отправителя.

Если в поле номера сети стоят только нули, то по умолчанию считается, что узел назначения принадлежит той же самой сети, что и узел, который отправил пакет. Такой адрес также может быть использован только в качестве адреса отправителя.

Если все двоичные разряды IP-адреса равны 1, то пакет с таким адресом назначения должен рассылаться всем узлам, находящимся в той же сети, что и источник этого пакета. Такой адрес называется ограниченным широковещательным (limited broadcast). Ограниченность в данном случае означает, что пакет не выйдет за границы данной сети ни при каких условиях.

Если в поле адреса назначения в разрядах, соответствующих номеру узла, стоят только единицы, то пакет, имеющий такой адрес, рассылается всем узлам сети, номер которой указан в адресе назначения. Например, пакет с адресом 192.190.21.255 будет направлен всем узлам сети 192.190.21.0. Такой тип адреса называется широковещательным (broadcast).

Особый смысл имеет IP-адрес, первый октет которого равен 127. Этот адрес является внутренним адресом стека протоколов компьютера (или маршрутизатора). Он используется для тестирования программ, а также для организации работы клиентской и серверной частей приложения, установленных на одном компьютере. Обе программные части данного приложения спроектированы в расчете на то, что они будут обмениваться сообщениями по сети. Какой же IP-адрес они должны использовать для этого? Адрес сетевого интерфейса компьютера, на котором они установлены? Но это приводит к избыточным передачам пакетов в сеть. Экономичным решением является применение внутреннего адреса 127.0.0.0. В IP-сети запрещается присваивать сетевым интерфейсам IP-адреса, начинающиеся со 127. Когда программа посылает данные по IP-адресу 127.x.x.x, данные не передаются в сеть, а возвращаются модулям верхнего уровня того же компьютера как только что принятые. Маршрут перемещения данных образует

«петлю», поэтому этот адрес называется адресом обратной петли (loopback).

Групповые адреса (multicast), относящиеся к классу *D*, предназначены для экономичного распространения в интернете или большой корпоративной сети аудио- или видеопрограмм, адресованных сразу большой аудитории слушателей или зрителей. Если групповой адрес помещен в поле адреса назначения IP-пакета, то данный пакет должен быть доставлен сразу нескольким узлам, которые образуют группу с номером, указанным в поле адреса. Один и тот же узел может входить в несколько групп. В общем случае члены группы могут распределяться по различным сетям, находящимся друг от друга на произвольно большом расстоянии.

Групповой адрес не делится на номера сети и узла и обрабатывается маршрутизатором особым образом. Основное назначение групповых адресов — распространение информации по схеме «один ко многим». От того, найдут ли групповые адреса широкое применение (сейчас их используют в основном небольшие экспериментальные «островки» в Интернете), зависит, сможет ли Интернет создать серьезную конкуренцию радио и телевидению.

Снабжая каждый IP-адрес маской, можно отказаться от понятий классов адресов и сделать более гибкой систему адресации. Пусть, например, для IP-адреса 129.64.134.5 указана маска 255.255.128.0, т. е. в двоичном виде IP-адрес 129.64.134.5 — это

10000001.01000000.10000110.00000101,

а маска 255.255.128.0 — это

11111111.11111111.10000000.00000000.

Если игнорировать маску и интерпретировать адрес 129.64.134.5 на основе классов, то номером сети является 129.64.0.0, а номером узла — 0.0.134.5 (поскольку адрес относится к классу *B*). Если же использовать маску, то 17 последовательных двоичных единиц в маске 255.255.128.0, «наложенные» на IP-адрес 129.64.134.5, делят его на две части: номер сети

10000001.01000000.1

и номер узла

0000110.00000101.

В десятичной форме записи номера сети и узла, дополненные нулями до 32 бит, выглядят, соответственно, как 129.64.128.0 и 0.0.6.5.

Наложение маски можно интерпретировать как выполнение логической операции И (AND). Так, в предыдущем примере номер сети из адреса 129.64.134.5 является результатом выполнения логической операции AND с маской 255.255.128.0:

10000001 01000000 10000110 00000101

AND

11111111.11111111.10000000.00000000

Для стандартных классов сетей маски имеют следующие значения:

- класс *A*: 11111111. 00000000. 00000000. 00000000 (255.0.0.0);
- класс *B*: 11111111. 11111111. 00000000. 00000000 (255.255.0.0);
- класс *C*: 11111111. 11111111. 11111111. 00000000 (255.255.255.0).

Механизм масок широко распространен в IP-маршрутизации, причем маски могут использоваться для самых разных целей. С их помощью администратор может разбивать одну, выделенную ему поставщиком услуг сеть определенного класса, на несколько других, не требуя дополнительных номеров сетей — эта операция называется разделением на подсети (subnetting). На основе этого же механизма поставщики услуг могут объединять адресные пространства нескольких сетей путем введения так называемых «префиксов» для уменьшения объема таблиц маршрутизации и повышения за счет этого производительности маршрутизаторов — такая операция называется объединением подсетей (supernetting).

Обратимся непосредственно к рассмотрению формата IP-пакета (рис. 4.2). Имеется прямая связь между количеством полей заголовка пакета и функциональной сложностью протокола, который работает с этим заголовком. Чем проще заголовок — тем проще соответствующий протокол. Большая часть действий протокола связана с обработкой той служебной информации, которая переносится в полях заголовка пакета.

Поле номера версии занимает 4 бита и идентифицирует версию протокола IP. Сейчас повсеместно используется версия 4 (IPv4), хотя все чаще встречается и новая версия (IPv6).

Значение длины заголовка IP-пакета также занимает 4 бита и измеряется в 32-битовых словах. Обычно заголовок имеет длину в 20 байт (пять 32-битовых слов), но при добавлении некоторой служебной информации это значение может быть увеличено за счет дополнительных байтов в поле параметров. Наибольшая длина заголовка составляет 60 байт.

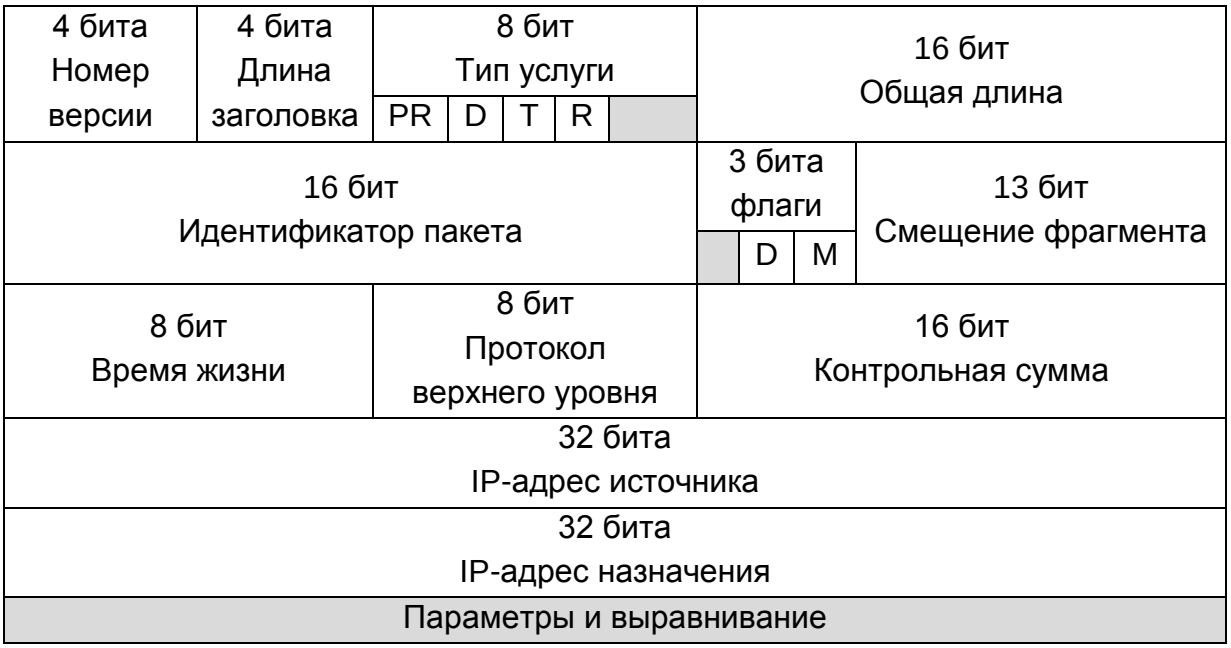


Рис. 4.2. Структура заголовка IP-пакета

Поле типа услуги (Type of Service, ToS) имеет и другое, более современное название — байт дифференцированного обслуживания, или DS-байт. Этим двум названиям соответствуют два варианта интерпретации данного поля. В обоих случаях данное поле служит одной цели — хранению признаков, которые отражают требования к качеству обслуживания пакета. В прежнем варианте первые три бита содержат значение приоритета пакета: от самого низкого 0 до самого высокого 7. Маршрутизаторы и компьютеры могут принимать во внимание приоритет пакета и обрабатывать более важные пакеты в первую очередь. Следующие три бита поля ToS определяют критерий выбора маршрута. Если бит D (Delay — задержка) установлен в 1, то маршрут должен выбираться для минимизации задержки доставки данного пакета, установленный бит T (Throughput — пропускная способность) — для максимизации пропускной способности, а бит R (Reliability — надежность) — для максимизации надежности доставки. Оставшиеся два бита имеют нулевое значение.

Стандарты дифференцированного обслуживания, принятые в конце 1990-х гг., дали новое название этому полю и переопределили назначение его битов. В DS-байте также используются только старшие 6 бит, а два младших бита остаются в качестве резерва.

Поле общей длины занимает 2 байта и характеризует общую длину пакета с учетом заголовка и поля данных. Максимальная длина пакета ограничена разрядностью поля, определяющего эту величину, и составляет 65 535 байт, однако в большинстве компьютеров и сетей столь большие пакеты не используются. При передаче по сетям различного типа длина пакета выбирается с учетом максимальной длины пакета протокола нижнего уровня, несущего IP-пакеты. Если это кадры Ethernet, то выбираются пакеты с максимальной длиной 1500 байт, умещающиеся в поле данных кадра Ethernet. В стандартах TCP/IP предусматривается, что все хосты должны быть готовы принимать пакеты вплоть до 576 байт длиной (независимо от того, приходят ли они целиком или фрагментами).

Идентификатор пакета занимает 2 байта и используется для распознавания пакетов, образовавшихся путем деления на части (фрагментации) исходного пакета. Все части (фрагменты) одного пакета должны иметь одинаковое значение этого поля.

Флаги занимают 3 бита и содержат признаки, связанные с фрагментацией. Установленный в 1 бит DF (Do not Fragment — не фрагментировать) запрещает маршрутизатору фрагментировать данный пакет, а установленный в 1 бит MF (More Fragments — больше фрагментов) говорит о том, что данный пакет является промежуточным (не последним) фрагментом. Оставшийся бит зарезервирован.

Поле смещения фрагмента занимает 13 бит и задает смещение в байтах поля данных этого фрагмента относительно начала поля данных исходного нефрагментированного пакета. Используется при сборке/разборке фрагментов пакетов. Смещение должно быть кратно 8 байт.

Поле времени жизни (Time To Live, TTL) занимает один байт и используется для задания предельного срока, в течение которого пакет может перемещаться по сети. Время жизни пакета измеряется в секундах и задается источником. По истечении каждой секунды пребывания на каждом из маршрутизаторов, через которые проходит пакет во время своего «путешествия» по сети, из его текущего времени жизни вычитается единица; единица вычитается и в том случае, если время пребывания было меньше секунды. Поскольку современные маршрутизаторы редко обрабатывают пакет дольше, чем за одну секунду, время жизни можно интерпретировать как максимальное число транзитных узлов, которые разрешено пройти па-

кету. Если значение поля времени жизни становится нулевым до того, как пакет достигает получателя, пакет уничтожается. Таким образом, время жизни является своего рода часовым механизмом уничтожения пакета.

Поле протокола верхнего уровня занимает один байт и содержит идентификатор, указывающий, какому протоколу верхнего уровня принадлежит информация, размещенная в поле данных пакета. (Значения идентификаторов для разных протоколов приводятся в документе RFC 1700, доступном по адресу http://www.Iana.org.) Например, 6 означает, что в пакете находится сообщение TCP, 17 — сообщение UDP, 1 — сообщение ICMP.

Контрольная сумма заголовка занимает 2 байта (16 бит) и рассчитывается только по заголовку. Поскольку некоторые поля заголовка меняют свое значение в процессе передачи пакета по сети (например, поле времени жизни), контрольная сумма проверяется и повторно рассчитывается на каждом маршрутизаторе и конечном узле как дополнение к сумме всех 16-битовых слов заголовка. При вычислении контрольной суммы значение самого поля контрольной суммы устанавливается в нуль. Если контрольная сумма неверна, то пакет отбрасывается, как только обнаруживается ошибка.

Поля IP-адресов источника и приемника имеют одинаковую длину 32 бита.

Поле параметров является необязательным и используется обычно только при отладке сети. Это поле состоит из нескольких подполей одного из восьми предопределенных типов. В этих подполях можно указывать точный маршрут, регистрировать проходимые пакетом маршрутизаторы, помещать данные системы безопасности или временные отметки. Так как число подполей в поле параметров может быть произвольным, в конце заголовка должно быть добавлено несколько нулевых байтов для выравнивания заголовка пакета по 32-битной границе.

4.2. ФРАГМЕНТАЦИЯ IP-ПАКЕТОВ

Прежде всего, отметим разницу между фрагментацией сообщений в узле-отправителе и динамической фрагментацией сообщений в транзитных узлах сети — маршрутизаторах. Практически во всех известных стеках есть протоколы, которые отвечают за деление (фрагментацию) сообщений

прикладного уровня на такие части, которые укладывались бы в кадры канального уровня. Для этого они анализируют тип технологии нижнего уровня и определяют ее максимальную единицу передачи MTU (Maximum Transmission Unit).

В стеке TCP/IP эту задачу решает протокол TCP, который разбивает поток байтов, передаваемый ему с прикладного уровня, на сегменты нужного размера (например, на 1460 байт, если на нижнем уровне данной сети работает протокол Ethernet). Поэтому протокол IP в узле-отправителе, как правило, не использует свои возможности по фрагментации пакетов, однако в маршрутизаторе, когда пакет необходимо передать из сети с большим в сеть с меньшим значением MTU, способности протокола IP выполнять фрагментацию становятся востребованными. В табл. 4.3 представлены значения MTU для различных сетевых технологий, откуда видно, что значения MTU для наиболее популярных технологий существенно отличаются, а это значит, что в современной сети, которой свойственна гетерогенность, т. е. наличие разнородности, фрагментация не является редким явлением.

Суть фрагментации — разбиение пакета, пришедшего из сети с б*о*льшим значением MTU и направляемого в сеть с меньшим значением MTU, на более короткие пакеты-фрагменты. Фрагмент, путешествуя по сети, может вторично подвергнуться фрагментации на каком-либо из промежуточных маршрутизаторов. Каждый из фрагментов должен быть снабжен полноценным заголовком IP.

Таблица 4.3

Значения MTU для различных сетей

Технология	MTU
DIX Ethernet	1500 байт
Ethernet 802.3	1492 байт
Token Ring (IBM, 16 Мбит/с)	17 914 байт
Token Ring (802.5, 4 Мбит/с)	4464 байт
FDDI	4352 байт
X.25	576 байт

Некоторые из полей заголовка, а именно, идентификатор, TTL, флаги DF и MF, а также смещение непосредственно предназначены для проведения последующей процедуры сборки фрагментов в исходное сообщение.

Получатель фрагмента использует идентификатор для того, чтобы опознать все фрагменты одного и того же пакета. Модуль IP, отправляющий пакет, устанавливает в поле идентификатора значение, которое должно быть уникальным для данной пары отправителя и получателя в течение всего времени, пока данный пакет (или любой его фрагмент) может существовать в составной IP-сети. Модуль IP может обеспечивать уникальность идентификаторов, например, поддерживая таблицу, где каждая запись соотносится с каждым отдельным получателем, с которым осуществлялась связь, и содержит последнее значение времени жизни пакета в IP-сети. Однако поскольку поле идентификатора допускает 65 536 различных значений, некоторые реализации IP выбирают из этого диапазона идентификаторы случайным образом, полагаясь на высокую вероятность того, что идентификатор окажется уникальным в течение времени передачи пакета.

Отправитель устанавливает в поле TTL время, в течение которого пакет может существовать в сети.

Поле смещения фрагмента предоставляет получателю информацию о положении фрагмента в исходном пакете. Так, первый фрагмент будет иметь в поле смещения нулевое значение. В пакете, не разбитом на фрагменты, поле смещения также имеет нулевое значение.

Флаг MF, установленный в значение 1, является признаком того, что пришедший фрагмент не является последним. Модуль IP, отправляющий нефрагментированный пакет, устанавливает флаг MF в нуль.

Флаг DF, установленный в значение 1, — признак того, что данный пакет не подлежит фрагментации ни при каких условиях. Если помеченный таким образом пакет не может достигнуть получателя без фрагментации, то модуль IP просто уничтожает пакет, а узлу-отправителю посылается диагностическое ICMP-сообщение.

Рассмотрим более подробно процедуру фрагментации.

Прежде чем разделить прибывший пакет на фрагменты, модуль протокола IP, установленный на маршрутизаторе, организует несколько буферов для новых пакетов-фрагментов. Затем он копирует в эти буферы содержимое некоторых полей заголовка IP из исходного пакета, создавая тем

самым «заготовки» заголовков IP всех новых пакетов-фрагментов. Одни параметры заголовка IP копируются в заголовки всех фрагментов, а другие остаются лишь в заголовке первого фрагмента. Процесс фрагментации может изменить значения некоторых полей заголовков IP в пакетах-фрагментах по сравнению с заголовком IP исходного пакета. Так, каждый фрагмент имеет собственное значение контрольной суммы заголовка, смещения фрагмента и общей длины пакета. Во всех пакетах, кроме последнего, флаг MF устанавливается в единицу, а в последнем фрагменте — в нуль.

Содержимое поля данных каждого фрагмента формируется в результате деления содержимого поля данных исходного пакета. При этом должны быть выполнены два условия. Во-первых, размер фрагмента (заголовка в сумме с полем данных) не должен превосходить MTU нижележащей технологии. Во-вторых, размер поля данных каждого фрагмента, кроме самого последнего, должен быть кратным 8 байт. Размер последней части данных равен полученному остатку.

Сборка пакета происходит на хосте назначения. Для этого на хосте назначения для каждого фрагментированного пакета отводится отдельный буфер, в который принимающий протокол IP помещает IP-фрагменты, у которых совпадают IP-адреса отправителя и получателя, а также значения в полях идентификатора и протокола. Все эти признаки говорят модулю IP, что данные пакеты являются фрагментами одного исходного пакета. Собственно сборка заключается в размещении данных из каждого фрагмента в позицию, определенную смещением, указанным в заголовке фрагмента.

Когда первый фрагмент исходного пакета приходит на хост-получатель, этот хост запускает таймер, который определяет максимальное время ожидания прибытия остальных фрагментов данного пакета. В различных реализациях IP применяются разные правила выбора максимального времени ожидания. В частности, таймер может быть установлен на фиксированный период времени (от 60 до 120 с), рекомендуемый RFC. Как правило, этот интервал достаточен для доставки пакета от отправителя получателю. В других реализациях максимальное время ожидания определяется с помощью адаптивных алгоритмов измерения и статистической обработки временных параметров сети, позволяющих оценивать ожидаемое время прибытия фрагментов. Наконец, тайм-аут может быть выбран на ба-

зе значений TTL прибывающих фрагментов. Последний подход основан на том, что нет смысла ожидать, пока прибудут другие фрагменты пакета, если время жизни одного из прибывших фрагментов уже истекло.

Признаком окончания сборки является отсутствие незаполненных промежутков в поле данных и прибытие последнего фрагмента (с равным нулю флагом MF). После того как данные собраны, их можно передавать вышележащему протоколу, например TCP.

4.3. ПРОТОКОЛ IPv6

В середине 90-х гг. начался лавинообразный рост интереса к Интернету, и стек протоколов TCP/IP столкнулся со следующими серьезными проблемами:

- резкий рост числа узлов сети, который породил проблему дефицита адресного пространства при принятой в IPv4 системе адресации;
- увеличение размера таблиц маршрутизации, вследствие роста числа узлов сети, что приводит к перегрузке маршрутизаторов и уменьшению скорости продвижения пакетов в сети;
- появление новых приложений и необходимость передачи разнородного трафика (речь, видео, данные и т.п.) при обеспечении требуемого качества его обслуживания (в первую очередь при передаче трафика реального времени);
- обострение проблемы обеспечения безопасности в IP-сетях.

Наряду с указанными, высветился еще ряд других проблем, таких, как необходимость упрощения многоадресных рассылок, поддержка мобильных хостов без изменения их адресации, обеспечение возможности дальнейшего развития протокола IP в будущем и т. п.

Перечисленные проблемы обусловлены следующими недостатками, свойственными протоколу IPv4:

- ограниченность и слабая масштабируемость адресного пространства;
- фактическое отсутствие встроенных механизмов поддержки качества обслуживания;
- слабая расширяемость протокола;
- отсутствие средств обеспечения безопасности на сетевом уровне;
- отсутствие механизма автоматической конфигурации адресов;
- проблемы, связанные с механизмом фрагментации.

Кратко рассмотрим указанные проблемы.

Ограниченность и слабая масштабируемость адресного пространства. Экспоненциальный рост числа хостов, подключенных к сети интернет, ведет к быстрому исчерпанию адресного пространства IPv4. Адрес IPv4 имеет длину 32 бита, что дает максимум 4 294 967 296 адресов (≈4,3 млрд.). В целях оптимизации использования ограниченного IPv4 адресного пространства был изменен метод адресации (адресация на основе масок и технология CIDR) и радикально уменьшены минимальные размеры выдаваемых блоков IP-адресов. Кроме того, начали развиваться технологии динамического назначения IP-адресов и технологии использования частного адресного пространства с системами трансляции сетевых адресов (NAT), позволяющие целой локальной сети занимать только один IP-адрес, который используется для взаимодействия с интернетом. Все эти меры позволили серьезно отсрочить исчерпание адресного пространства. Но лишь отсрочить, так как по некоторым оценкам исчерпание IPv4-адресного пространства произойдет в середине 2011 года. Кроме того, использование широко распространенной в IPv4 сетях технологии NAT может нарушать функционирование услуг, требующих передачи информации в зашифрованном виде, а это для мультимедийных многопотоковых услуг существенно снижает производительность оборудования. Поэтому применение прозрачной (прямой) адресации без необходимости трансляции сетевых адресов — серьезный аргумент в пользу IPv6.

Адресация IPv4 обладает и другим недостатком — слабой агрегацией адресов, что приводит к катастрофическому росту таблиц маршрутизации в магистральных маршрутизаторах и уменьшению скорости продвижения пакетов в сети.

Отсутствие встроенных механизмов поддержки качества обслуживания. С момента создания протокола IPv4 появилось множество новых сетевых приложений, таких, как потоковое аудио, потоковое видео и т. д., для нормальной работы которых требуется гарантированное обеспечение таких параметров передачи данных, как пропускная способность, задержка и вариация задержки. Набор таких параметров получил название качества услуг. Протокол IPv4 не может обеспечить предоставление гарантированного качества услуг. Для этой цели в заголовке IPv4 служит поле ToS (Type of Service), названное позднее «DS-байт», но ни механизм интерпре-

тации этого поля, ни механизм резервирования необходимых сетевых ресурсов в IPv4 определены не были, поэтому большинство существующих маршрутизаторов попросту игнорируют это поле в заголовке IPv4.

Слабая расширяемость протокола. В протоколе IPv4 предусмотрен единственный механизм расширения: добавление к заголовку IPv4 дополнительных опций. Однако общая длина всех опций не может превышать 40 байт, что крайне мало в современных условиях.

Отсутствие механизма автоматической конфигурации адресов. Изначально, с момента создания, в протокол IPv4 не было заложено механизма автоматического назначения адресов хостам сети (интерфейсам хостов). Эта операция обычно проводится сетевым администратором вручную либо полуавтоматически с использованием таких средств, как протоколы DHCP, RARP или BOOTP. Эта процедура является трудоемкой даже в малых сетях, а в больших сетях вручную попросту невозможна.

Проблемы, связанные с механизмом фрагментации. Одной из функций протоколов сетевого уровня является фрагментация слишком больших дейтаграмм перед посылкой их к следующему узлу (размер IP-пакета не должен превышать размера максимального блока данных, передаваемых в кадре канального уровня — MTU). Фрагментацию в протоколе IPv4 может осуществлять как отправитель, так и любой промежуточный маршрутизатор на пути следования пакета. Возможность фрагментации пакетов промежуточными маршрутизаторами в IPv4 ограничивает производительность этих маршрутизаторов, так как процедура фрагментации является очень ресурсоемкой. Фрагментации в промежуточных узлах можно избежать, но для этого отправитель должен определять размер MTU пути, по которому пойдет пакет. Для вычисления значения MTU пути в протоколе IPv4 существует механизм path MTU discovery, однако его применение не является обязательным для протокола IPv4.

Все эти недостатки IPv4 и привели к необходимости разработки IP протокола следующего поколения, который получил название IPv6. В этом протоколе были учтены недостатки протокола IPv4, а также были добавлены некоторые новые возможности, повышающие его эффективность и удобство использования.

Активные работы по модернизации протокола IP и разработке новых, ассоциированных с ним, протоколов начались в 1992 г. Документом, фик-

сирующим появление IPv6, стал RFC 1752. Базовый набор протоколов IPv6 был принят IETF в сентябре 1995 г. В августе 1998 г. были приняты пересмотренные версии группы стандартов, определяющих как общую архитектуру IPv6 (RFC 2460), так и его отдельные аспекты, например, систему адресации (RFC 2373[1]). Разработка этого протокола продолжается и по сей день, кроме того, для его испытаний в реальных условиях были созданы экспериментальные IPv6-сети.

В результате IPv4 был подвергнут серьезной переработке, что привело к появлению протокола IPv6, основные отличия которого по сравнению с IPv4 состоят в следующем:

1. *Новая масштабируемая система адресации:*

- увеличена разрядность адреса до 128 бит, в то время как в IPv4 было 32 бита;

- введено 4 уровня иерархии адресов вместо двух в IPv4, что позволяет эффективно использовать технологию CIDR (три из них используются для нумерации сетей, один — узлов).

2. *Снижение нагрузки на маршрутизаторы* за счет:

- отказа от обработки необязательных полей заголовка; заголовок делится на основной, который присутствует всегда и обрабатывается всеми маршрутизаторами сети, и опциональные (дополнительные), которые могут отсутствовать или не обрабатываться промежуточными маршрутизаторами;

- применения маршрутизации от источника;

- перенесения функций фрагментации на конечные узлы;

- агрегирования адресов и возможности использования в качестве номера узла его MAC-адреса.

3. *Предоставление гарантий качества транспортных услуг за счет введения в пакет поля метки*.

4. *Обеспечение защиты данных, передаваемых по сети, на сетевом уровне* (IPsec).

[1] Впоследствии на смену RFC 2373 пришли RFC 3513 (2003 г.) и RFC 4291 (2006 г.). Последний документ в настоящее время является основным и находится на стадии обсуждения интернет-сообществом.

В общем случае протокол IPv6 несовместим с протоколом IPv4, но зато обеспечивается его совместимость со всеми остальными протоколами Интернета, включая TCP, UDP, ICMP, OSPF, BGP, DNS и др.

Увеличение разрядности адресных полей у IPv6 до 16 байт обеспечивает практически неограниченный запас адресов, равный необозримо большому числу - $2^{128} \approx 10^{38}$.

Снижению нагрузки на маршрутизаторы способствует упрощение заголовка пакета, который состоит всего из 8 полей (вместо 13 у протокола IPv4). Поэтому маршрутизаторы могут быстрее обрабатывать пакеты, что повышает их производительность.

Рассмотрим формат основного заголовка IPv6.

Основной заголовок пакета IPv6 имеет фиксированную длину 40 байт и содержит следующие поля (рис. 4.3).

Поле *Класс Трафика* (Traffic Class) эквивалентно по назначению полю *Тип Обслуживания* в IPv4.

(Новое) поле *Метка Потока* (Flow Label) позволяет выделять и особым образом обрабатывать отдельные потоки данных без необходимости анализировать содержимое пакетов. Это важно как с точки зрения снижения нагрузки на маршрутизаторы, так и с позиций обеспечения требуемого QoS.

<table>
<tr><td>Версия
(4 бита)</td><td>Класс трафика
(8 бит)</td><td colspan="2">Метка потока (20 бит)</td></tr>
<tr><td colspan="2">Длина полезной нагрузки
(16 бит)</td><td>Следующий заголовок
(8 бит)</td><td>Лимит переходов
(8 бит)</td></tr>
<tr><td colspan="4">Адрес отправителя (128 бит)</td></tr>
<tr><td colspan="4">Адрес получателя (128 бит)</td></tr>
</table>

Рис. 4.3. Структура основного заголовка пакета IPv6

Поле *Следующий заголовок* определяет тип опционального заголовка (заголовка расширения), который следует за данным заголовком. Если дополнительных заголовков нет, то в нем будет записана версия протокола

(TCP, UDP, RIP, OSPF, …), данные которого переносятся IP-пакетом (в IPv4 это поле *Тип протокола*).

Поле *Длина полезной нагрузки* сообщает, сколько байт следует за основным 40-байтовым заголовком в пакете. В отличие от аналогичного поля *Полная длина пакета* в заголовке IPv4, в котором указывалась длина пакета вместе с заголовком, в IPv6 40-байтовый основной заголовок учитывается отдельно.

Поле *Лимит переходов* задает максимально возможное число транзитных участков при продвижении пакета по сети, что ограничивает время жизни пакета и не дает возможности пакетам вечно блуждать по сети. Назначение данного поля практически то же, что и поля *Время жизни* в заголовке протокола IPv4. Значение этого поля уменьшается на 1 при прохождении пакета через каждый транзитный узел. При лимите переходов, равном нулю, пакет удаляется.

Для представления адресов в протоколе IPv6 выбрана новая форма. Адрес IPv6 записывается в виде восьми групп по четыре шестнадцатеричных цифры, разделенных двоеточиями, например:

FEDC:0A98:0000:0000:0000:0000:7654:3210. (4.1)

Правило перевода из шестнадцатеричной формы представления IP-адреса в двоичную достаточно простое: каждая шестнадцатеричная цифра представляется четырьмя разрядами двоичного кода. Так, адрес (4.1) в двоичной форме записывается в виде

1111 1110 1101 1100 : 0000 1010 1001 1000:
0000 0000 0000 0000 : 0000 0000 0000 0000:
0000 0000 0000 0000 : 0000 0000 0000 0000:
1110 0110 0101 0100 : 0011 0010 0001 0000.

Поскольку многие адреса будут содержать большое количество нулей, предусмотрено три варианта сокращенной записи адресов. Во-первых, могут быть опущены ведущие нули в каждой группе, например, 0A98 можно записывать как A98. Во-вторых, одна или несколько групп, полностью состоящих из нулей, можно заменять парой двоеточий. К примеру, приведенный выше адрес (4.1) можно записать в сокращенном виде как:

FEDC:A98:0:0:0:0:7654:3210

или

FEDC:A98::7654:3210.

В-третьих, адреса IP-v4 могут записываться как пара двоеточий, после которой пишется адрес в старом десятичном формате, например:

::192.31.20.46.

В протоколе IPv6 предусмотрено три типа адресов (RFC 2373 от 1998 г.).

1. И*ндивидуальный адрес* (unicast address) определяет уникальный идентификатор отдельного порта конечного узла или маршрутизатора.

2. *Адрес произвольной рассылки* (anycast) идентифицирует группу интерфейсов разных маршрутизаторов, но пакет доставляется любому из интерфейсов группы, как правило, «ближайшему» в соответствии с метрикой протокола маршрутизации. Такие адреса используются при маршрутизации от источника, при которой узел-отправитель указывает маршрут путем задания адресов промежуточных маршрутизаторов. Порты маршрутизаторов группы в этом случае имеют как уникальные индивидуальные адреса, так и общий адрес произвольной рассылки.

3. *Групповой адрес* (multicast) идентифицирует группу интерфейсов, относящихся, как правило, к разным узлам. Аналогичен по назначению групповым адресам IPv4. Пакет доставляется всем узлам (интерфейсам), которые образуют группу с данным адресом. Групповые адреса в IPv6 используются также для замены широковещательных адресов IPv4. Для этого имеется адрес особой группы, объединяющей все интерфейсы подсети.

IPv6 адреса всех типов ассоциируются с интерфейсами, а не узлами. Так как каждый интерфейс принадлежит только одному узлу, индивидуальный адрес интерфейса может идентифицировать узел. Индивидуальный IPv6-адрес соотносится только с одним интерфейсом. Одному интерфейсу могут назначаться несколько IPv6-адресов различного типа (индивидуальные, групповые и произвольной рассылки).

Кроме указанных выше типов адресов, также как и в IPv4, имеются еще два адреса:

- *адрес обратной связи* (обратной петли) 0:0:0:0:0:0:0:1, который может использоваться для посылки IPv6-дейтограмм самому себе и который нельзя использовать в качестве идентификатора интерфейса (IPv6-пакет с адресом обратной связи в качестве адреса места назначения не может быть послан за пределы узла);

- *неопределенный адрес* 0:0:0:0:0:0:0:0, который имеет то же назначение, что и в IPv4, т. е. означает отсутствие адреса и используется в качестве адреса источника, когда тот еще не знает свой IP-адрес. Неопределённый адрес не должен использоваться в качестве указателя места назначения IPv6-дейтограмм или в IPv6 заголовках маршрутизации.

Типы адресов определяются несколькими старшими битами адреса, совокупность которых называется *префиксом формата* (Format Prefix, FP). Закрепление адресного пространства IPv6 на настоящее время показано в табл. 4.4.

Данное распределение адресов поддерживает прямое выделение индивидуальных адресов провайдера, адресов локального применения и групповых адресов. Групповые адреса отличаются от индивидуальных значением старшего октета: значение FF (1111 1111) идентифицирует групповой адрес; любые другие значения говорят о том, что адрес индивидуальный. Адреса произвольной рассылки берутся из пространства индивидуальных адресов и синтаксически неотличимы от них.

Таблица 4.4

Распределение адресного пространства IPv6 [RFC 2373]

Назначение	Префикс (двоичный)	Часть адресного пространства
Агрегируемые глобальные индивидуальные (unicast) адреса	001	1/8
Групповые (multicast) адреса	1111 1111	1/256
Локальные канальные адреса	1111 1110 10	1/1024
Локальные сетевые адреса (site)	1111 1110 11	1/1024
Зарезервировано для NSAP	0000 001	1/128
Зарезервировано для IPX	0000 010	1/128
Зарезервировано	0000 0000	1/256

Оставшаяся часть адресного пространства зарезервирована для будущего использования. Эти адреса могут использоваться для расширения имеющихся возможностей, например, дополнительных адресов провайдеров, или новых приложений. Пятнадцать процентов адресного пространства IPv6 уже распределено, остальные 85% зарезервированы.

Индивидуальные адреса делятся на несколько подтипов:

- глобальный агрегируемый индивидуальный адрес;
- адрес NSAP;
- иерархический адрес IPX;
- локальный адрес.

В будущем могут быть определены дополнительные типы адресов.

Основным подтипом индивидуальных адресов является глобальный агрегируемый индивидуальный адрес, структура которого показана на рис. 4.4.

Поле TLA (Top-Level Aggregation) предназначено для идентификации крупных провайдерских сетей и представляет собой префикс (общую часть пространства адресов) провайдера верхнего уровня. Количество адресов сетей крупных провайдеров на данном уровне сравнительно небольшое (<8196), что ограничивает число записей в таблицах магистральных маршрутизаторов и способствует ускорению их работы.

3 бита	13 бит	8 бит	24 бита	16 бит	64 бита
Префикс формата FP	Идентификатор верхнего уровня агрегирования TLA ID	Резерв	Идентификатор следующего уровня агрегирования NLA ID	Идентификатор местного уровня агрегирования SLA ID	Идентификатор интерфейса

Рис. 4.4. Структура глобального индивидуального адреса

Поле NLA (Next-Level Aggregation) предназначено для идентификации сетей средних (мелких) поставщиков услуг. Значительный размер данного поля (24 бита) позволяет адресовать большое количество провайдерских сетей среднего уровня. Совместно содержимое полей TLA и NLA несет информацию о провайдерских префиксах верхнего и среднего уровня.

Поле SLA (Site-Level Aggregation) предназначено для адресации подсетей одной корпоративной сети или отдельного абонента.

На нижнем уровне иерархии находится *идентификатор интерфейса* (узла), который является аналогом номера узла в адресе IPv4. В отличие от IPv4, в этом поле (64 бита) может записываться локальный (аппаратный) адрес, а не IP-адрес, назначаемый администратором узла. Здесь, в частности, может быть записан 48-битный MAC-адрес оборудования технологии

канального уровня, к примеру, Ethernet — 48 бит, адрес узла ATM — 48 бит и др.

Таким образом, глобальные индивидуальные адреса IPv6 имеют иерархическую структуру (рис. 4.5) и включают в себя все функции бесклассовой междоменной маршрутизации CIDR, реализованные для IPv4. Благодаря иерархической структуре адресов в протоколе IPv6 изначально заложена возможность агрегирования потоков (объединения подсетей) и разделения сетей на подсети. Это способствует снижению нагрузки на маршрутизаторы, обеспечивает возможность оптимизации распределения адресного пространства и построения сетей с гибкой структурой.

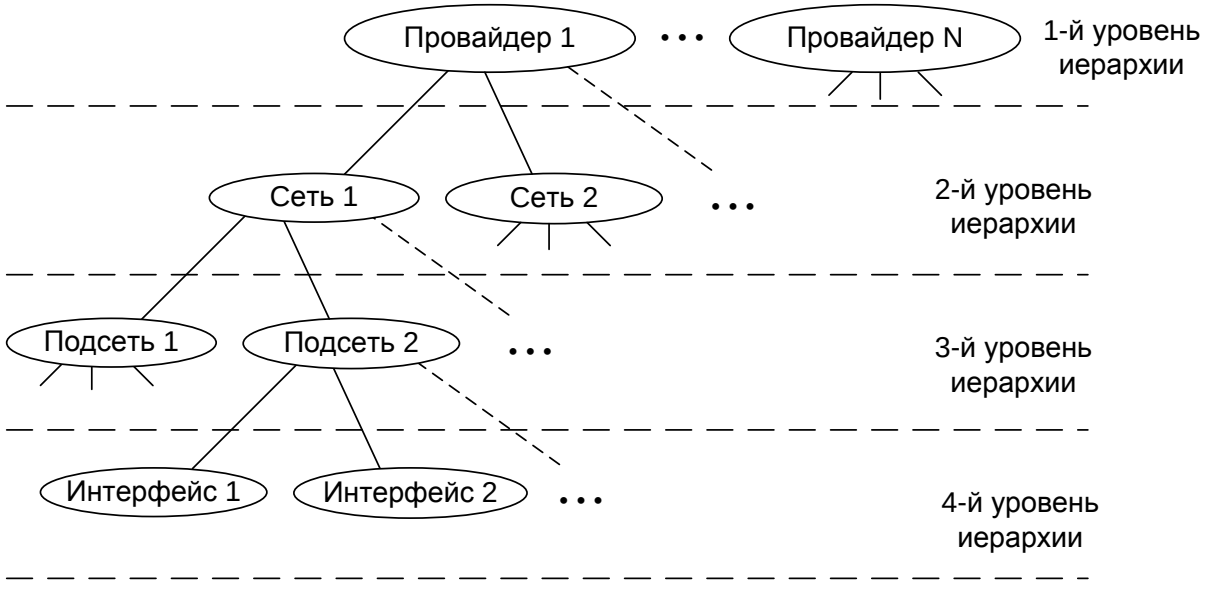


Рис. 4.5. Иерархия в назначении адресов IPv6

Необходимо отметить, что в документе RFC 4291 (2006 г.), который пришёл на смену RFC 2373, общий формат глобального индивидуального адреса несколько видоизменен (рис. 4.6). В отличие RFC 2373, в формате глобального индивидуального адреса, представленном в RFC 4291, упразднены поля TLA и NLA, место которых заняло поле глобального префикса маршрутизации. Поле SLA осталось, но имеет другое название: «Идентификатор подсети».

n бит	*m* бит	(128 – *n* – *m*) бит
Глобальный префикс маршрутизации	Идентификатор подсети	Идентификатор интерфейса

Рис. 4.6. Общий формат глобального индивидуального адреса

Глобальный префикс маршрутизации обычно имеет иерархическую структуру. При этом, если три старших двоичных разряда глобального префикса не содержат комбинацию 000, то длина идентификатора интерфейса равна 64 бит ($n + m = 64$). Примером таких адресов являются глобальные агрегируемые индивидуальные адреса, формат которых, согласно RFC 4291, имеет вид, представленный на рис. 4.7.

Считается, что такой формат глобального индивидуального адреса позволяет проводить более гибкую политику адресации в сравнении с форматом по RFC 2373.

3 бита	45 бит	16 бит	64 бита
001	Глобальный префикс маршрутизации	Идентификатор подсети	Идентификатор интерфейса

Рис. 4.7. Формат глобального агрегируемого индивидуального адреса

Идентификаторы интерфейсов служат для идентификации интерфейсов узлов и должны быть уникальными в пределах подсети. В качестве идентификаторов интерфейсов могут использоваться MAC-адреса оборудования технологий канального уровня. При этом допускается, что различные интерфейсы одного узла могут иметь одинаковый идентификатор интерфейса при условии, что они принадлежат различным подсетям. Идентификатор интерфейса записывается в формате EUI-64 [RFC 4291].

Если глобальный префикс начинается с 000, то поле идентификатора интерфейса не имеет ограничений по размеру и формату. Примером таких адресов являются IPv6-адреса с вложенным IPv4-адресом, которые представляют собой отдельный подкласс глобальных индивидуальных адресов, предназначенных для использования в смешанных IPv6 / IPv4 сетях на этапе перехода от IPv4 к IPv6.

Определены два типа IPv6-адресов с вложенным IPv4 адресом [RFC 4291]: IPv4-совместимые IPv6 адреса (IPv4-Compatible IPv6 address) и IPv4-отображенные IPv6 адреса (IPv4-mapped IPv6 address).

IPv4-совместимые IPv6-адреса имеют формат, представленный на рис. 4.8.

80 бит	16 бит	32 бита
0000……0000	0000	IPv4-адрес

Рис. 4.8. Формат IPv6 адреса, совместимого с IPv4

Старшие биты являются нулевыми, а в 32 младших разрядах записывается адрес IPv4, который должен быть уникальным в IPv4-сети. Эти адреса предназначались для использования в механизмах туннелирования при передаче IPv6-пакетов через фрагменты IPv4-сетей. В рекомендации RFC 4291 тип адресов помечен как не поддерживаемый в новых или модернизированных реализациях.

IPv4-отображенные IPv6 адреса имеют формат, представленный на рис. 4.9.

80 бит	16 бит	32 бита
0000……0000	FFFF	IPv4 адрес

Рис. 4.9. Формат IPv6 адреса, совместимого с IPv4

Они используются для представления адресов IPv4-узлов в формате IPv6-адреса в целях обеспечения взаимодействия приложений, ориентированных на IPv6 с сетевым протоколом IPv4 (рис. 4.10). Порядок их использования определен в RFC 4038.

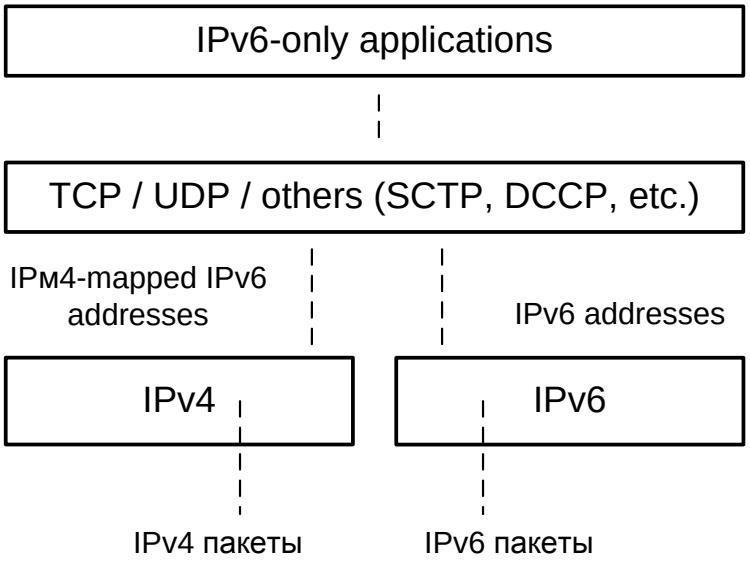


Рис. 4.10. Использование IPv4-отображенных IPv6-адресов

В IPv6 существует два типа индивидуальных адресов для локального использования: *локальный адрес линии* (Link-local-use) и *локальный адрес сети* (Site-local-use).

Локальный адрес линии (канала) предназначен для работы с одним каналом в целях автоконфигурации адресов, поиска соседних узлов или при отсутствии маршрутизатора. Он имеет формат, показанный на рис. 4.11. Маршрутизаторы не должны переадресовывать пакеты с локальными канальными адресами отправителя в другие линии.

10 бит	54 бита	64 бита
1111111010	0	Идентификатор интерфейса

Рис. 4.11. Структура локального адреса канала

Локальный адрес сети предназначен для использования в одной локальной сети (site). Он имеет формат, показанный на рис. 4.12.

Маршрутизаторы не должны переадресовывать пакеты с локальными адресами сети, отправителя за ее пределы. Локальные IPv6-адреса сети выполняют функции частных IPv4-адресов и могут использоваться в локальных сетях или в сетях организаций, которые (пока еще) не подключены к интернету. Когда организация соединяется с глобальным интернетом,

она может сформировать глобальные адреса путем замещения локального префикса сети префиксом поставщика услуг.

10 бит	54 бита	64 бита
1111111011	Идентификатор подсети	Идентификатор интерфейса

Рис. 4.12. Структура локального адреса сети

В заключение обсуждения вопросов адресации в протоколе IPv6 следует отметить, что адресное пространство IPv6 будет распределяться специальной комиссией по стандартным числам IANA (Internet Assigned Numbers Authority). В качестве советников будут выступать Совет по архитектуре интернета IAB (Internet Architecture Board) и инженерная группа управления интернетом IESG (Internet Engineering Steering Group). При этом IANA будет делегировать права выдачи IP-адресов региональным сервис-провайдерам, субрегиональным структурам и организациям. Отдельные лица и организации могут получить адреса непосредственно от регионального распределителя или сервис провайдера.

Дополнительные заголовки (заголовки расширения).

Заголовки расширения являются необязательными и предназначены для переноса дополнительной информации в промежуточных маршрутизаторах и хостах о различных параметрах продвижения и обработки пакета (поддержка механизмов безопасности, фрагментация, сетевое управление и т. п.). При их наличии заголовки расширения размещаются вслед за основным заголовком до поля данных пакета IPv6. У некоторых заголовков формат фиксированный, другие содержат переменное количество полей переменной длины. На настоящий момент определено несколько типов заголовков расширения, которые представлены в табл. 4.5.

Заголовок параметров ретрансляции (Hop-by-Hop Options) определяет специальные параметры обработки пакетов на каждом ретрансляционном участке (в каждом промежуточном маршрутизаторе).

Таблица 4.5

Дополнительные заголовки IPv6

Тип заголовка расширения	№ типа	Размер	RFC
Hop-by-Hop Options (параметры ретрансляции)	0	—	RFC 2460
Routing (заголовок маршрутизации)	43	—	RFC 2460 RFC 3775 RFC 5095
Fragmentation (заголовок фрагментации)	44	64 бита	RFC 2460
Authentication (заголовок аутентификации)	51	—	RFC 4302
ESP (заголовок системы безопасности)	50	—	RFC 4303
Destination Options (параметры получателя)	60	—	RFC 2460
No Next Header (нет следующего заголовка)	59	0	RFC 2460

Этот заголовок расширения имеет наиболее общий формат и включает следующие поля (рис. 4.13):

- *Next Header* — номер (тип) следующего за данным заголовка (8 бит);

- *Hdr Ext Len* — длина данного дополнительного заголовка (8 бит), не считая первых 64 бит, причем единицей измерения служат 64-битные слова;

- *Options* — содержимое дополнительного заголовка (поле опций или параметров переменной длины).

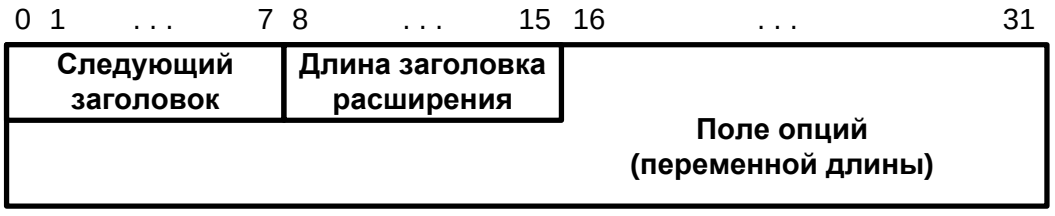


Рис. 4.13. Формат заголовка параметров ретрансляции

В свою очередь, поле опций (параметров) состоит из одного или нескольких фрагментов, каждый из которых включает три поля:

- тип параметра (8 бит) — идентифицирует параметр;
- длина (8 бит);
- данные параметра (переменной длины).

Заголовок ретрансляционных участков, обрабатываемый маршрутизаторами, может использоваться, например, для резервирования ресурсов по протоколу RSVP перед прохождением IP-потока, чувствительного к качеству обслуживания. Еще одно применение — для указания того, что передается так называемая «джамбограмма», т. е. IP-пакет длиной более 64 Кбайт (65535 бит). В режиме *Jumbo* поддерживается пересылка пакетов размером до 4 Гбайт.

Длина джамбограммы в байтах задается 32-битным целым числом (поле Payload Length в стандартном заголовке IPv6 должно быть нулевым). Способ представления этой опции показан на рис. 4.14.

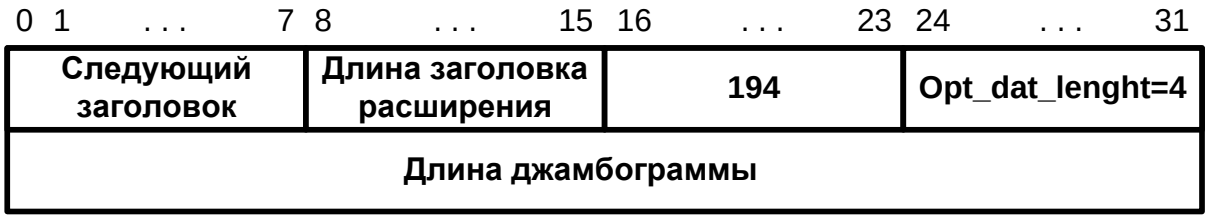


Рис. 4.14. Формат дополнительного заголовка Hop-by-Hop при передаче джамбограмм

Заголовок маршрутизации (Routing Header) содержит информацию о маршруте, выбранном отправителем, что позволяет использовать маршрутизацию от источника и освободить транзитный маршрутизатор от просмотра адресных таблиц при выборе следующего маршрутизатора. Здесь источник задает список узлов сети, через которые должен пройти пакет (в зависимости от режима маршрутизации разрешается или запрещается проход через дополнительные узлы, не указанные в списке). Формат заголовка маршрутизации показан на рис. 4.15.

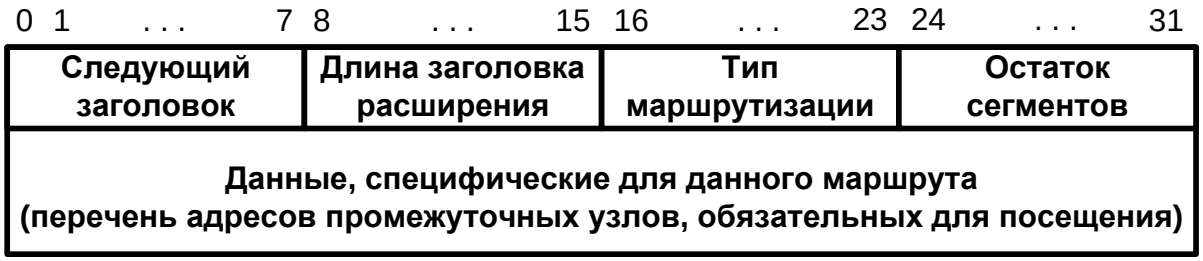


Рис. 4.15. Формат заголовка маршрутизации

Поле *Тип маршрутизации* идентифицирует вариант заголовка маршрутизации. Если маршрутизатор не распознает значение этого поля, то он отбрасывает пакет.

Поле *Остаток сегментов* — количество явно указанных промежуточных узлов, которые должен посетить пакет.

Определен (RFC 2460) один вариант заголовка маршрутизации типа 0. При этом в поле данных заголовка помещается список адресов обязательных для посещения промежуточных узлов маршрута. Адрес получателя — последний в списке, а в поле адреса получателя основного заголовка IPv6 записывается адрес очередного (из списка) маршрутизатора на пути. После посещения каждого маршрутизатора из списка содержимое поля адреса получателя основного заголовка IPv6 обновляется.

Заголовок фрагмента. В протоколе IPv6 фрагментация может производиться только узлами-источниками, но не промежуточными маршртутизаторами. Формат дополнительного заголовка фрагментации показан на рис. 4.16.

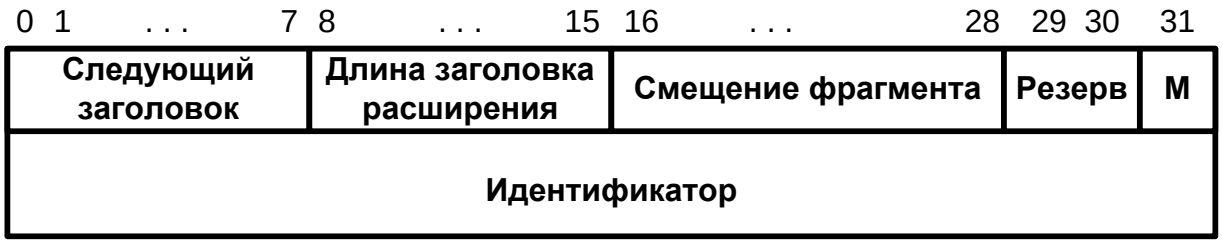


Рис. 4.16. Формат дополнительного заголовка фрагментации

Поле *Смещение фрагмента* указывает, к какой части исходного пакета принадлежит полезная нагрузка данного фрагмента. Она измеряется в единицах по 64 бита, т. е. длина всех фрагментов в байтах, кроме последнего, должна быть кратна 8.

Флаг *M*: если *M* = 1, то это означает, что ещё есть фрагменты, если *M* = 0 — это последний фрагмент.

Идентификатор (32 бита) предназначен для уникальной идентификации исходного пакета в течение времени нахождения дейтаграммы в сети. Все фрагменты с одинаковым идентификатором, адресом отправителя и адресом получателя собираются вместе для дефрагментации в конечном узле.

4.4. ПРОТОКОЛЫ ТРАНСПОРТНОГО УРОВНЯ TCP И UDP

Главная задача транспортного уровня заключается в передаче данных между прикладными процессами. Эту задачу решают *протокол управления передачей* (Transmission Control Protocol, TCP), описанный в RFC 793, и *протокол пользовательских дейтаграмм* (User Datagram Protocol, UDP), описанный в RFC 768. Протоколы TCP и UDP имеют много общего. И тот, и другой обеспечивают интерфейс с вышележащим прикладным уровнем, передавая данные, поступающие на входной интерфейс хоста, соответствующему приложению. При этом оба протокола используют концепции *порт* и *сокет* (гнездо). Оба они также поддерживают интерфейс с нижележащим сетевым уровнем IP, упаковывая свои PDU в IP-пакеты. Протокольные сущности TCP и UDP, как и в случае протоколов прикладного уровня, устанавливаются только на конечных узлах. Однако, как будет сказано ниже, различий между TCP и UDP гораздо больше, чем сходств.

Каждый компьютер может выполнять несколько процессов; более того, прикладной процесс тоже может иметь несколько точек входа, выступающих в качестве адреса назначения для пакетов данных. Поэтому после того, как пакет средствами протокола IP доставлен на сетевой интерфейс компьютера-получателя, данные необходимо переправить конкретному процессу-получателю.

Существует и обратная задача: пакеты, которые отправляют в сеть разные приложения, работающие на одном конечном узле, обрабатываются общим для них протоколом IP. Следовательно, в стеке должно быть

предусмотрено средство сбора пакетов от разных приложений для передачи протоколу IP. Эту работу выполняют протоколы TCP и UDP. Процедура приема данных протоколами TCP и UDP, поступающих от нескольких различных прикладных служб, называется мультиплексированием. Обратная процедура — процедура распределения протоколами TCP и UDP, поступающих от сетевого уровня пакетов между набором высокоуровневых служб, — называется демультиплексированием (рис. 4.17).

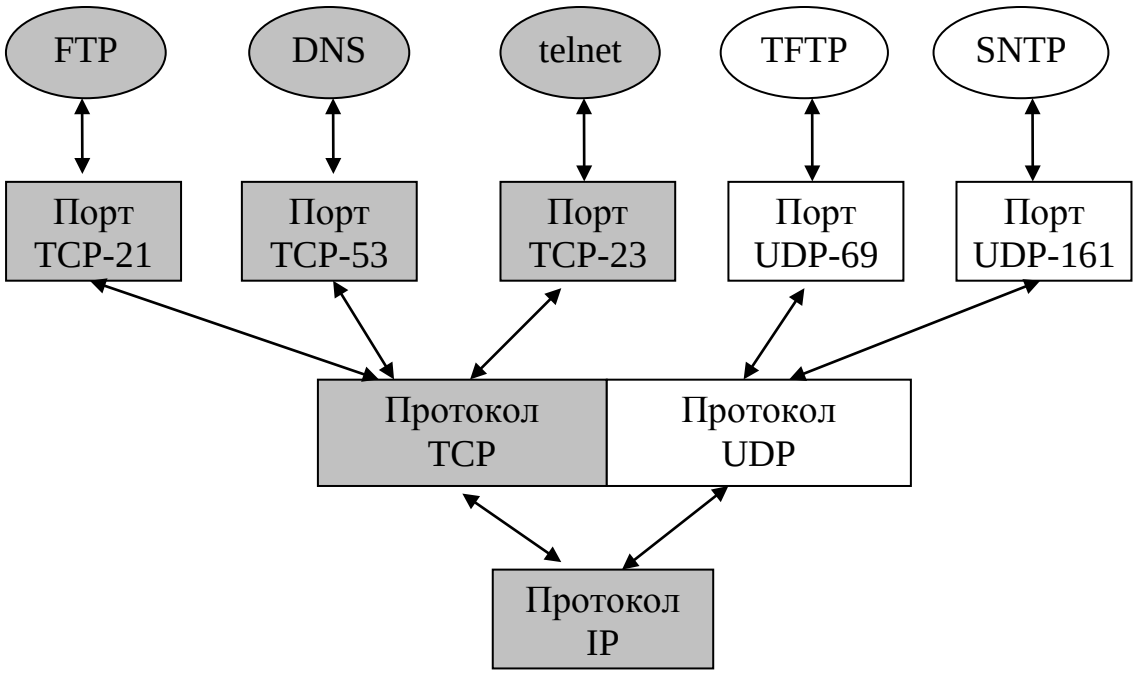


Рис. 4.17. Мультиплексирование и демультиплексирование на транспортном уровне

Протоколы TCP и UDP ведут для каждого приложения две очереди: очередь пакетов, поступающих к данному приложению из сети, и очередь пакетов, отправляемых данным приложением в сеть. Пакеты, поступающие на транспортный уровень, организуются операционной системой в виде множества очередей к точкам входа различных прикладных процессов. В терминологии TCP / IP такие системные очереди называются *портами* (порты приложения не надо путать с портами как сетевыми интерфейсами оборудования), причем входная и выходная очереди одного приложения рассматриваются как один порт. Для однозначной идентификации

портов им присваивают номера. Номера портов используются для адресации приложений.

Если процессы представляют собой популярные общедоступные службы, такие как FTP, telnet, HTTP, TFTP, DNS и т. п., то за ними закрепляются стандартные, назначенные номера, также называемые хорошо известными (well-known) номерами портов. Эти номера закрепляются и публикуются в стандартах Интернета (RFC 1700, RFC 3232). Так, номер 21 закреплен за службой удаленного доступа к файлам FTP, а номер 23 — за службой удаленного управления telnet. Назначенные номера являются уникальными в пределах Интернета и назначаются приложениям централизованно из диапазона от 0 до 1023.

Для тех приложений, которые еще не стали столь распространенными, чтобы закреплять за ними стандартные номера, номера портов назначаются разработчиками этих приложений или операционной системой локально в ответ на поступление запроса от приложения. На каждом компьютере операционная система ведет список занятых и свободных номеров портов. При поступлении запроса от приложения, выполняемого на данном компьютере, операционная система выделяет ему первый свободный номер. Такие номера называют динамическими.

Все, что было сказано о портах, в равной степени относится к обоим протоколам транспортного уровня (TCP и UDP). В принципе нет никакой зависимости между назначением номеров для приложений, использующих протокол TCP, и приложений, работающих с протоколом UDP. Приложения, которые передают данные на уровень IP по протоколу UDP, получают номера, называемые UDP-портами. Аналогично приложениям, обращающимся к протоколу TCP, выделяются TCP-порты.

Рассмотрим более подробно организацию и функционирование протокола UDP.

Протокол UDP, являясь дейтаграммным, реализует *услугу по возможности*, т. е. не гарантирует доставку своих сообщений, а, следовательно, никаким образом не компенсирует ненадёжность дейтаграммного протокола IP.

Единица данных протокола UDP называется UDP-дейтаграммой, или пользовательской дейтаграммой. Каждая дейтаграмма переносит отдельное пользовательское сообщение, что приводит к естественному ограниче-

нию: длина дейтаграммы UDP не может превышать длины поля данных протокола IP, которое, в свою очередь, ограничено размером кадра технологии нижнего уровня. Поэтому, если UDP-буфер переполняется, то данные приложения отбрасываются.

Заголовок UDP, состоящий из четырех 2-байтовых полей, содержит номера портов отправителя и получателя, контрольную сумму и длину дейтаграммы.

Рассмотрим, как протокол UDP решает задачу демультиплексирования. Казалось бы, для этой цели достаточно использовать номера портов. Кадры, несущие UDP-дейтаграммы, пребывают на сетевой интерфейс хоста, последовательно обрабатываются протоколами стека и, наконец, поступают в распоряжение протокола UDP, который извлекает из заголовка номер порта назначения и передает данные на соответствующий порт соответствующему приложению, т. е. выполняет демультиплексирование.

Это решение выглядит очень логично и просто, однако оно неработоспособно в ситуации, когда на одном конечном узле выполняется несколько копий одного и того же приложения. Пусть, например, на одном хосте запущены два DNS-сервера, причем оба используют для передачи своих сообщений протокол UDP (рис. 4.18). DNS-сервер имеет хорошо известный UDP-порт 53. В то же время, у каждого из DNS-серверов могут быть свои клиенты, собственные базы данных, собственные настройки. Когда на сетевой интерфейс данного компьютера придет запрос от DNS-клиента, в UDP-дейтаграмме будет указан номер порта 53, который в равной степени относится к обоим DNS-серверам.

Чтобы снять такую неоднозначность, применяют следующий подход. Разным копиям одного приложения, даже установленным на одном компьютере, присваивают разные IP-адреса. В данном примере DNS-сервер 1 имеет IP-адрес IP1 а DNS-сервер 2 — IP-адрес IP2. Таким образом, однозначно определяет прикладной процесс в сети (а тем более, в пределах компьютера) и пара (IP-адрес, номер порта UDP), называемая UDP-сокетом (UDP socket). Таким образом, протокол UDP выполняет демультиплексирование на основе сокетов.

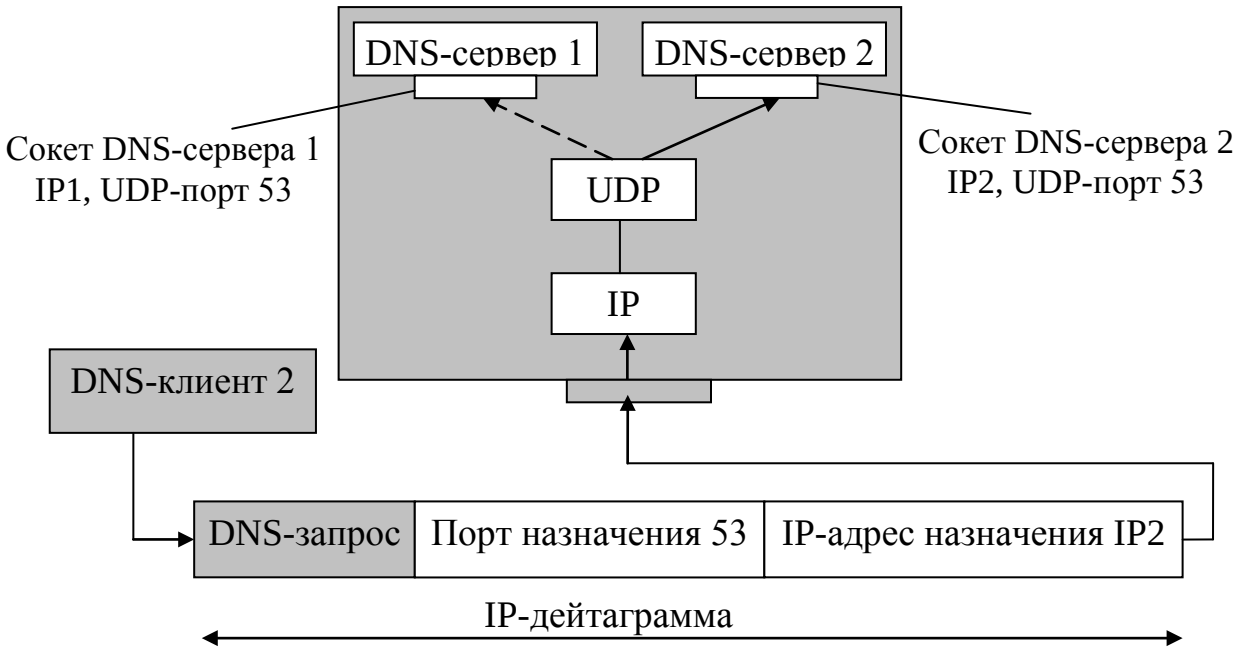


Рис. 4.18. Демультиплексирование протокола UDP на основе сокетов

Обратимся теперь к особенностям организации и функционирования протокола TCP.

Информация, поступающая к протоколу TCP от протоколов более высокого уровня, рассматривается протоколом TCP как неструктурированный поток байтов. Поступающие данные буферизуются средствами TCP. Для передачи на сетевой уровень из буфера «вырезается» некоторая непрерывная часть данных, которая называется *сегментом* и снабжается заголовком (рис. 4.19).

Заголовок TCP-сегмента (рис. 4.20) содержит значительно больше полей, чем заголовок UDP, что отражает более развитые возможности этого протокола.

Порт источника (source port) занимает 2 байта и идентифицирует процесс-отправитель.

Порт приемника (destination port) занимает 2 байта и идентифицирует процесс-получатель.

Последовательный номер (sequence number) занимает 4 байта и представляет собой номер байта, который определяет смещение сегмента относительно потока отправляемых данных (другими словами, номер первого байта данных в сегменте).

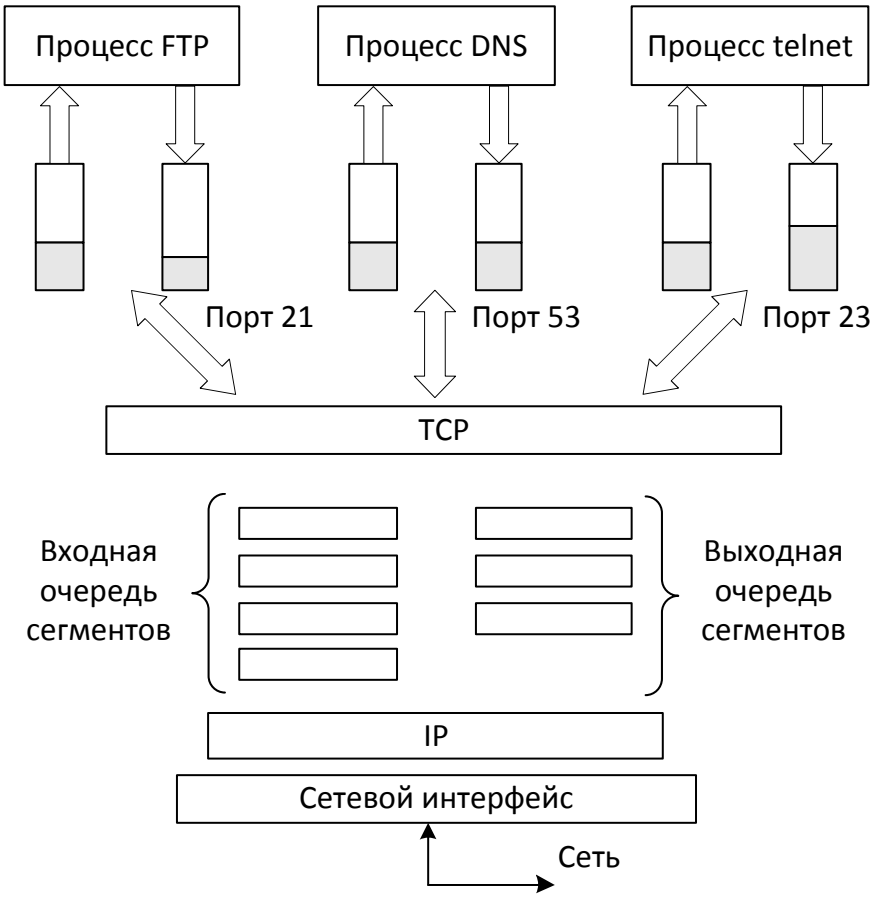


Рис. 4.19. Формирование TCP-сегментов из потока байтов

Порт источника (2 байта)								**Порт назначения** (2 байта)	
Последовательный номер (4 байта)									
Подтвержденный номер (4 байта)									
Длина заголовка (4 бита)	Резерв (6 бит)	Флаги						**Окно** (2 байта)	
		URG	ACK	PSH	RST	SIN	FIN		
Контрольная сумма (2 байта)								**Указатель срочности** (2 байта)	
≈ **Параметры** (0 или более 32-разрядных слов) ≈									
≈ **Данные** (необязательное поле) ≈									

Рис. 4.20. Формат TCP-сегмента

Подтвержденный номер (acknowledgement number) занимает 4 байта и содержит максимальный номер байта в полученном сегменте, увеличенный на единицу. Именно это значение используется в качестве квитанции. Если установлен контрольный бит АСК, то это поле содержит следующий номер очереди, который отправитель данного сегмента желает получить в обратном направлении.

Длина заголовка занимает 4 бита и представляет собой длину заголовка TCP-сегмента, измеренную в 32-битовых словах. Длина заголовка не фиксирована и может изменяться в зависимости от значений, устанавливаемых в поле параметров.

Резерв (reserved) занимает 6 бит.

Кодовые биты (code bits) числом 6 содержат служебную информацию о типе данного сегмента. Положительное значение сигнализируется установкой этих битов в единицу:

- URG — срочное сообщение;
- АСК — квитанция на принятый сегмент;
- PSH — запрос на отправку сообщения без ожидания заполнения буфера (протокол TCP может выжидать заполнения буфера перед отправкой сегмента, но если требуется срочная передача, то приложение сообщает об этом протоколу TCP с помощью данного бита);
- RST — запрос на восстановление соединения;
- SYN — сообщение, используемое для синхронизации счетчиков переданных данных при установлении соединения;
- FIN — признак достижения передающей стороной последнего байта в потоке передаваемых данных.

Окно (window) занимает 2 байта и задает количество байтов данных, ожидаемых отправителем данного сегмента, начиная с байта, номер которого указан в поле подтвержденного номера.

Контрольная сумма (checksum) занимает 2 байта.

Указатель срочности (urgent pointer) занимает 2 байта и указывает на конец данных, которые необходимо срочно принять, несмотря на переполнение буфера. Указатель срочности используется совместно с кодовым битом URG. Т. е., если какие-то данные необходимо переслать приложению-получателю вне очереди, то приложение-отправитель должно сообщить об этом протоколу TCP путем установки в единицу бита URG.

Параметры (options) имеют переменную длину и могут вообще отсутствовать. Максимальная величина поля составляет 3 байта; оно используется для решения вспомогательных задач, например для выбора максимального размера сегмента. Поле параметров может располагаться в конце заголовка TCP, а его длина кратна 8 бит.

Заполнитель (padding) может иметь переменную длину. Это фиктивное поле, используемое для доведения размера заголовка до целого числа 32-битовых слов.

При установлении логического соединения модули TCP «договариваются» между собой о параметрах процедуры обмена данными. В протоколе TCP каждая сторона соединения посылает противоположной стороне следующие параметры:

- максимальный размер сегмента, который она готова принимать;
- максимальный объем данных (возможно несколько сегментов), которые она разрешает другой стороне передавать в свою сторону, даже если та еще не получила квитанцию на предыдущую порцию данных (размер окна);
- начальный порядковый номер байта, с которого она начинает отсчет потока данных в рамках данного соединения.

В результате процесса взаимодействия модулей TCP с двух сторон соединения определяются параметры соединения. Одни из них остаются постоянными в течение всего сеанса связи, а другие адаптивно изменяются. В частности, в зависимости от загрузки буфера принимающей стороны, а также надежности работы сети динамически изменяется размер окна отправителя. Создание соединения означает также выделение операционной системой на каждой стороне соединения определенных системных ресурсов: для организации буферов, таймеров, счетчиков. Эти ресурсы будут закреплены за соединением с момента создания и до момента разрыва. При этом каждый сокет одновременно может участвовать в нескольких соединениях.

В рамках установленного соединения в протоколе TCP правильность передачи каждого сегмента должна подтверждаться квитанцией от получателя. Квитирование — это один из традиционных методов обеспечения надежной связи. В протоколе TCP используется частный случай квитирования — *алгоритм скользящего окна*.

При установлении соединения обе стороны договариваются о начальном номере байта, с которого будет вестись отсчет в течение всего данного соединения. У каждой стороны свой начальный номер. Идентификатором каждого сегмента является номер его первого байта. Нумерация байтов в пределах сегмента осуществляется так, что первый байт данных сразу вслед за заголовком имеет наименьший номер, а следующие за ним байты имеют следующие порядковые номера.

Когда отправитель посылает TCP-сегмент, он в качестве идентификатора сегмента помещает в поле последовательного номера номер первого байта данного сегмента. Так, на рис. 4.21 идентификаторами сегментов являются номера 32600, 34060, 35520 и т. д. На основании этих номеров TCP-получатель не только отличает данный сегмент от других, но и позиционирует полученный фрагмент относительно общего потока байтов. Кроме того, он может сделать вывод, что полученный сегмент является дубликатом или что между двумя полученными сегментами пропущены данные.

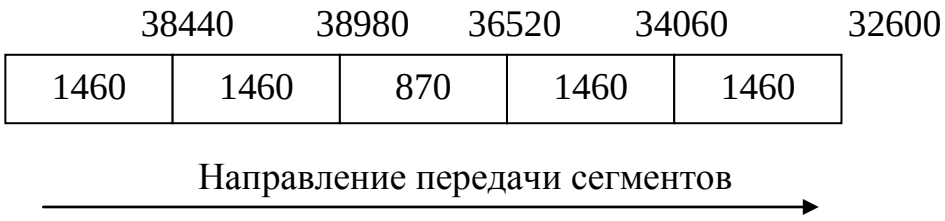


Рис. 4.21. Порядковый номер и номер квитанции

В качестве квитанции получатель сегмента отсылает ответное сообщение (сегмент), в которое помещает число (подтверждающий номер), на единицу превышающее максимальный номер байта в полученном сегменте. Для сегментов, изображенных на рис. 4.21, квитанцией о получении (подтвержденным номером) служат номера последнего байта каждого сегмента +1. Так для первого отправленного сегмента это будет число 34060, для второго — 36520 и т. д. Подтверждающий номер часто интерпретируют как номер следующего ожидаемого байта данных. Квитанция (подтверждение) в протоколе TCP посылается только в случае правильного приема данных, отрицательные квитанции не посылаются. Таким образом,

отсутствие квитанции означает либо потерю сегмента, либо прием искаженного сегмента, либо потерю квитанции.

В протоколе TCP в одном и том же сегменте могут быть помещены и данные, которые посылает приложение другой стороне, и квитанция, которой модуль TCP подтверждает получение данных.

Протокол TCP является дуплексным, т. е. в рамках одного соединения регламентируется процедура обмена данными в обе стороны. Каждая сторона одновременно выступает и как отправитель, и как получатель. При этом у каждой стороны есть пара буферов: один — для хранения принятых сегментов, другой — для сегментов, которые только еще предстоит отправить. Кроме того, имеется буфер для хранения копий сегментов, которые были отправлены, но квитанции, о получении которых еще не поступили (рис. 4.22). И при установлении соединения, и в ходе передачи обе стороны, выступая в роли получателя, посылают друг другу так называемые *окна приема*. Каждая из сторон, получив окно приема, понимает, сколько байт ей разрешается отправить с момента получения последней квитанции. Другими словами, посылая окна приема, обе стороны пытаются регулировать поток байтов в свою сторону, сообщая противоположной стороне, какое количество байт (начиная с номера байта, о котором уже была выслана квитанция) они готовы в настоящий момент принять.

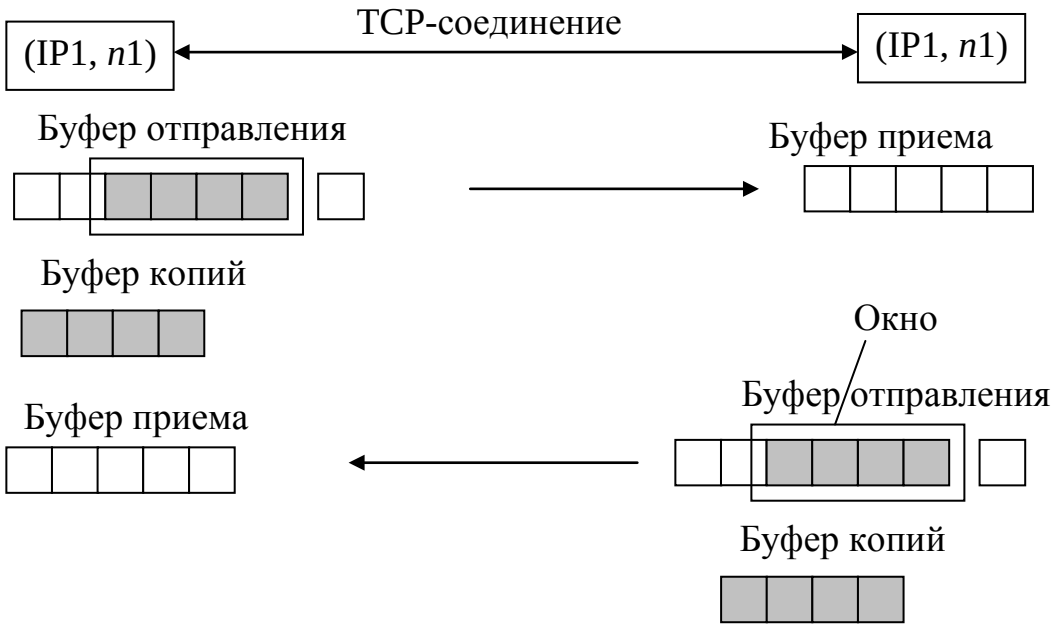


Рис. 4.22. Система буферов TCP-соединения

Получатель может послать квитанцию, подтверждающую получение сразу нескольких сегментов, если они образуют непрерывный поток байтов. Например (рис. 4.23, *а*), если в буфер плотно без пропусков заполненный потоком байтов до 2354 включительно поочередно поступили сегменты (2355–3816), (3817–5275) и (5276–8400), где цифры в скобках означают номера первых и последних байт каждого сегмента, то получателю достаточно отправить только одну квитанцию на все три сегмента, указав в ней в качестве номера квитанции значение 8401. Таким образом, процесс квитирования является накопительным.

Однако сегменты могут прийти к получателю не в том порядке, в котором были посланы, т. е. в приемном буфере может образоваться «разрыв» (рис. 4.23, *б*). Пусть, к примеру, после указанных выше трех сегментов вместо следующего по порядку сегмента (8401–10566) пришел сегмент (10567–12430). Очевидно, что послать в качестве номера квитанции значение 12 431 нельзя, потому что это бы означало, что получены все байты вплоть до 12 430. Поскольку в потоке байт образовался разрыв, получатель может только еще раз повторить квитанцию 8401, говоря, тем самым, что все еще ожидает поступления потока байт, начиная с 8401.

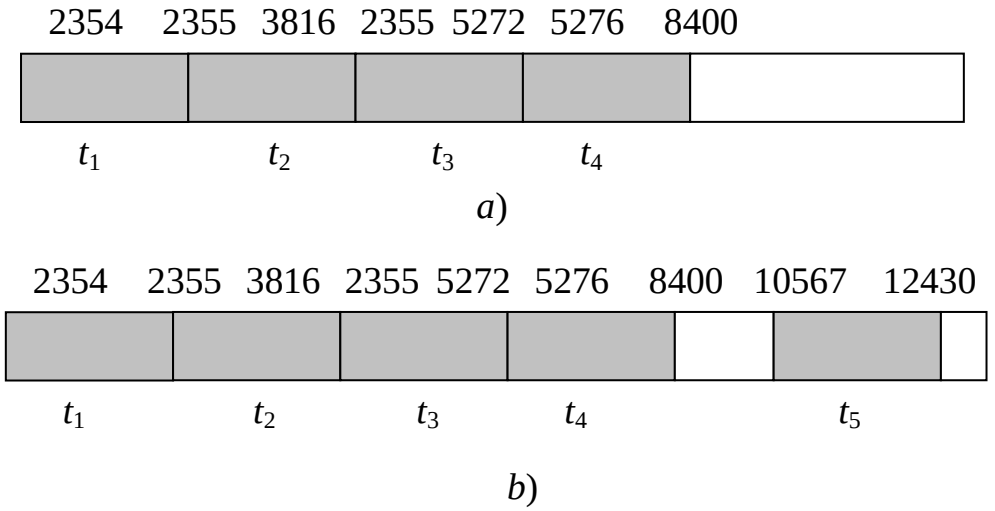


Рис. 4.23. Накопительный принцип квитирования: *а* — плотное заполнение буфера; *б* — неплотное заполнение буфера

Из этого примера видно, что, в отличие от многих других протоколов, протокол TCP подтверждает получение не отдельных блоков данных, а непрерывной последовательности байт.

Когда протокол TCP передает в сеть сегмент, он дополнительно помещает его копию в очередь повторной передачи и запускает таймер. Когда приходит квитанция на этот сегмент, соответствующая копия удаляется из очереди. Если же квитанция не приходит до истечения срока, то сегмент посылается повторно. Может случиться так, что повторный сегмент придет тогда, когда исходный сегмент уже окажется на месте, тогда дубликат будет попросту отброшен. Возникает вопрос: каким должно быть время ожидания (тайм-аут) очередной квитанции? От решения этой задачи зависит производительность протокола TCP. Тайм-аут не должен быть слишком коротким, чтобы по возможности исключить избыточные повторные передачи, снижающие полезную пропускную способность системы, но также он не должен быть и слишком длинным, чтобы избежать длительных простоев, связанных с ожиданием несуществующей или «заблудившейся» квитанции.

При выборе величины тайм-аута должны учитываться скорость и надежность линий связи, их протяженность и многие другие факторы. В протоколе TCP тайм-аут определяется с помощью достаточно сложного адаптивного алгоритма, идея которого состоит в следующем. При каждой передаче засекается время от момента отправки сегмента до прихода квитанции о его приеме (время оборота). Получаемые значения времени оборота усредняются с весовыми коэффициентами, возрастающими от предыдущего замера к последующему. Это делается с тем, чтобы усилить влияние последних замеров. В качестве тайм-аута выбирается среднее время оборота, умноженное на некоторый коэффициент. Практика показывает, что значение этого коэффициента должно превышать 2. В сетях с большим разбросом времени оборота при выборе тайм-аута учитывается и дисперсия этой величины.

ВОПРОСЫ И ЗАДАНИЯ К ГЛАВЕ 4

1. Сравните достоинства и недостатки адресации на основе классов и масок?

2. Маска подсети равна 255.255.220.0. Какое максимальное количество хостов в этой сети?

3. Пусть код, обнаруживающий ошибки в протоколе IP, может обнаруживать ошибки кратности не выше трёх. На промежуточный маршрутизатор прибыл IP-пакет, содержащий в заголовке 4 ошибки. Что произойдет с этим пакетом, если ошибки сосредоточены в поле «IP-адрес назначения», в поле «Протокол верхнего уровня» или в поле «Смещение фрагмента»?

4. Поясните способ сборки фрагментированного пакета в пункте назначения.

5. Пусть в сети протокол IPv4 работает поверх Ethernet. Необходимо передать IP-пакет размером 4480 байт. Укажите, на сколько фрагментов этот пакет будет фрагментирован, а также значения полей «Общая длина», «Смещение фрагмента» и флага M для каждого из получающихся в результате фрагментов.

6. Можно ли агрегировать IP-адреса 192.6.96.0/21, 192.6.112.0/21 и 192.6.120.0/21, если для них используется одна исходящая линия. Если да, то какова будет агрегированная запись адреса? Если нет, то почему?

7. На какое время хватит адресного пространства протокола IPv6, если каждую пикосекунду назначать блок размером 1 млн. адресов?

8. Почему в протоколе IPv6, в отличие от IPv4, не предусматривается возможность фрагментации пакетов в промежуточных маршрутизаторах?

9. Какие механизмы обеспечения гарантий надежной доставки пакетов применяются в протоколе TCP?

10. Для чего служит механизм скользящего окна?

11. В протоколе IP контрольная сумма вычисляется только по заголовку пакета, а в TCP она покрывает весь сегмент. Поясните почему?

5. МАРШРУТИЗАЦИЯ В ОБЪЕДИНЕННЫХ СЕТЯХ

Важнейшей функцией сетевого уровня, наряду с объединением разнородных сетей, является маршрутизация пакетов. Исторически маршрутизация применялась для определения оптимальных по какому-либо критерию путей (с минимальной задержкой, стоимостью, с максимальной надежностью или пропускной способностью). Позднее маршрутизация стала использоваться как метод балансировки нагрузки в сети, а также для обеспечения требуемого качества обслуживания пользователей. Указанные функции выполняются протоколами маршрутизации, рассмотрению которых и посвящена данная глава [1, 3, 6, 8].

5.1. ПРИНЦИПЫ МАРШРУТИЗАЦИИ В ОБЪЕДИНЕННЫХ СЕТЯХ. КЛАССИФИКАЦИЯ ПРОТОКОЛОВ МАРШРУТИЗАЦИИ

Маршрутизация, как важнейший процесс сетевого уровня подразумевает решение двух задач:

- определение оптимального в определенном смысле пути (маршрута) продвижения пакетов в сети;
- пересылка данных (пакетов) по выбранному пути между двумя конечными узлами в составной сети.

Указанные задачи решаются маршртутизаторами и конечными узлами пакетной сети (рис. 5.1). Каждый порт маршрутизатора можно рассматривать как отдельный узел сети: он имеет собственный сетевой адрес и собственный локальный адрес в той подсети, которая к нему подключена.

Маршрут выбирается на основании имеющейся у маршрутизаторов информации о текущей конфигурации сети, а также на основании указанного критерия выбора маршрута. В маршрутизаторах сети создаются и поддерживаются таблицы маршрутизации, содержащие маршрутную информацию, на основе которой осуществляется выбор пути следования пакетов.

Современные протоколы маршрутизации предусматривают автоматическое формирование таблиц маршрутизации и поддержание их виртуального состояния на основе взаимодействия маршрутизаторов друг с другом.

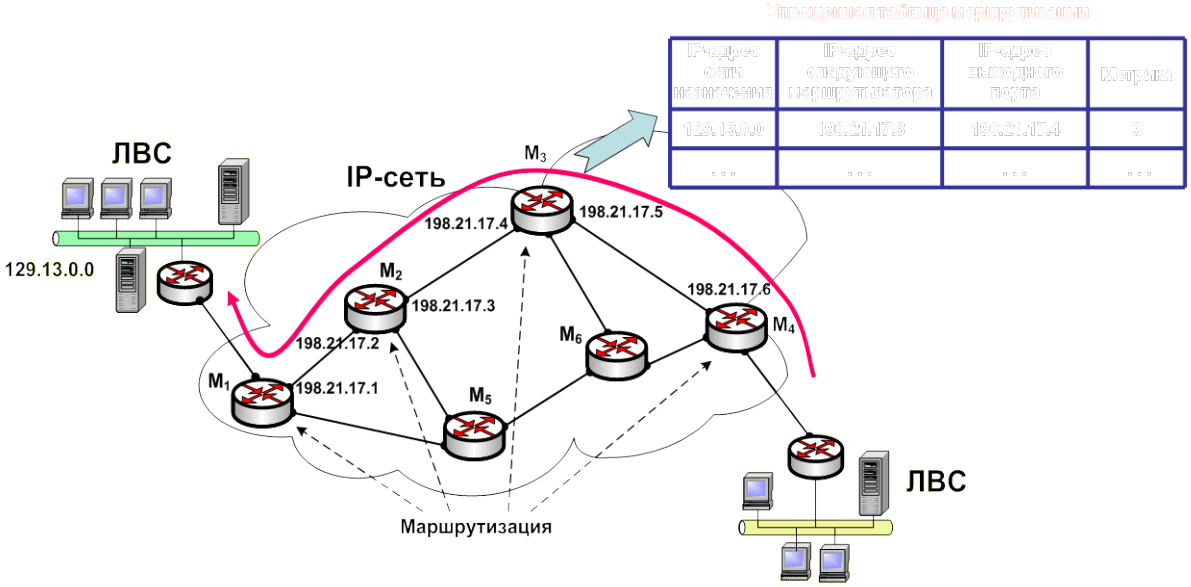


Рис. 5.1. Маршрутизация в IP-сети

Таблица маршрутизации, иногда называемая базой данных маршрутизации, включает набор оптимальных путей, используемых маршрутизатором при передаче пакетов в данный момент времени. Каждая строка этой таблицы содержит, по крайней мере, следующую информацию (рис. 5.1):

- сетевой адрес получателя (адрес сети назначения);
- адрес следующего маршрутизатора, пересылка к которому соответствует оптимальному пути до пункта назначения;
- характеристику пути (метрику) и отметку времени, когда эта характеристика была определена;
- информацию о способе пересылки, например, номер или IP-адрес выходного порта.

В качестве метрики могут выступать задержка прохождения маршрута отдельным пакетом или средняя пропускная способность маршрута. Часто также используется весьма простой показатель, учитывающий только количество промежуточных маршрутизаторов (хопов) в маршруте.

После получения очередного пакета маршрутизатор выполняет следующие действия:

- считывает из заголовка пакета сетевой адрес получателя (IP-адрес);
- по таблице маршрутизации определяется сетевой адрес следующего

транзитного маршрутизатора на пути до пункта назначения (он будет находиться в строке с совпавшим адресом сети назначения);

- заменяет в заголовке пакета свой MAC-адрес, соответствующий канальному уровню модели OSI, на MAC-адрес выбранного транзитного маршрутизатора;

- отсылает пакет выбранному транзитному маршрутизатору.

Следует отметить, что в дейтаграммных сетях выбор маршрута для каждого пакета, как правило, производится заново, так как маршрутная информация могла измениться. Если в сети используется механизм виртуальных каналов, то маршрут выбирается только при создании нового виртуального канала.

В крупных составных сетях объем таблиц маршрутизации может быть очень большим, что замедляет их просмотр. Для уменьшения размеров таблиц маршрутизации может использоваться специальная запись «маршрутизатор по умолчанию» (default). При отсутствии совпадений в ходе просмотра таблицы пакет пересылается по пути, определяемому данной записью. Эффективным средством уменьшения объема таблиц маршрутизации в IP-сетях является также применение технологии бесклассовой междоменной маршрутизации (CIDR), обеспечивающей агрегирование адресов подсетей в составной сети.

Протоколы маршрутизации могут быть построены на основе разных алгоритмов, отличающихся способами построения таблиц маршрутизации, способами выбора наилучшего маршрута и другими особенностями своей работы. Соответственно, протоколы (алгоритмы) маршрутизации могут быть классифицированы по различным признакам. Один из вариантов классификации протоколов маршрутизации представлен на рис. 5.2.

По способу сбора и распространения маршрутной информации различают методы *централизованной* и *распределенной* маршрутизации.

При централизованной маршрутизации сбор информации о топологии и текущем состоянии сети, а также решение задачи выбора наилучших маршрутов осуществляется одним центральным узлом (или несколькими в сети по одному на каждую подсеть). На центральный узел возлагается также задача доставки маршрутной информации каждому узлу сети.

Распределенная маршрутизация характерна для дейтаграммных сетей. При этом способе задачи сбора информации о топологии сети и ее текущем

состоянии, выбора маршрутов решаются каждым узлом сети независимо друг от друга.

Рис. 5.2. Классификация протоколов маршрутизации

Протоколы распределенной маршрутизации относятся к классу *одношаговых*, когда каждый маршрутизатор выбирает один следующий ретрансляционный участок, а весь маршрут складывается из совместной работы всех маршрутизаторов на пути следования пакетов.

В противоположность этому существует и *многошаговый подход* — маршрутизация от источника (Source Routing). В соответствии с ним узел-источник задает в отправляемом в сеть пакете часть или полный маршрут его следования через промежуточные маршрутизаторы. Это ускоряет прохождение пакета по сети, разгружает маршрутизаторы, но при этом большая нагрузка ложится на конечные узлы. Применение такой схемы маршрутизации, наряду с классической одношаговой, предусмотрено в протоколе IPv6.

По способу формирования и обновления таблиц маршрутизации (по степени гибкости) различают:

- алгоритмы статической (фиксированной) маршрутизации;

- алгоритмы динамической (адаптивной) маршрутизации;
- алгоритмы простой маршрутизации.

При фиксированной схеме маршрутизации маршруты между узлами сети являются постоянными. Любые изменения в маршрутные таблицы вносит администратор сети (как правило, только при изменении топологии сети). Алгоритмы статической маршрутизации не учитывают текущих изменений состояния сети, таких, как объем трафика на различных направлениях, характеристики качества обслуживания и т. д. Однако данные алгоритмы достаточно часто используется, прежде всего, в небольших сетях с простой топологией и на магистралях крупных объединенных сетей.

По количеству маршрутов к одной сети различают *одномаршрутные* и *многомаршрутные* алгоритмы. В одномаршрутных таблицах для каждого адресата задан один путь, а в многомаршрутных — несколько альтернативных путей для каждого адресата. В многомаршрутных таблицах должно быть задано правило выбора одного из маршрутов. Чаще всего один путь является основным, а остальные — резервными, с указанием предпочтения перехода на каждый из них.

Самыми распространенными являются алгоритмы динамической маршрутизации. Эти алгоритмы обеспечивают автоматическое обновление таблиц маршрутизации после изменения конфигурации или состояния сети, т. е. при изменении состояния сети могут меняться маршруты следования дейтаграмм. Адаптивные алгоритмы обычно имеют распределенный характер, который выражается в том, что в сети отсутствуют какие-либо выделенные маршрутизаторы, которые собирали бы и обобщали топологическую информацию — эта работа распределена между всеми маршртутизаторами. В таблицах маршрутизации при адаптивной маршрутизации обычно имеется информация об интервале времени, в течение которого данный маршрут будет оставаться действительным. Это время называют временем жизни маршрута (Time To Live, TTL).

Алгоритмы адаптивной маршрутизации в первую очередь реагируют на возникновение таких событий, как неисправность узлов сети или перегрузка в определенной части сети. В обоих случаях пакетный трафик направляется в обход неисправного или перегруженного участка сети. Таким образом, стратегия адаптивной маршрутизации может помочь избежать перегрузки или предотвратить разрастание перегрузки путем балансировки

трафика по различным участкам сети. Как следствие, повышается и производительность сети.

Однако применение адаптивной маршрутизации связано с некоторыми особенностями.

1. Увеличение времени на обработку пакета в маршрутизаторе вследствие усложнения выбора маршрута.

2. Противоречие между качеством (полнотой) информации о состоянии различных частей сети и расходом ресурса на доставку этой информации ко всем узлам сети. Чем больше объем информации, которой обмениваются маршрутизаторы, и чем чаще они ею обмениваются, тем лучше будут решения о выборе маршрутов, принимаемые каждым узлом. Но при этом возрастает поток служебной информации в сети, что снижает ее производительность.

3. Проблема выбора скорости реагирования на изменения состояния сети. Слишком быстрая реакция может привести к перегрузке, а слишком медленная — к устареванию маршрутной информации: обновления маршрутных таблиц не будут успевать за изменениями состояния сети.

4. Применение адаптивной маршрутизации может приводить к таким эффектам как *флаттер* и *зацикливание*.

Флаттером (fluttering) называют явление быстрой циклической смены маршрута, которое может быть вызвано стремлением маршрутизатора распределить или сбалансировать нагрузку. Это приводит к тому, что пакеты могут передаваться по нескольким неравноценным маршрутам (с различными характеристиками) как в одном, так и во встречных направлениях. Это усложняет оценку вышестоящими протоколами и приложениями сетевых характеристик, таких, как время прохождения сигнала в оба конца и пропускная способность. При различии времени прохождения пакета по двум маршрутам TCP-сегменты могут прибывать с нарушением порядка, что порождает дополнительные повторные передачи, дублирование подтверждений и ведет к соответствующим затратам на эти действия ресурсов пропускной способности.

Зацикливание (looping) — это ситуация, при которой пакеты, переданные маршрутизатором, возвращаются на тот же маршрутизатор, и она представляет собой еще более серьезную проблему. Такое явление возникает тогда, когда в объединенной сети происходит изменение связности

(топологической структуры), а информация об этом изменении не успевает распространиться по всем маршрутизаторам.

Несмотря на указанные недостатки, адаптивная маршрутизация преобладает над фиксированной в дейтаграммных сетях в силу обеспечиваемых ею возможностей повышения производительности и борьбы с перегрузками при изменении состояния сети.

Адаптивные протоколы обмена маршрутной информацией, применяемые в настоящее время в вычислительных сетях, в свою очередь, делятся на две группы, каждая из которых связана с одним из следующих типов алгоритмов:

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

- *алгоритмы состояния связей (*Link State Algorithms, LSA*).*

Дистанционно-векторные протоколы маршрутизации базируются на алгоритме Беллмана — Форда, позволяющего вычислить кратчайшие пути от данного узла до всех остальных узлов сети. Дистанционно-векторными они называются потому, что каждый маршрутизатор хранит, периодически (например, каждые 30 с.) обновляет и пересылает соседям вектор расстояний

$$\mathbf{L}_a = \left[L(a,1), L(a,2), ..., L(a,M)\right]^T$$

от данного a-го узла до всех известных ему сетей (M – количество сетей). В свою очередь, он получает аналогичные векторы расстояний от соседних узлов. В качестве метрики расстояния может использоваться время прохождения пакета или количество хопов (ретрансляционных участков или промежуточных маршрутизаторов) до данной сети. После каждого получения вектора расстояния от соседей маршрутизатор, пользуясь принятой информацией и своими данными о расстояниях до непосредственно подключенных к нему сетей и соседних узлов, вычисляет кратчайшие пути до всех сетей. После этого он обновляет свою таблицу маршрутизации и рассылает соседним маршрутизаторам свой обновленный вектор расстояний. Аналогичные процессы происходят в других маршрутизаторах, поэтому информация об обновления распространяется в сети с задержкой — поэтапно от узла к узлу.

Из описанного вытекают следующие особенности дистанционно-

векторных алгоритмов:

- значительный объем передаваемой информации по сети;

- при изменениях в сети может потребоваться значительное время, чтобы информация об этом распространилась по всей объединенной сети.

Дистанционно-векторные алгоритмы хорошо работают только в небольших сетях, В больших сетях они засоряют линии связи интенсивным широковещательным трафиком. К тому же, изменения конфигурации сети могут отрабатываться по этому алгоритму не всегда корректно, так как маршрутизаторы не имеют точного представления о топологии связей в сети, а располагают только оценочной информацией — вектором расстояний, полученной, к тому же, через посредников.

Наиболее распространенным протоколом, основанным на дистанционно-векторном алгоритме, является протокол RIP, который распространен в двух версиях: RIPv1 (не передающий маски и, соответственно, не поддерживающий бесклассовую маршрутизацию) и RIPv2 (поддерживающий маски). Дистанционно-векторный алгоритм служит основой также для протоколов внутреннего шлюза IGRP и EIGRP (улучшенная версия IGRP), разработанных корпорацией Cisco для маршрутизации в больших гетерогенных сетях. Последний (EIGRP) сочетает достоинства дистанционно-векторных алгоритмов и алгоритмов состояния связей и имеет очень быструю сходимость.

Алгоритмы состояния связей (маршрутизация с учетом состояния линий) предполагают построение каждым маршрутизатором точного графа связей сети на основе информации, получаемой от всех остальных маршрутизаторов об их связях с соседними узлами. По этому графу все маршрутизаторы осуществляют выбор кратчайших путей до сетей назначения и заносят их в свои маршрутные таблицы. Для поиска кратчайших путей в протоколах состояния связей используется *алгоритм Дейкстры* (Dijkstra).

Так как все маршрутизаторы работают на основании одинаковых графов, это делает процесс маршрутизации более устойчивым к изменениям конфигурации и состояния сети. Каждый маршрутизатор оценивает состояние своих связей только с непосредственно подключенными к нему узлами и рассылает эту информацию соседним узлам. Те, в свою очередь, проверив актуальность этой информации, пересылают её всем своим соседям, кроме того узла, с которого эта информация была получена. Таким

образом, информация о состоянии связей данного маршрутизатора через соседей рассылается всем маршрутизаторам сети, а не только соседним узлам, как в дистанционно-векторных алгоритмах. Такая «широковещательная» рассылка (передача пакета всем непосредственным соседям маршрутизатора) используется здесь не периодически, а только при изменениях состояния связей, что происходит в надежных сетях не так часто. Да и количество передаваемой информации соседним маршрутизаторам существенно меньше, чем в дистанционно-векторных алгоритмах, так как ограничено лишь информацией о состоянии связей только с соседними узлами.

В качестве оценок состояния связей, в отличие от дистанционно-векторных алгоритмов, кроме времени прохождении пакета могут использоваться различные метрики, в том числе, пропускная способность линий.

В общем виде логику работы протоколов маршрутизации с учетом состояния линий можно представить следующей последовательностью действий каждого маршрутизатора:

1. Обнаружение соседей, непосредственно подключенных к данному маршрутизатору, и определение их сетевых адресов.

2. Оценка характеристик линий связи, подключенных к портам маршрутизатора (измерение времени прохождения сигнала, оценка пропускной способности).

3. Формирование сообщения, содержащего информацию об изменениях состояния своих линий связи до смежных маршрутизаторов, и рассылка их соседям.

4. Получение сообщений от маршрутизаторов сети об изменении состояния их линий связи с соседями, обновление своей базы данных о состояниях линий сети и пересылка полученных сообщений другим маршрутизаторам. После получения информации от всех маршрутизаторов сети будет построен полный граф сети.

5. Вычисление кратчайших путей ко всем узлам.

Чтобы понять, в каком состоянии находятся линии связи, подключенные к его портам, маршрутизатор периодически обменивается короткими пакетами HELLO со своими ближайшими соседями. Этот служебный трафик также засоряет сеть, но не в такой степени как, например, RIP-пакеты, так как пакеты HELLO имеют намного меньший объем.

Примерами протоколов маршрутизации, основанных на алгоритме со-

стояния связей, являются протоколы IS-IS (Intermediate System to Intermediate System) стека OSI, OSPF (Open Shortest Path First) стека TCP/IP.

В табл. 5.1 приведено сравнение алгоритмов дистанционно-векторной маршрутизации и маршрутизации с учетом состояния линий.

Таблица 5.1

Сравнение алгоритмов маршрутизации

Дистанционно-векторная маршрутизация	Маршрутизация с учетом состояния линий
1. Каждый маршрутизатор посылает сведения о маршрутах своим соседям.	1. Каждый маршрутизатор посылает сведения о маршрутах всем остальным узлам.
2. Посылаемая информация представляет собой оценку путей ко всем сетям.	2. Посылаемая информация представляет собой точное значение метрик линий к смежным узлам.
3. Сообщения посылаются регулярно через определенный период времени.	3. Сообщения посылаются в случае значительных изменений состояния линий.
4. Маршрутизатор выбирает следующий ретрансляционный участок с помощью распределенного алгоритма Беллмана — Форда.	4. Маршрутизатор строит полный граф сети, а затем выбирает кратчайшие пути с помощью любого алгоритма (к примеру, Дейкстры).
Сравнение характеристик алгоритмов	
1. Значительный объем передаваемой информации. 2. При изменениях состояния сети может потребоваться значительное время для распространения информации по всей сети. 3. Объем расчетов в каждом узле меньше.	1. Объем передаваемой информации существенно меньше. 2. Меньшее время распространения информации по всей сети. 3. Большой объем расчетов в каждом узле.

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 попавшем случайном направлении, кроме исходного;

- лавинная маршрутизация, когда пакет широковещательно посылается по всем возможным направлениям, кроме исходного (аналогично обработке мостами кадров с неизвестным адресом);

- маршрутизация по предыдущему опыту, когда выбор маршрута осуществляется по таблице, но таблица строится по принципу моста путем анализа адресных полей пакетов, появляющихся на входных портах.

Метод лавинной маршрутизации обладает тремя замечательными свойствами:

1. Перебираются все маршруты между отправителем и получателем, поэтому пакет всегда будет доставлен, если сохранился хотя бы один путь до адресата-назначения. Благодаря этому метод лавинной маршрутизации характеризуется высокой устойчивостью.

2. Поскольку перебираются все маршруты, по меньшей мере, одна копия пакета пройдет по маршруту с минимальной задержкой. Поэтому рассылаемая маршрутная информация быстро достигает всех маршрутизаторов.

3. Посещаются все узлы, прямо или косвенно соединенные с источником. Благодаря этому все маршрутизаторы получают информацию, необходимую для создания таблиц маршрутизации.

Главный недостаток метода лавинной маршрутизации заключается в оказываемом им на сеть высоком уровне нагрузки, пропорциональном количеству линий в сети.

По области функционирования (влияния) протоколы маршрутизации делятся на *внутридоменные* (или протоколы внутренних шлюзов — Interior Gateway Protocol, IGP) и *междоменные* (или протоколы внешних шлюзов — Exterior Gateway Protocol, EGP). Их еще называют протоколами внутренней и внешней маршрутизации соответственно. В глобальной интернет-сети протоколы внутренней маршрутизации работают внутри автономных систем, а область применения протоколов внешней маршрутиза-

ции — это магистральная сеть, связывающая автономные системы между собой (рис. 5.3). Примером протокола внешних шлюзов является протокол BGP.

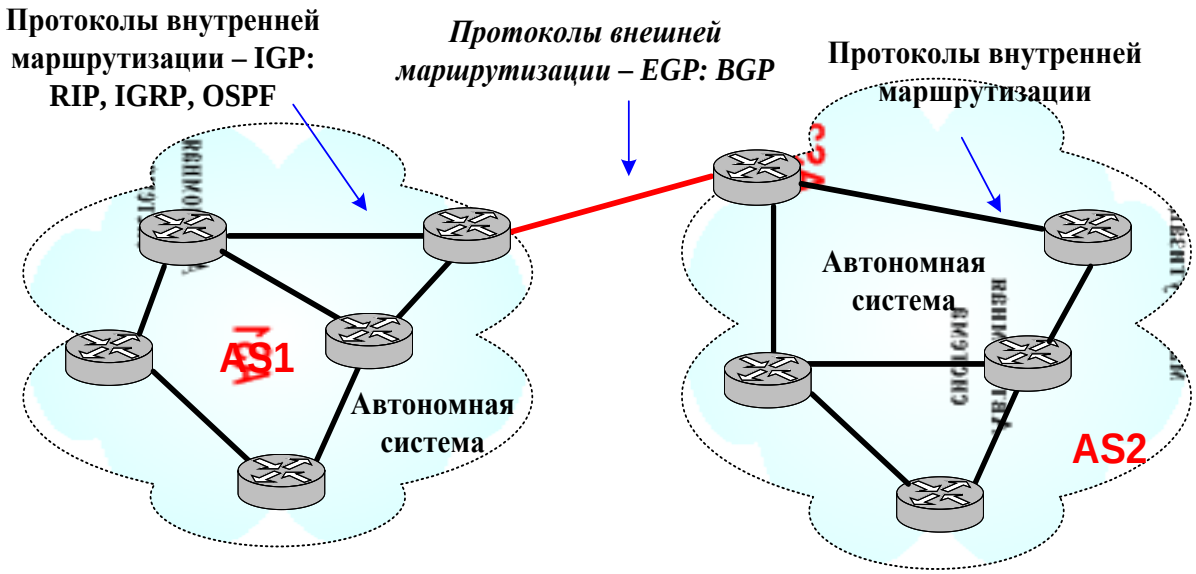


Рис. 5.3. Области применения протоколов внутренней и внешней маршрутизации

Напомним, что автономной системой называется сеть или группа сетей, находящихся под единым административным управлением и использующая единую политику маршрутизации.

Для сокращения объема таблиц маршрутизации в крупных сетях применяют иерархическую (многоуровневую) маршрутизацию. В отличие от одноуровневой маршрутизации при использовании иерархической маршрутизации маршрутизаторы сети разбиваются на отдельные области. Каждый маршрутизатор знает все детали выбора маршрутов в пределах своей области (подсети), но ему ничего не известно о внутреннем строении (топологии и состоянии связей) других областей.

На нижнем уровне иерархии с помощью протоколов внутренней маршрутизации осуществляется выбор маршрута до конечного или выходного узла внутри подсети (области) как последовательности маршрутизаторов на пути следования пакетов. На втором уровне иерархии осуществляется выбор маршрута как последовательности подсетей на пути до подсети-

назначения. При этом подсети (области) рассматриваются как неделимые элементы (узлы) объединенной сети. В глобальной сети интернет на третьем уровне иерархии маршрут будет определяться с помощью протоколов внешней маршрутизации как последовательность автономных систем.

В очень больших сетях области могут объединяться в кластеры, кластеры в зоны, зоны в группы и т. д.

Необходимо отметить, что при применении иерархической маршрутизации кратчайший маршрут может быть строго найден только в пределах области (подсети) на нижнем уровне иерархии. Выбор оптимального пути для составного маршрута (через последовательность областей, зон и т. п.) верхних уровней иерархии не гарантируется. Это является платой за сокращение объема таблиц маршрутизации.

5.2. АЛГОРИТМЫ МАРШРУТИЗАЦИИ

Задача определения оптимального по заданному критерию маршрута (кратчайшего пути) решается с применением алгоритмов маршрутизации, реализуемых протоколами маршрутизации. Вообще говоря, задача поиска кратчайшего пути формализуется как задача линейного программирования. Однако для решения данной задачи разработаны специальные алгоритмы. В частности, широкое применение находят уже упоминавшиеся алгоритмы Дейкстры и Беллмана — Форда, которые позволяют вычислить кратчайшие пути от заданного узла до всех узлов сети. В качестве метрики при определении кратчайшего пути, как указывалось выше, могут выступать задержка, пропускная способность и др. В результате работы алгоритма маршрутизации в маршрутизаторе создается таблица маршрутизации, в которой хранятся маршрутные записи, содержащие информацию о маршрутах с указанием соответствующих им метрик.

5.2.1. АЛГОРИТМ ДЕЙКСТРЫ

Алгоритм, предложенный нидерландским ученым Э. Дейкстрой в 1959 г., позволяет находить кратчайшие пути от одной из вершин графа до всех остальных. Алгоритм работает только для графов, не имеющих ребер с отрицательной метрикой (весом).

Работа алгоритма происходит поэтапно путем последовательной обработки всех вершин графа, начиная с вершины-источника и вычисления при

этом кратчайших расстояний до k узлов за k шагов. В процессе выполнения алгоритма при переходе от узла i к узлу j используется специальная процедура пометки вершин графа. Метки бывают *временные* и *постоянные* и представляют собой метрику пути до данной вершины. Если узлу присвоена постоянная метрика, то это означает что данный путь до узла от вершины-источника — кратчайший. После того, как все вершины графа обработаны, и все метки стали постоянными, считается, что все кратчайшие пути от узла-источника до остальных узлов найдены, и алгоритм прекращает свою работу.

Пусть $G(V,E)$ — граф, описывающий моделируемую сеть, характеризуемую множеством узлов V и множеством ребер E; T — множество вершин графа, уже обработанных алгоритмом; c_{ij} — длина ребра между i-м (υ_i) и j-м (υ_j) смежными узлами ($c_{ij} \geq 0$). Метка j-го узла (u_j, i) определяется как

$$(u_j, i) = (u_i + c_{ij}, i),$$

где u_i — кратчайшее расстояние от узла-источника до i-го узла.

Формально алгоритм представляется в виде следующей последовательности этапов (2-й этап повторяется до окончания работы алгоритма).

1. Первый этап — инициализация

Исходное множество узлов состоит из одной вершины-источника $T = \{\upsilon_i\}$. Метка самой вершины υ_i полагается постоянной и равной $(0, -)$, метки остальных вершин — временные и равные бесконечности $(\infty, -)$. Это отражает то, что метрики от υ_i до других вершин пока неизвестны. Все вершины графа помечаются как необработанные.

2. Второй этап

2.1. Вычисляются временные метки

$$(u_j, i) = (u_i + c_{ij}, i)$$

для всех узлов (соседей), которых можно достичь непосредственно из вершины υ_i и которые не имеют постоянных меток. Далее, если узел υ_j уже имеет временную метку (u_j, k), полученную от узла υ_k, то при $u_i + c_{ij} < u_j$ (т. е. когда значение новой метки меньше, чем старой) метка (u_j, k) заменяется на (u_j, i). В противном случае, когда $u_i + c_{ij} > u_j$, метка

не меняется.

2.2. Узел υ_i отмечается как обработанный и добавляется во множество T, а его метка становится постоянной.

а). Если все узлы имеют постоянные метки, то процесс вычислений заканчивается. Или, что то же самое, когда $T = V$, т. е. когда все вершины графа обработаны и попали во множество T.

б). В противном случае из ещё не обработанных узлов (не имеющих постоянной метки) выбирается тот, который имеет минимальное значение u_r временной метки (u_r, s). Напомним, что u_r — это метрика пути от узла-источника до r-го узла через предпоследний узел υ_s на этом пути. Далее полагаем $i = r$ и возвращаемся ко второму шагу.

Рассмотрим работу алгоритма Дейкстры на примере сети на рис. 5.5. Здесь $V = \{\upsilon_1, \upsilon_2, ..., \upsilon_6\}$. Будем искать кратчайшие пути из узла 1 ко всем остальным узлам сети. В кружках обозначены номера вершин, над ребрами обозначена их «стоимость» — метрика ребра. Рядом с каждой вершиной в круглых скобках обозначается временная метка, а в квадратных — постоянная (длина кратчайшего пути в эту вершину из вершины 1). Пунктиром обведен тот узел, который в данный момент выбран для обработки. Затенены уже обработанные вершины графа.

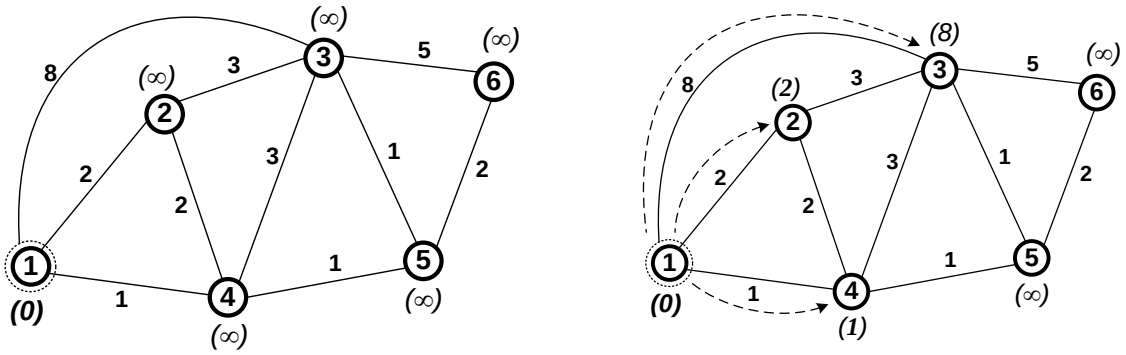


Рис. 5.4. Инициализация (*а*) и вычисление меток пути (*б*) в алгоритме Дейкстры

1. Первый этап

При инициализации (рис. 5.4, *а*) 1-й узел υ_1 отмечается как обрабатываемый, ему присваивается метка $(0, -)$, остальным узлам — метки $(\infty, -)$.

2. **Второй этап**

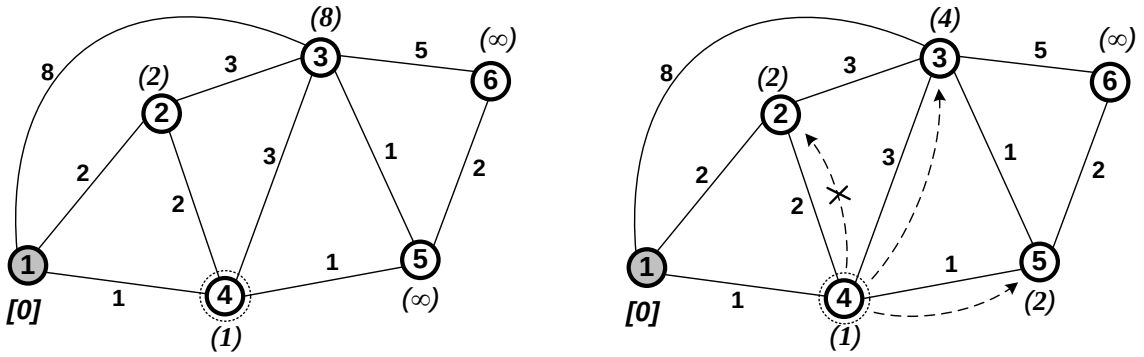
2.1. Определяем временные метки смежных с υ_1 узлов υ_2, υ_3 и υ_4. Они вычисляются как сумма значения метки 1-го узла и метрики ребра между υ_1 и соответствующим соседним узлом (рис. 5.4, *б*):

$$\text{для } \upsilon_2 \ \ (u_2, 1) = (0 + 2, 1) = (2, 1);$$

$$\text{для } \upsilon_3 \ \ (u_3, 1) = (0 + 8, 1) = (8, 1);$$

$$\text{для } \upsilon_4 \ \ (u_4, 1) = (0 + 1, 1) = (1, 1).$$

Так как все рассматриваемые узлы до этого имели значения меток, равные бесконечности, их метки заменяются на вычисленные $(u_2, 1)$, $(u_3, 1)$ и $(u_4, 1)$.



в) *г*)

Рис. 5.4. Второй (*в*) и третий (*г*) шаги в алгоритме Дейкстры

2.2. 1-й узел υ_1 отмечается как обработанный (эта вершина на рис. 5.4, *в* затенена), а его метка приобретает постоянный статус (в квадратных скобках). В качестве следующего обрабатываемого узла выбирается 4-й узел υ_4, так как он имеет минимальную из оставшихся узлов временную метку (метрику пути из 1-го узла) $u_4 = 1$. Так как не всем узлам присвоены постоянные метки, на следующем шаге возвращаемся к началу второго этапа.

3. Второй этап (повторяется)

3.1. Определяем временные метки смежных с υ_4 узлов υ_2, υ_3 и υ_5 (рис. 5.4, *г*). Они вычисляются как сумма значения метки 4-го узла и мет-

рики ребра между υ_4 и соответствующим соседним узлом:

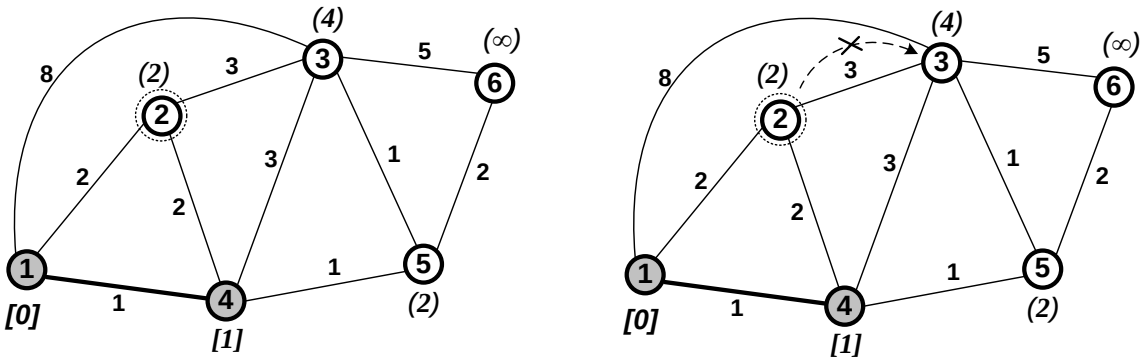
$$\text{для } \upsilon_2 \ \ (u_2, 4) = (1+2, 1) = (3, 1);$$

$$\text{для } \upsilon_3 \ \ (u_3, 4) = (1+3, 1) = (4, 1);$$

$$\text{для } \upsilon_5 \ \ (u_5, 4) = (1+1, 1) = (2, 1).$$

Значение текущей временной метки 2-го узла $(u_2, 1) = (2, 1)$ меньше вычисленной $(u_2, 4) = (3, 1)$, поэтому она не заменяется. Вычисленные значения меток для 3-го и 5-го узлов меньше текущих, поэтому метки этих узлов меняются на новые: $(u_3, 4) = (4, 1)$ и $(u_5, 4) = (2, 1)$.

3.2. 4-й узел υ_4 отмечается как обработанный (рис. 5.4д), а его метка приобретает постоянный статус (в квадратных скобках). Это означает, что найден кратчайший путь из 1-го узла во 4-й (это ребро e_{14}, оно выделено утолщенной линией). В качестве следующего обрабатываемого узла выбирается 2-й узел υ_2, так как он имеет минимальную из оставшихся узлов временную метку (метрику пути из 1-го узла) $u_2 = 2$. Так как не всем узлам присвоены постоянные метки, то на четвертом шаге возвращаемся в начало второго этапа.



д) *е*)

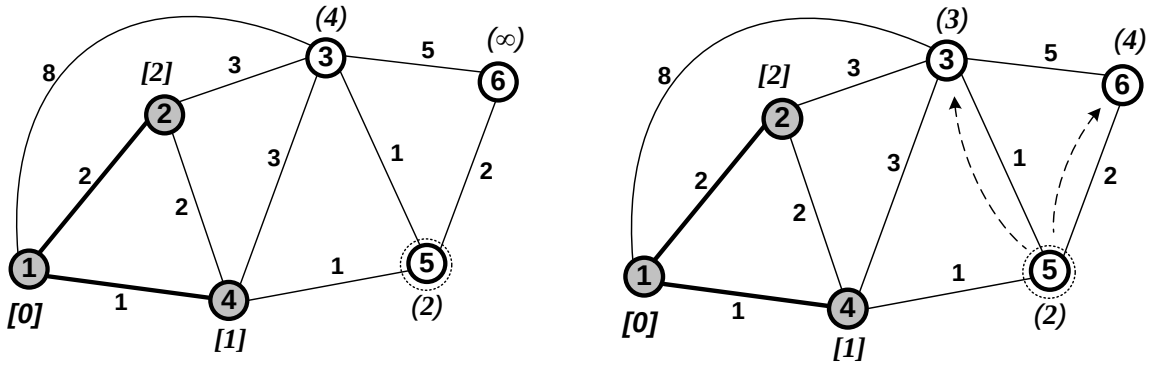
Рис. 5.4. Третий (*д*) и четвёртый (*е*) шаги в алгоритме Дейкстры

4. Второй этап (повторяется)

4.1. Определяем временные метки соседних с υ_2 узлов, которым ещё не присвоены постоянные метки (рис. 5.4, *е*). Это 3-й узел υ_3: $(u_3, 2) = (2+3, 1) = (5, 1)$. Значение вычисленной метки $(u_3, 2)$ больше значения текущей метки 3-го узла, поэтому замена метки не осуществляет-

ся: $(u_3, 4) = (4, 1)$.

4.2. 2-й узел υ_2 отмечается как обработанный (рис. 5.4, *ж*), а его метка становится постоянной. Это означает, что найден кратчайший путь из 1-го узла во 2-й (это ребро e_{12}, оно выделено утолщённой линией). В качестве следующего обрабатываемого узла выбирается 5-й узел υ_5, так как он имеет минимальную из оставшихся узлов временную метку (метрику пути из 1-го узла) $u_5 = 2$. Так как не всем узлам присвоены постоянные метки, на пятом шаге возвращаемся ко второму этапу.



ж) *з*)

Рис. 5.4. Четвёртый (*ж*) и пятый (*з*) шаги в алгоритме Дейкстры

5. Второй этап (повторяется)

5.1. Определяем временные метки соседних с υ_5 узлов, которым ещё не присвоены постоянные метки (рис. 5.4, *з*). Это 3-й и 6-й узлы:

$$\text{для } \upsilon_3\colon (u_3, 5) = (2+1, 1) = (3, 1);$$

$$\text{для } \upsilon_6\colon (u_6, 5) = (2+2, 1) = (4, 1).$$

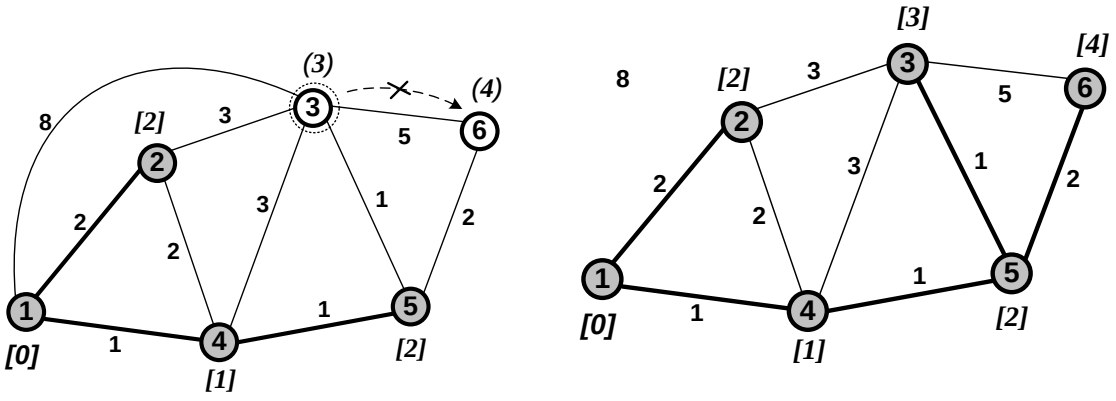
Вычисленные значения меток для 3-го и 6-го узлов меньше текущих. Поэтому метки этих узлов меняются на новые: $(u_3, 5) = (3, 1)$ и $(u_6, 5) = (4, 1)$.

5.2. 5-й узел υ_5 отмечается как обработанный (рис. 5.4, *и*), а его метка становится постоянной. Это означает, что найден кратчайший путь из 1-го узла во 5-й (это путь $\upsilon_1 - \upsilon_4 - \upsilon_5$, включающий ребра e_{14} и e_{45}). В качестве следующего обрабатываемого узла выбирается 3-й узел υ_3, так как из оставшихся не обработанными узлов υ_3 и υ_6 он имеет минимальную вре-

менную метку: $(u_3, 5) = (3, 1)$, значение метки – $u_3 = 3$. Так как не всем узлам присвоены постоянные метки, на шестом шаге возвращаемся ко второму этапу.

6. Второй этап (повторяется)

6.1. Определяем временные метки соседних с υ_3 узлов, которым ещё не присвоены постоянные метки (также иллюстрируется на рис. 5.4, *и*). Это 6-й узел υ_6: $(u_6, 3) = (3 + 5, 1) = (8, 1)$. Значение вычисленной метки $(u_6, 3)$ больше значения текущей метки 3-го узла, поэтому замена метки не осуществляется: $(u_6, 5) = (4, 5)$.



и) *к*)

Рис. 5.4. Пятый (*и*) и шестой (*к*) шаги в алгоритме Дейкстры

6.2. 3-й узел υ_3 отмечается как обработанный (рис. 5.4, *к*), а его метка становится постоянной. Это означает, что найден кратчайший путь из 1-го узла во 3-й (это путь $\upsilon_1 - \upsilon_4 - \upsilon_5 - \upsilon_3$, включающий ребра e_{14}, e_{45} и e_{53}). В качестве следующего обрабатываемого узла выбирается один не обработанный 6-й узел υ_6. Так как у него нет соседей с временными метками, статус его временной метки меняется на постоянный (рис. 5.4, *к*). Кратчайшим путем 1-го узла в 6-й является путь $\upsilon_1 - \upsilon_4 - \upsilon_5 - \upsilon_6$. Все вершины графа обработаны, и на этом алгоритм заканчивает работу.

В рассмотренном примере за шесть шагов алгоритм Дейкстры нашел все кратчайшие пути от 1-го узла до остальных пяти узлов.

Приведенный выше пример работы алгоритма Дейкстры можно представить в табличной форме (табл. 5.2). Жирным шрифтом выделены кратчайшие пути, находимые алгоритмом на каждом шаге работы. Значе-

ния постоянных меток узлов (метрик путей до этих узлов), как и ранее, обозначены в квадратных скобках. Из таблицы видно, что на каждом шаге, начиная со второго, множество T обработанных узлов увеличивается на один узел, и, соответственно, находится один кратчайший путь до какого-либо узла.

Таблица 5.2

Пример пошаговой работы алгоритма Дейкстры

Шаг	Множество T	Метрика пути/маршрут от узла υ_1 до узлов									
		υ_2		υ_3		υ_4		υ_5		υ_6	
1	$\{\upsilon_1\}$	2	1-2	8	1-3	1	1-4	—	—	—	—
2	$\{\upsilon_1,\upsilon_4\}$	2	1-2	4	1-4-3	[1]	**1-4**	2	1-4-5	—	—
3	$\{\upsilon_1,\upsilon_4,\upsilon_2\}$	[2]	**1-2**	4	1-4-3	[1]	**1-4**	2	1-4-5	—	—
4	$\{\upsilon_1,\upsilon_4,\upsilon_2,\upsilon_5\}$	[2]	**1-2**	3	1-4-5-3	[1]	**1-4**	[2]	**1-4-5**	4	1-4-5-6
5	$\{\upsilon_1,\upsilon_4,\upsilon_2,\upsilon_5,\upsilon_3\}$	[2]	**1-2**	[3]	**1-4-5-3**	[1]	**1-4**	[2]	**1-4-5**	4	1-4-5-6
6	$\{\upsilon_1,\upsilon_2,\upsilon_3,\upsilon_4,\upsilon_5,\upsilon_6\}$	[2]	**1-2**	[3]	**1-4-5-3**	[1]	**1-4**	[2]	**1-4-5**	[4]	**1-4-5-6**

Время работы алгоритма Дейкстры пропорционально $|V|^2$, так как алгоритм выполняет $|V|-1$ итераций, а количество операций, выполняемых на каждой итерации, пропорционально $|V|$.

5.2.2. АЛГОРИТМ БЕЛЛМАНА — ФОРДА

Алгоритм Беллмана — Форда был использован в 1969 г. в протоколе маршрутизации RIP сети ARPANET. Он, как и алгоритм Дейкстры, находит кратчайшие пути от одной вершины графа до всех остальных, но в отличие от последнего алгоритм Беллмана-Форда допускает наличие в графе рёбер с отрицательным весом.

Этот алгоритм также работает поэтапно. Но идея его состоит в том, что сначала находятся кратчайшие пути от заданной вершины, состоящие из одного ребра, затем находятся кратчайшие пути при условии, что они

состоят максиму из двух рёбер, затем — максимум из трёх и т. д.

Пусть по-прежнему $G(V,E)$ — граф, описывающий моделируемую сеть; $c_{ij} = c(i, j)$ — метрика ребра между i-м (υ_i) и j-м (υ_j) смежными узлами ($c_{ij} = \infty$, если вершины υ_i и υ_j не соединены напрямую); h — максимальное количество рёбер в пути на текущем шаге алгоритма; $L_h(i)$ — минимальная метрика пути от узла-источника s до узла υ_i.

Алгоритм состоит из двух этапов. Этап 2 повторяется до тех пор, пока метрики путей не перестанут изменяться, но для значений h не более $h \le |V| - 1$, поскольку кратчайший путь не может содержать большее число ребер, иначе он будет содержать цикл, который точно можно выкинуть.

1. Этап 1 — инициализация

$L_0(i) = \infty$ для всех $i \ne s$; $L_h(s) = 0$ для всех h.

2. Этап 2.

Вычисление для каждого узла j метрик минимальных путей до узла-источника s, включающих не более h рёбер (начиная с $h = 0$ и далее при повторении 2-го этапа: $h = 1, 2, ...$):

$$L_{h+1}(\mathrm{j}) = \min_k \{L_h(k) + c(k, j\} . \qquad (5.1)$$

Согласно (5.1) находится проходящий через соседний с узлом j узел k путь, состоящий из $h+1$ рёбер и имеющий минимальную метрику среди всех возможных соседних узлов k, пути $L_h(k)$ до которых от вершины-источника s были найдены на предыдущей итерации. При $h = k$ для каждой вершины j алгоритм сравнивает путь от вершины-источника s до вершины j длиной $k+1$ с путем, существовавшим к концу предыдущей итерации. Если предыдущий путь имеет меньшую метрику, то он сохраняется. В противном случае сохраняется новый путь от вершины-источника s до узла j длиной $k+1$.

В табл. 5.3 и на рис. 5.5 показано применение этого алгоритма для поиска кратчайших путей от 1-го узла υ_1 до всех остальных узлов графа.

1. Первый этап

При инициализации метрики путей до всех узлов от вершины-источника (от 1-го узла) полагаются равными бесконечности:

$$L_0(i) = \infty,\ i \ne 1,\ i = \overline{2, |V|} .$$

2. Второй этап

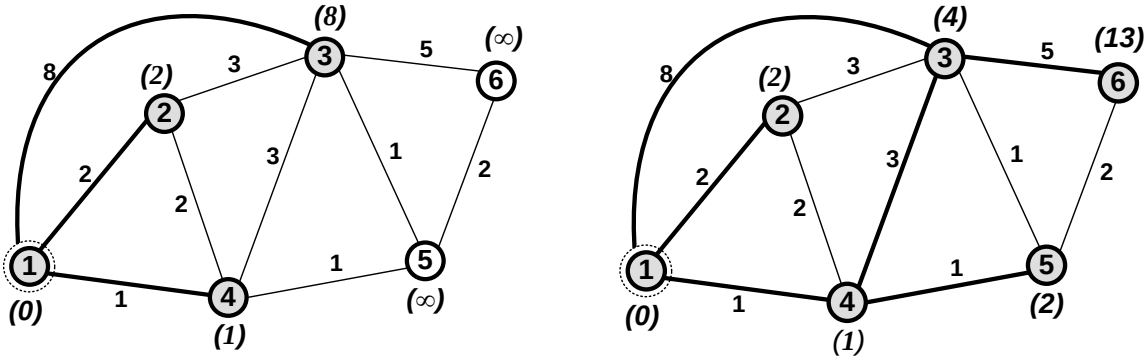
При первой итерации вычисляются метрики путей, состоящих из одного ребра (при $h=1$), от 1-го узла до смежных узлов. Результат показан на рис. 5.5, *а* и в соответствующей строке табл. 5.3.

Таблица 5.3

Пример пошаговой работы алгоритма Беллмана — Форда

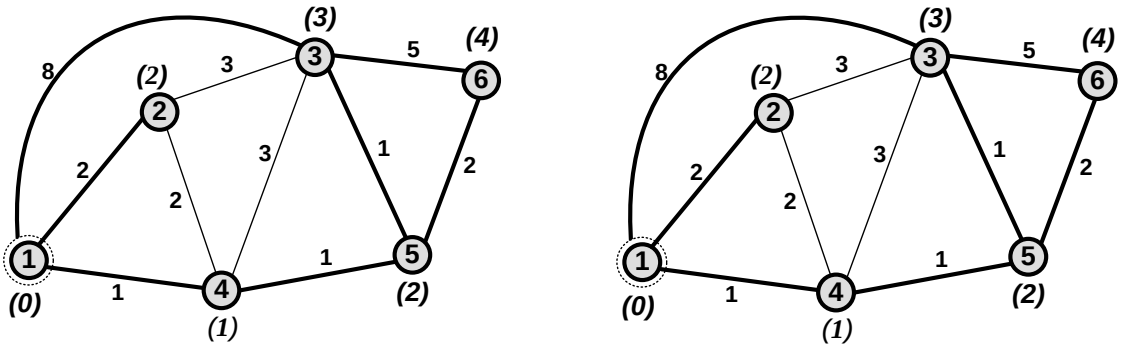
h	$L_h(2)$	Путь	$L_h(3)$	Путь	$L_h(4)$	Путь	$L_h(5)$	Путь	$L_h(6)$	Путь
0	∞	—	∞	—	∞	—	∞	—	∞	—
1	2	1–2	8	1–3	1	1–4	∞	—	∞	—
2	2	1–2	4	1–4–3	1	1–4	2	1–4–5	10	1–3–6
3	2	1–2	3	1–4–5–3	1	1–4	2	1–4–5	4	1–4–5–6
4	2	1–2	3	1–4–5–3	1	1–4	2	1–4–5	4	1–4–5–6

На рис. 5.5, *а* в круглых скобках рядом с вершинами графа записаны значения метрик минимальных путей на каждом шаге от 1-го узла до данного узла.



а) $h = 1$ *б*) $h = 2$

Рис. 5.5. Пошаговая работа алгоритма Беллмана — Форда



в) $h = 3$ *г*) $h = 4$

Рис. 5.5. Пошаговая работа алгоритма Беллмана — Форда

Далее этап 2 повторяется для значений $h = 2$, 3 и 4. На каждом шаге алгоритм находит путь с минимальной метрикой, максимальное число ретрансляционных участков в котором не превышает h. Результаты после каждой итерации показаны на рис. 5.5, *б–г* и в соответствующих строках табл. 5.3. В конечном итоге определяются кратчайшие пути от 1-го узла до всех остальных (последняя строка табл. 5.3), и построено так называемое *остовное дерево* (рис. 5.5, *г*).

Время работы алгоритма Беллмана-Форда пропорционально $|V| \times |E|$, так как алгоритм выполняет $|V| - 1$ итераций, а каждая итерация включает определение веса каждого ребра.

5.3. ПРОТОКОЛ ВНУТРЕННЕЙ МАРШРУТИЗАЦИИ RIP

Протокол RIP (Routing Information Protocol — протокол маршрутной информации) представляет собой относительно простой протокол внутренней маршрутизации дистанционно-векторного типа. Это один из наиболее ранних протоколов маршрутизации, он вполне пригоден для небольших объединенных сетей и остается одним из наиболее распространенных маршрутных ввиду простоты реализации.

Как уже было сказано, для межсетевого протокола IP имеются две версии RIP: RIPvl, определенный в RFC 1058-1988 г., и RIPv2, описанный в RFC 2453-1998 г. Протокол RIPvl не поддерживает масок; протокол RIPv2 передает информацию о масках сетей, поэтому он в большей степени соответствует требованиям сегодняшнего дня.

Протокол для поиска кратчайших путей использует алгоритм Беллмана — Форда, для работы которого необходим обмен маршрутной информацией между соседними узлами. Два узла называются соседними, если они напрямую соединены с одной и той же сетью (маршрутизаторы A и B, B и C, B и C, B и D на рис. 5.6).

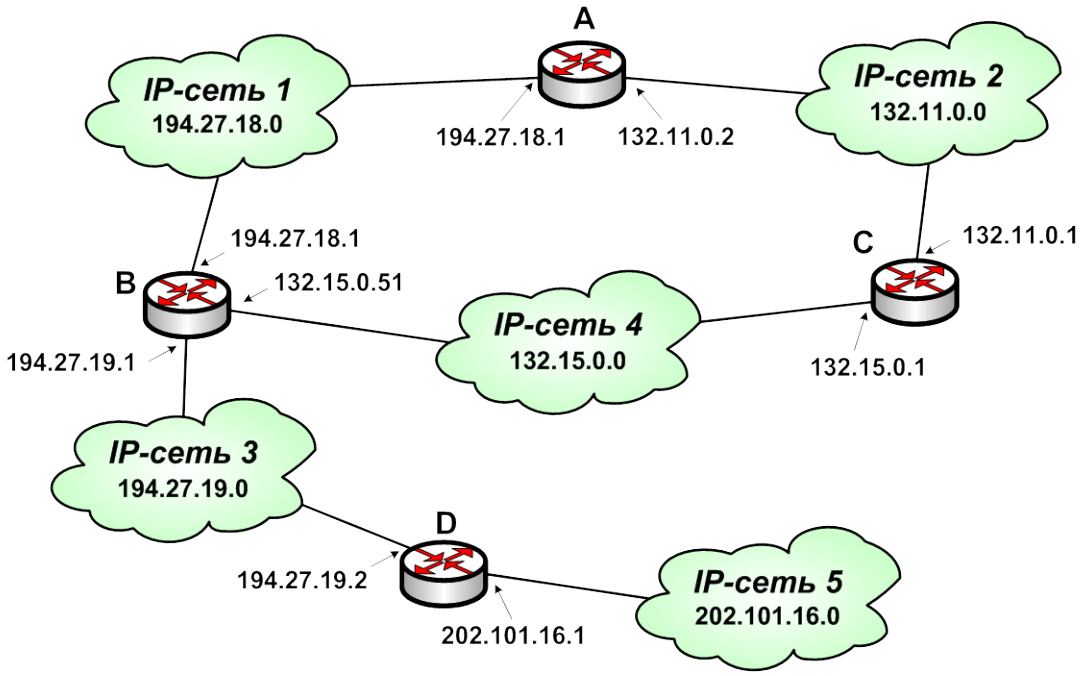


Рис. 5.6. Пример сети на RIP-маршрутизаторах

В каждом RIP-маршрутизаторе поддерживается таблица маршрутизации, в которой для каждого адресата сети RIP хранится не более, чем по одной записи, содержащей следующие сведения (табл. 5.4):

- IP-адрес сети назначения (или хоста);

- метрика, которая представляет общую стоимость доставки дейтаграммы от маршрутизатора к адресату (эта метрика представляет собой сумму стоимостей, связанных с каждой сетью на пути к адресату);

- IP-адрес следующего маршрутизатора на пути к получателю (next hop); если адресат находится в подключенной напрямую сети, этот параметр не требуется;

- интерфейс (номер порта) — физический интерфейс, используемый для связи со следующим маршрутизатором.

Таблица 5.4

Информация, хранящаяся в RIP- маршрутизаторе (А)

Адрес сети	Адрес следующего маршрутизатора	Порт	Метрика
195.27.18.0	195.27.18.1	1	1
132.11.0.0	132.11.0.2	2	1
132.15.0.0	132.11.0.1	2	2
195.27.19.0	195.27.18.1	3	2
202.101.16.0	195.27.18.1	3	3

Кроме того, маршрутные записи могут содержать такие сведения о маршруте (в табл. 5.4 не показаны), как:

- флаг, показывающий, что информация о маршруте была недавно изменена — его называют флагом изменения маршрута (route change flag);

- различные таймеры, связанные с маршрутом (к примеру, время, прошедшее с момента последнего обновления маршрута); применяемые таймеры будут описаны ниже.

Наиболее важным параметром является метрика или «стоимость» пути. RIP-метрика сети выражается целым числом от 1 до 15 включительно. В качестве метрики обычно используется количество ретрансляционных участков (хопов). Однако могут использоваться и другие параметры: пропускная способность, надежность, выражаемые опять-таки числами от 1 до 15. В случае недоступности сети ей присваивается метрика 16 (бесконечность). Метрика для непосредственно подключенной сети обычно устанавливается равной 1. В новых разработках администратору должна предоставляться возможность установки метрики для каждой сети.

Сообщения протокола RIP

Для передачи сообщений RIP-маршрутизаторы используют транспортный протокол UDP и специально выделенный для протоколов RIP-1 и RIP-2 порт UDP-520.

Формат RIP-пакета показан на рис. 5.7.

Каждое сообщение содержит заголовок RIP, включающий идентификатор команды и номер версии.

Поле *Команда* идентифицирует тип сообщения и может принимать два значения:

1 – запрос (request) на получение таблицы маршрутизации или её части;

2 – отклик (response) или сообщение обновления, содержит запрошенную таблицу маршрутизации или ее часть; это сообщение может быть передано в ответ на запрос или как не запрошенное (unsolicited) обновление таблицы маршрутизации.

Рис. 5.7. Формат RIP-пакета

В поле «Версия» указывается версия протокола RIP: 1 или 2.

Оставшаяся часть дейтаграммы содержит маршрутные записи (Route Entry, RTE) от 1 до 25, каждая из которых (длиной 20 байт) для RIP-2 включает следующие поля:

- идентификатор семейства адресов (AFI);
- метка маршрута (Route Tag или RT)*;
- адрес сети (хоста) назначения;
- маска подсети*;
- адрес следующего маршрутизатора*;
- метрика пути.

Поля, помеченные символом «*», для RIP-1должны иметь нулевое значение.

Поле AFI указывает тип адреса (к примеру, IP-адрес) и для RIP-1 может принимать единственное значение AF_INET = 2 (адрес IPv4).

Поле *Метка маршрута* является атрибутом маршрута и позволяет отличать «внутренние» RIP-маршруты (маршруты для сетей внутри области RIP-маршрутизации) от «внешних», которые импортируются от EGP или других IGP. Например, маршруты, импортируемые от протокола внешней маршрутизации BGP, в этом поле могут содержать номер автономной системы. Возможны и другие варианты использования этого поля в целях обеспечения взаимодействия RIP и BGP.

Поле *Маска подсети* (Subnet Mask) содержит маску подсети, которая применяется к IP-адресу для того, чтобы выделить из него номер сети (подсети). Если запись не включает маску подсети, то поле должно иметь нулевое значение.

Поле Следующий маршрутизатор (Next Hop) позволяет избавиться от лишней пересылки пакетов в системе. Особенно полезна такая возможность в тех случаях, когда протокол RIP поддерживается не всеми маршртутизаторами в сети. Данное поле является информационным (advisory), т. е. его можно игнорировать (это может привести к снижению производительности, но маршрутизация будет работать нормально).

Поле метрики, как указывалось ранее, может содержать значения от 1 до 15 (включительно), или значение 16 (бесконечность), говорящее о недоступности адресата.

Запросы

Запросы используются для получения от соседних маршрутизаторов сообщений, содержащих полную таблицу маршрутизации или её часть.

Запросы на передачу полной таблицы маршрутизации обычно передаются в широковещательном режиме (групповая адресация в RIP-2) RIP-маршртутизаторами, которые были недавно инициализированы и хотят как можно быстрее заполнить свою таблицу маршрутизации. Такой запрос содержит единственную запись, идентификатор семейства адресов в нем имеет нулевое значение, а метрика бесконечна (16). Отклик с полной таблицей маршрутизации передается с использованием адреса и порта из принятого запроса.

Могут возникать ситуации, когда требуется получить таблицу маршрутизации от одного маршрутизатора (например, мониторинг маршрутизатора). Тогда запрос адресуется напрямую конкретному маршрутизатору через порт UDP, отличный от порта, закрепленного за RIP. Такие запросы

обрабатываются последовательно — запись за записью. А именно, последовательно просматривается список RTE (маршрутных записей) в запросе, и для каждой записи находится адресат в базе данных маршрутизатора, получившего запрос. Если искомый маршрут найден, то его метрика помещается в соответствующее поле RTE . Если по искомому адресу нет явного маршрута, то в поле метрики указывается бесконечное значение (16). После просмотра всех записей значение поля команды изменяется на Response (отклик), и дейтаграмма возвращается отправителю запроса. Таким образом, в отклике на запрос возвращается не полная таблица маршрутизации, а обработанные (обновленные) отдельные записи, содержавшиеся в запросе. Если запрос не содержит ни одной записи, то отклик на такой запрос не передается.

Отклики

Генерация и передача отклика может быть инициирована любым из перечисленных ниже событий:

- получение запроса, как описано выше;
- регулярное (периодическое) обновление или незапрашиваемый отклик;
- обновление по событию (рассылка изменений с использованием группового или широковещательного адреса).

При периодическом обновлении или обновлении по событию отклик (сообщение обновления) посылается всем соседям (напрямую подключенным сетям) с использованием прямой адресации или в широковещательном режиме (групповой рассылки в RIP-2).

Периодические сообщения обновления рассылаются с использованием групповых или широковещательных адресов соседним маршрутизаторам каждые 30 с и включают полную таблицу маршрутизации. Если число записей, которые необходимо передать, превышает 25, то генерируется несколько сообщений обновления.

Генерация и передача сообщений об обновлении по событию имеет особенности. Во-первых, такие сообщения об обновлении при большой их частоте могут приводить к перегрузке сетей, поэтому принимаются меры по ограничению частоты обновлений по событию. В частности, после передачи такого обновления производится установка таймера на случайный промежуток времени от 1 до 5 с, в течение которого передача обновлений

по событию не производится, даже если события произошли в течение данного временного интервала. Кроме того, обновления по событиям не передаются, если пришло время генерации периодического обновления. Во-вторых, в обновлениях по событию передача полных таблиц маршрутизации запрещена. В сообщение обновления по событию включаются только те маршрутные записи, для которых установлен флаг изменения маршрута (или другие по усмотрению разработчика). После передачи сообщений обновления по событию соседним маршрутизаторам флаг изменения маршрута снимается.

Обработка маршрутизатором полученного отклика не зависит от причины, породившей передачу этого сообщения обновления соседним маршрутизатором (переданный ранее запрос, периодическое обновление или обновление по событию). Обработка полученного отклика включает следующую последовательность действий.

1. Проверка корректности дейтаграммы в целом.
2. Проверка корректностей маршрутных записей в сообщении.
3. Расчет метрики пути для каждой записи в отклике.
4. Изменение (обновление) записей таблицы маршрутизации.
5. Активизация процесса для инициирования отклика (сообщения обновления), при необходимости (при бесконечной метрике) — процесса удаления маршрута.

При проверке дейтаграммы в целом проверяются номер порта, из которого отправлен отклик, IP-адрес отправителя отклика. Если номер порта отличен от порта UDP, закрепленного за RIP, или дейтаграмма отправлена маршрутизатором, который не является непосредственным (и допустимым) соседом, то такие отклики игнорируются. Должны игнорироваться также копии своих пакетов, переданных с широковещательным или групповым адресом.

Проверка маршрутных записей RTE в сообщении включает проверку корректности адреса получателя, а также корректности значения метрики (от 1 до 16 включительно). При обнаружении некорректностей запись игнорируется.

После завершения проверки маршрутных записей производится расчет значений метрики в записях путем добавления метрики сети, из кото-

рой принято сообщение обновления. Если результирующее значение метрики превышает 15, то устанавливается метрика 16 (бесконечность).

После этого производится обновление записей в таблице маршрутизации. Сначала проверяется наличие явного маршрута для данного адресата. Если такой записи нет в таблице и если метрика для нее не имеет бесконечного значения, то она добавляется в таблицу. Процесс добавления записи в таблицу включает:

- установку в поле метрики полученного в результате расчета значения;

- установку в поле next-hop (следующий маршрутизатор) адреса маршрутизатора, от которого получено сообщение обновления;

- инициализацию отсчета тайм-аута (если уже включен таймер сбора мусора, то он останавливается);

- установку флага изменения маршрута;

- активизацию процесса для генерации сообщения обновления для соседних маршрутизаторов.

Если маршрут уже присутствует в таблице, то сравнивается значение поля next-hop с адресом маршрутизатора, от которого получена дейтаграмма. Если дейтаграмма пришла от указанного в записи маршрутизатора, то отсчет тайм-аута для маршрута начинается заново. После этого проверяется значение метрики. Если метрика отличается от имеющейся в записи или новая метрика меньше старой (независимо от передавшего ее маршрутизатора), то выполняются следующие операции:

- в таблицу вносятся изменения (т. е., корректируется метрика и при необходимости изменяется поле next-hop);

- устанавливается флаг изменения маршрута и активизируется процесс инициирования сообщения обновления.

Если новая метрика бесконечна, то активизируется процесс удаления маршрута, который больше не используется для пересылки. Необходимо заметить, что процесс удаления инициируется только при первоначальной установке бесконечной метрики. Если метрика уже была бесконечной, новый процесс удаления не запускается.

Таймеры для маршрутных записей RIP

Каждому маршруту в таблице сопоставляется два таймера: таймер тайм-аута (timeout) и таймер «сборки мусора» (garbage-collection). Указан-

ные таймеры необходимы для того, чтобы не хранить и вовремя удалять устаревшие маршруты (например, при выходе из строя какого-либо маршрутизатора на данном пути).

Отсчет тайм-аута (запуск таймера тайм-аута) начинается при организации маршрута и каждом обновлении данного маршрута (периодическом, или при обновлении по событию). Длительность тайм-аута ограничена 180 с. Если с момента после последней инициализации тайм-аута прошло 180 с, то маршрут объявляется недоступным и начинается процесс его удаления. Удаление маршрута может произойти в двух случаях: по тайм-ауту (если в течение 180 с не поступило ни одного сообщения обновления маршрута) и в результате установки для метрики значения 16 на основании обновлений, принятых от текущего маршрутизатора. В обоих случаях выполняются следующие действия:

- запускается таймер “сбора мусора” (удаления маршрута) – 120 секунд;

- для маршрута устанавливается метрика 16 (бесконечность) в результате чего маршрут перестает обслуживаться;

- устанавливается флаг смены маршрута, показывающий изменение записи;

- запускается процесс активизации отклика.

Пока (через 120 с) не наступит время «сборки мусора» (время удаления маршрута), маршрут продолжает включаться во все передаваемые маршрутизатором сообщения обновления. По истечении 120 с маршрут окончательно удаляется из таблицы. Если в процессе ожидания времени «сборки мусора» взамен утраченного организован новый маршрут в ту же сеть, то таймер сбрасывается.

Отметим, что кроме двух рассмотренных таймеров при получении сообщения обновления по событию запускается специальный таймер обновления по событию (на случайное время от 1 до 5 с), описанный выше.

Пример таблицы маршрутизации с использованием флагов и таймеров для маршрутизатора *A* на рис. 5.8 приведен в табл. 5.5 (ТОТ — таймер тайм-аута; GCT — таймер удаления маршрута). Здесь в строках 1 и 4 таймер тайм-аута имеет значение 20 с (с момента получения сообщения обновления от маршрутизатора *B* прошло 20 с). Значение таймера тайм-аута 0 в строках 2 и 3 говорит о том, что сообщение только что получено, про-

ведено обновление маршрутных записей в строках, и для активирования сообщения о произошедших изменениях соседним маршрутизаторам в данных записях установлен флаг изменения маршрута RCF. Запись в последней строке означает, что с момента получения обновления о данном маршруте прошло (180 + 80) с, т. е. отработал таймер тайм-аута, после чего данный маршрут стал считаться недоступным (его метрика получила значение 16 — бесконечность). Затем запустился таймер «сборки мусора», который уже отработал 80 с, и до истечения 120-секундного интервала маршрут будет сохраняться в таблице. Если в течение этого времени не поступит обновления для данной записи, то маршрут будет удален. Причиной этого мог быть выход из строя маршрутизатора *D*.

Таблица 5.5

Информация, хранящаяся в RIP-маршрутизаторе (A)

Номер сети	Адрес следующего маршрутизато-	Порт	Метрика	Таймеры		Флаги
				ТОТ	GCT	
195.27.18.0	195.27.18.1	1	1	20	—	
132.11.0.0	132.11.0.2	2	1	0	—	RCF
132.15.0.0	132.11.0.1	2	2	0	—	RCF
195.27.19.0	195.27.18.1	3	2	20	—	
202.101.16.0	195.27.18.1	3	16	180	80	

Ограничения и недостатки протокола RIP

Протокол RIP широко распространен благодаря своей простоте и удобству использования в небольших объединенных сетях, однако ему свойственны некоторые ограничения.

1. По мере роста размеров объединенной сети адресаты, расстояния до которых превышают 15, становятся недоступными. Если же разрешить использование больших (чем 15) значений метрики, то время сходимости протокола становится может оказаться недопустимо большим.

2. Использование чрезмерно упрощенной метрики (целочисленной в пределах от 1 до 15) приводит к выбору неоптимальных маршрутов при формировании таблиц маршрутизации.

3. Маршрутизаторы, поддерживающие протокол RIP, принимают обновления от любых RIP-устройств, что может привести к нарушению их конфигурации из-за одного неверно сконфигурированного устройства.

Одним из наиболее серьёзных недостатков протокола RIP является его медленная реакция на изменения состояния (топологии) сети. Это касается, прежде всего, изменений деградационного характера (выхода из строя маршрутизаторов или отдельных линий), которые приводят к проблеме счёта до бесконечности.

Рассмотрим эту проблему на примере объединенной сети рис. 5.6, схематично представленной в виде графа на рис. 5.8. Предположим, что метрики всех линий равны 1, а периодическая (каждые 30 с) рассылка сообщений обновления производится маршртутизаторами сети синхронно. На рис. 5.8 числами в строках показаны метрики путей от соответствующего узла до узла *D* в исходном состоянии и после рассылки каждого сообщения обновления.

-	-	-	исходное состояние	2	2	1
-	-	1	после 1-го обновления	2	2	3
2	2	1	после 2-го обновления	4	4	3
			после 3-го обновления	4	4	5
			после 4-го обновления	6	6	5
			после 5-го обновления	6	6	7
			⋮		⋮	
			после 15-го обновления	16	16	16

Рис. 5.8. Проблема счета до бесконечности

Слева на рис. 5.8 изображен случай, когда узел *D* вначале отсутствовал в сети. Известие о его появлении распространяется по всей сети за 2 шага (после рассылки второго обновления), т. е. примерно за 30…60 с.

Справа на рис. 5.8 рассматривается случай, когда узел *D* первоначально присутствовал в сети, а затем вышел из строя. После выхода из строя маршрутизатора *D* узел *B* перестает получать от него сообщения об обновлении, но продолжает получать их от узлов *A* и *C*. При первом обновлении маршрутизаторы *A* и *C* сообщат узлу *B*, что расстояние от них до узла *D* равно 2, и маршрутизатор *B*, добавив к этой метрике расстояние до сети *A* (или *C*), запишет в маршрут до узла *D* значение метрики 3. После рассылки второго сообщения обновления маршрутизаторы *A* и *C*, получив от *B* значение метрики 3 до узла *D*, в свою очередь, добавят к нему по 1 и т. д. Видно, что информация о недоступности сети *D* распространяется по сети очень медленно, процесс будет завершен примерно через 8 мин.

5.4. ПРОТОКОЛ OSPF

Протокол OSPF (Open Shortest Path Firs — «выбор кратчайшего пути первым») является протоколом внутренней маршрутизации и применительно к сети интернет предназначен для использования внутри автономных систем. Первая его версия была принята в 1990 г., а современная вторая версия описана в документе RFC 2328 от 1998 г., который заменил RFC 2178. Протокол OSPF относится к классу протоколов на основе состояния связей, и для поиска кратчайших путей использует алгоритм Дейкстры.

Среди основных особенностей протокола OCPF можно выделить следующие.

1. Все OSPF-маршрутизаторы поддерживают идентичные базы данных с описанием топологической структуры сети (автономной системы, области сети или подсети). На основании данных из базы каждый маршрутизатор рассчитывает кратчайшие маршруты до остальных узлов и сетей.

2. Каждый маршрутизатор пересылает другим узлам сети не полную таблицу маршрутизации (как в протоколе RIP), а маршрутные сообщения, содержащие информацию только о состоянии своих связей (каналов) с соседними маршртутизаторами. Причем, маршрутные сообщения, называемые в протоколе OSPF объявлениями о состоянии связей (Link State Advertisements, LSA) или анонсами, посылаются не периодически, как в RIP, а только при изменении состояния связей (каналов). Это существенно

снижает служебный трафик в сети и увеличивает эффективность использования пропускной способности сети.

3. Протокол OSPF может использоваться в иерархических сетях или, как указано в RFC 2328, поддерживается маршрутизация областей (area routing), за счет чего обеспечивается дополнительный уровень защиты маршрутизации и снижение служебного трафика при обмене маршрутной информацией.

4. OSPF поддерживает множество равноценных маршрутов, поскольку таблица маршрутизации может включать насколько маршрутных записей с одинаковой метрикой к одному адресату. В этом случае маршрутизатор имеет возможность распределения нагрузки по равноценным маршрутам, что снижает вероятность перегрузки в отдельных направлениях связи и узлах сети.

5. OSPF использует концепцию выделенного (назначенного) маршрутизатора (Designated Router, DR), что также обеспечивает снижение уровня служебного трафика.

6. Используемые метрики: пропускная способность (по умолчанию), задержка, надежность, стоимость.

7. Обмен маршрутной информации осуществляется с использованием средств аутентификации.

8. Протокол включает явную поддержку бесклассовой междоменной маршрутизации.

Каждый OSPF-маршрутизатор на основе маршрутной информации, получаемой от других маршрутизаторов сети, строит направленный граф сети, в котором вершинами графа являются маршрутизаторы и сети, а ребрами — интерфейсы маршрутизаторов. Сети в автономной системе (или подсети в объединенной сети) могут быть оконечными (тупиковыми) или транзитными. Транзитные сети обозначаются вершинами графа, имеющими входящие и исходящие ребра, и могут передавать транзитный трафик, когда отправитель и получатель сообщений принадлежат другим сетям. Вершины графов, обозначающие тупиковые имеют только входящее ребро.

Пример топологической структуры автономной системы в виде графа показан на рис. 5.9, на котором вершины, соответствующие сетям изображены эллипсами, а маршрутизаторы — квадратами. Метрики ребер (ин-

терфейсов) между маршртутизаторами и сетями в обоих направлениях показаны числами.

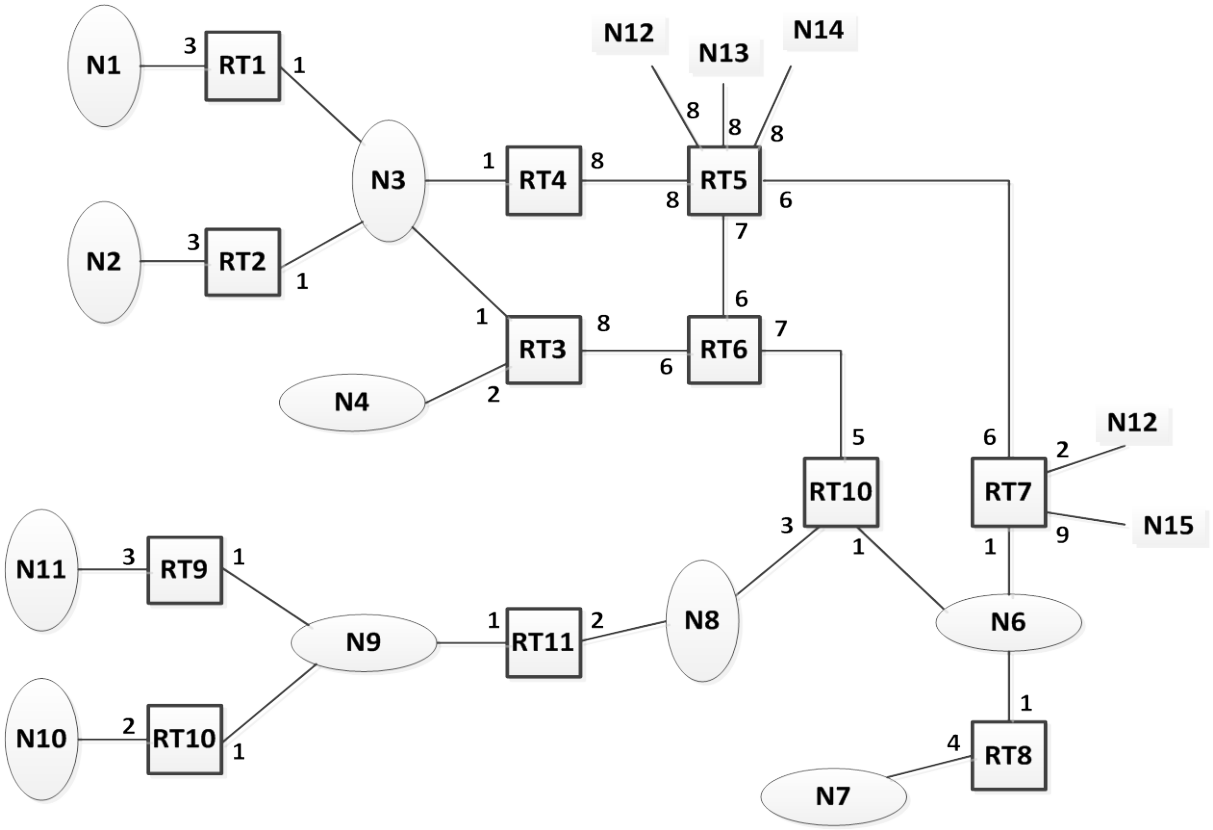


Рис. 5.9. Пример автономной системы без областей

Сети N1, N2, N4, N7, N10, N11 являются оконечными, сети N3, N6, N8, N9 — транзитными, а сети N12…N15 — внешними, т. е. принадлежат другим автономным системам.

Протокол OSPF позволяет объединять смежные сети, маршрутизаторы и хосты в группы, называемые областями. Пример сети с областями приведен на рис. 5.10.

Маршрутизаторы, непосредственно подключенные к сетям внутри каждой области, называют внутренними. К примеру, внутренними для области 1 на рис. 5.10 являются маршрутизаторы RT1 и RT2, для области 3 — RT9 и RT10, для области 2 — RT8, для магистральной части сети — RT5 и RT6.

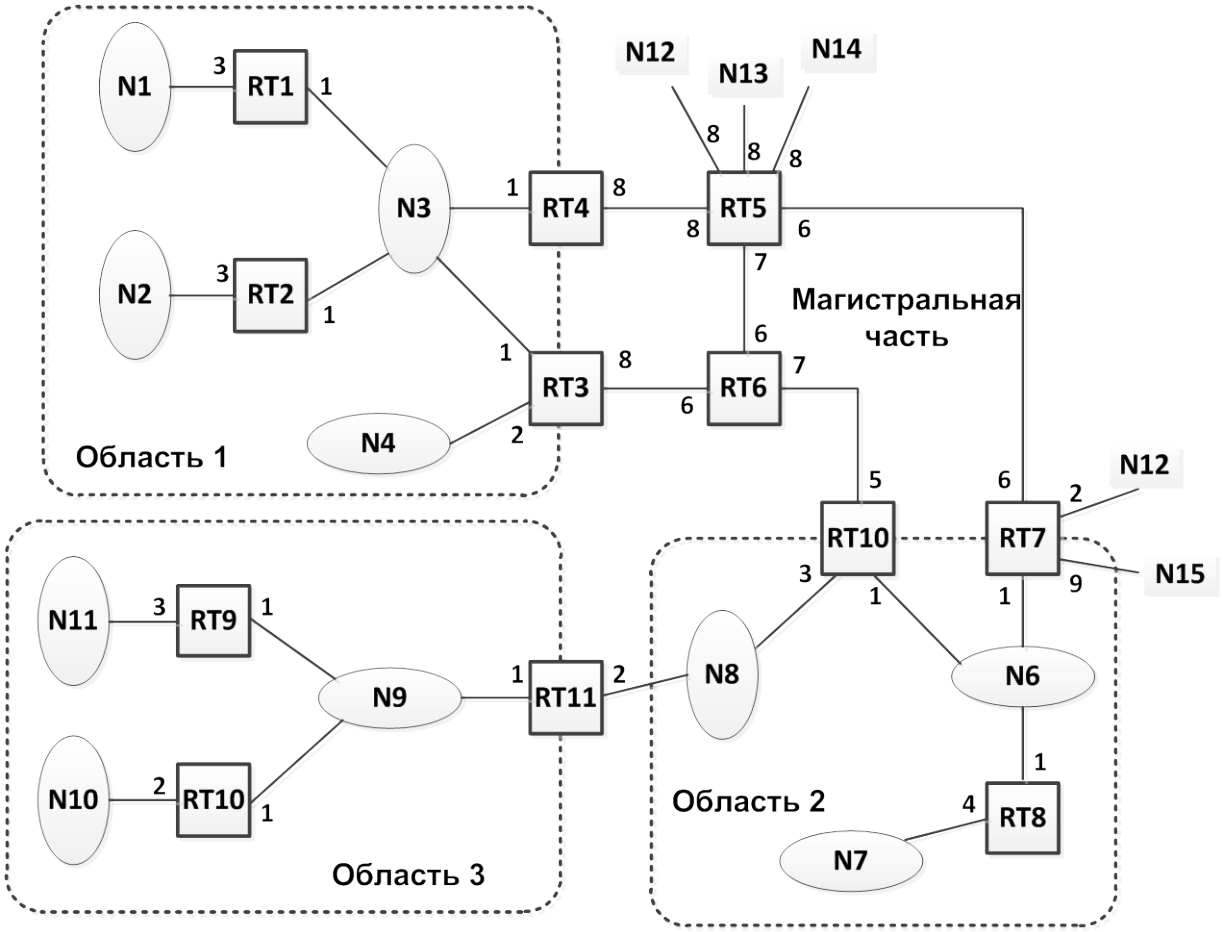


Рис. 5.10. Пример автономной системы с делением на области

Маршрутизаторы, подключенные к нескольким областям сети, называются граничными. Маршрутизаторы RT3, RT4, RT7, RT10 и RT11 являются граничными маршртутизаторами областей, а RT5 и RT7 — граничными маршртутизаторами автономной системы.

Граничные маршрутизаторы автономной системы — это маршрутизаторы, обменивающиеся маршрутными данными с маршртутизаторами других автономных систем. Магистральные маршрутизаторы — это маршрутизаторы, имеющие интерфейс в магистральную область (RT3, RT4, RT5, RT6, RT7, RT10 и RT11). На рис. 5.10 между маршртутизаторами RT10 и RT11 организован виртуальный канал, что позволяет включить маршрутизатор RT11 в магистральную область. Область 2 для этого виртуального соединения является транзитной.

В каждой области работает своя независимая копия алгоритма маршрутизации. Это означает, что в каждой области, включая магистральную

часть сети, поддерживается своя база каналов и соответствующий граф области сети. Топология области невидима за ее пределами и наоборот, внутренние маршрутизаторы области ничего не знают о топологии сети за пределами данной области. Маршрутизаторы, относящиеся к одной области, имеют идентичные базы данных о каналах, а маршрутизаторы, находящиеся в разных областях сети, имеют и различные базы данных. Такое разделение сведений о топологической структуре сети позволяет существенно снизить объем служебного трафика.

Маршрутизация в автономной системе осуществляется на двух уровнях в зависимости от местоположения отправителя и получателя. Если отправитель и получатель находятся в одной области, то используется внутридоменная (внутри области) маршрутизация (intra-area routing). При внутридоменной маршрутизации используются только сведения, полученные из данной области. Если же отправитель и получатель расположены в разных областях, то применяется междоменная маршрутизация (inter-area routing).

База данных, поддерживаемая каждым внутренним маршрутизатором области 1, приведена на рис. 5.11.

Выделенная часть таблицы представляет собой данные о каналах внутри области 1, собираемые каждым из маршрутизаторов RT1…RT4 путем обмена маршрутной информацией о состоянии связей (метриках) между ними. Эта часть таблица полностью описывает внутридоменную (внутри области 1) маршрутизацию первого уровня иерархии. Кроме того, в таблице приведены данные о результирующих метриках путей до сетей, находящихся в других областях и в других автономных системах. Эти данные транслируются в область 1 граничными маршртутизаторами RT3 и RT5.

На граничных маршрутизаторах областей используется несколько копий базового алгоритма маршрутизации — по одной копии на каждую подключенную к маршрутизатору область. Граничные маршрутизаторы областей собирают маршрутную информацию областей (перечень сетей и метрики путей до них) для передачи этих сведений в магистральную область. Магистраль распространяет полученную информацию между всеми областями, а магистральные маршрутизаторы обеспечивают междоменную маршрутизацию второго уровня иерархии.

От узла

К узлу	RT1	RT2	RT3	RT4	N3	RT5	RT7
RT1					0		
RT2					0		
RT3					0		
RT4					0		
N1	3						
N2		3					
N3	1	1	1	1			
N4			2				
RT5			14	8			
RT7			20	14			
Ia, Ib			20	27			
N6			16	15			
N7			20	19			
N8			18	18			
N9…N11			29	36			
N12						8	2
N13						8	
N14						8	
N15							9

Рис. 5.11. База данных области 1

Таким образом, граничные маршрутизаторы областей участвуют во внутридоменной и междоменной маршрутизации, а также транслируют суммарную маршрутную информацию области (перечень сетей и результирующие метрики путей до них) в магистральную часть и наоборот – транслируют суммарную внешнюю маршрутную информацию от других областей и AS в данную область.

На основе базы данных о каналах строится таблица маршрутизации. Пример таблицы маршрутизации для граничного маршрутизатора RT4 области 1 показан в табл. 5.6.

Таблица 5.6

Таблица маршрутизации маршрутизатора

Тип адресата	Адрес назначения	Маска	Область	Тип пути	Метрика	Next hop
N	N1		1	Внутридомен.	4	RT1
N	N2		1	Внутридомен.	4	RT2
N	N3		1	Внутридомен.	1	*
N	N4		1	Внутридомен.	3	RT3
R	RT3		1	Внутридомен.	1	*
N	Ia		0	Внутридомен.	27	RT5
N	Ib		0	Внутридомен.	22	RT5
R	RT3		0	Внутридомен.	21	RT5
R	RT5		0	Внутридомен.	8	*
R	RT7		0	Внутридомен.	14	RT5
R	RT10		0	Внутридомен.	22	RT5
R	RT11		0	Внутридомен.	25	RT5
N	N6		0	Междомен.	15	RT5
N	N7		0	Междомен.	19	RT5
N	N8		0	Междомен.	18	RT5
N	N9… N11		0	Междомен.	36	RT5
N	N12		*	Внешн. типа 1	16	RT5, RT7
N	N13		*	Внешн. типа 1	16	RT5
N	N14		*	Внешн. типа 1	16	RT5
N	N15		*	Внешн. типа 1	23	RT7

Каждая маршрутная запись содержит следующую маршрутную информацию:

- тип адресата — сеть (N) или маршрутизатор (R);
- адрес назначения — IP-адрес сети назначения или маршрутизатора;
- маска сети или подсети;
- номер области;

- тип маршрута — внутридоменный, междоменный, внешний типа 1, внешний типа 2;

- стоимость (метрика) пути;

- следующий маршрутизатор (Next hop) — выходной интерфейс маршрутизатора, используемый для пересылки пакетов получателю, а в широковещательных сетях — IP-адрес первого маршрутизатора на пути к адресату;

- маршрутизатор-источник маршрутной информации (в таблице данное поле не показано).

Адресатом может быть сеть или маршрутизатор. В таблице сохраняются записи, связанные с граничными маршртутизаторами областей и автономных систем. Записи в таблице маршрутов для граничных маршрутизаторов областей используются при расчете междоменных маршрутов и для поддержки виртуальных соединений, а записи для граничных маршрутизаторов автономных систем служат для расчета внешних по отношению к ним маршрутов.

Поле *Метрика* для всех путей кроме внешних типа 2 содержит значение метрики пути в целом. Для внешних путей типа 2 это поле описывает метрику части пути внутри данной автономной системы. Эта метрика вычисляется как сумма метрик составляющих путь каналов. Для внешних путей к другим автономным системам используется поле *Type 2 cost* (в табл. 5.6 не показано). Оно содержит значение метрики внешней части пути (внутри других автономных систем).

Маршрутизатор RT4 соединен с областью 1 и с магистральной областью. Поэтому таблица маршрутизации содержит маршрутные записи для внутридоменной маршрутизации внутри области 1 и маршрутные записи для внутридоменной маршрутизации внутри магистральной области. Магистральные пути рассчитываются до всех граничных маршрутизаторов областей, которые служат для расчета междоменных маршрутов.

Кроме того, таблица маршрутизации маршрутизатора RT4 содержит междоменные маршруты до сетей в областях 2 и 3 (N6…N11), а также внешние пути, которые ведут к другим автономным системам (к сетям N12…N15). В рассматриваемом примере предполагается, что область 3 определена так, что сети N9…N11 находятся на одном маршруте, анонси-

руемом в магистраль маршрутизатором RT11 через виртуальный канал между RT11 и RT10.

Внешние маршрутные данные рассылаются путем лавинной маршрутизации через всю автономную систему, поэтому все маршрутизаторы данной автономной системы, к примеру, RT7, имеют два внешних маршрута с метрикой 2 и 9. Внешняя маршрутная информация может быть получена от других протоколов маршрутизации (например, BGP), функционирующих на граничных маршрутизаторах автономной системы, или задаваться статически администратором сети.

Пути до сетей в других автономных системах бывают двух типов: внешние типа 1 и внешние типа 2. Тип 1 использует для метрики те же единицы измерения, что и для метрики внутренних маршрутов в автономной системе. Поэтому результирующая метрика пути до сети в другой автономной системе рассчитывается как сумма метрики внутреннего маршрута до граничного маршрутизатора автономной системы и метрики внешнего пути типа 1. Если до внешней сети в другой автономной системе имеется два пути через различные граничные маршрутизаторы автономной системе, то из них выбирается путь с меньшей метрикой.

При использовании внешних маршрутов типа 2 предполагается, что маршрутизация между автономными системами составляет основную часть стоимости пересылки пакетов. Это избавляет от необходимости преобразования внешней метрики во внутренние единицы. Среди нескольких маршрутов типа 2 к одному адресату выбирается тот, который имеет меньшую внешнюю метрику, независимо от значения метрики пути до граничного маршрутизатора внутри автономной системы. Если же метрики внешних путей равны, то выбор осуществляется с учетом метрики внутреннего маршрута.

Внешние пути типа 1 предпочтительней, нежели пути типа 2. Когда все внешние пути относятся к типу 2, среди них предпочтительным является путь с наименьшей анонсируемой метрикой.

При получении пакета данных IP маршрутизатор OSPF находит запись таблицы маршрутизации, наиболее точно соответствующую адресату. Из этой записи определяется выходной интерфейс маршрутизатора и следующий маршрутизатор для пересылки пакета.

Как указывалось выше, маршрутизаторы обмениваются между собой маршрутной информацией путем рассылки объявлений о состоянии связей LSA. Эти маршрутные сообщения содержат информацию маршрутизатора только о состоянии своих связей с соседями и рассылаются только смежным маршрутизаторам. Далее LSA распространяются по сети путем лавинной маршрутизации. Объявления о состоянии связей рассылаются только смежным маршрутизаторам. Смежными называют маршрутизаторы, между которыми передаются обновления маршрутной информации. Таким образом, соседние маршрутизаторы не обязательно будут смежными.

Для уменьшения числа смежных пар узлов, а, следовательно, и снижения уровня служебного трафика в сети, протокол OSPF поддерживает концепцию выделенного маршрутизатора. Выделенный (назначенный) маршрутизатор назначается в широковещательной сети (области сети), включающей не менее двух маршрутизаторов.

Выделенный маршрутизатор DR является смежным для всех остальных маршрутизаторов сети и поэтому все остальные маршрутизаторы сети (области сети) передают LSA выделенному маршрутизатору, который в свою очередь рассылает их по сети. Выделенный маршрутизатор также генерирует маршрутные объявления LSA для сети. Эти LSA содержат список маршрутизаторов (включая DR), подключенных в данный момент к сети.

Для повышения устойчивости маршрутизации в сети назначается также резервный выделенный маршрутизатор (backup DR), который берет на себя функции DR, когда тот выходит из строя. Такой маршрутизатор в состоянии резерва обновления маршрутной информации не передает. За DR и Backup DR закреплен групповой адрес 225.0.0.6, по которому они должны быть готовы принимать сообщения от других маршрутизаторов области сети.

Пример назначения DR и Backup DR в магистральной области автономной системы (рис. 5.10) и граф смежности для магистральной области приведены на рис. 5.12. Маршрутизаторы RT4…RT6, RT3 посылают свои LSA выделенному маршрутизатору RT7, который пересылает их LSA и сгенерированные им сетевые LSA всем смежным с ним маршрутизаторам. При выходе из строя RT7 функции выделенного маршрутизатора переходят к RT3 (его смежные связи показаны точечной линией).

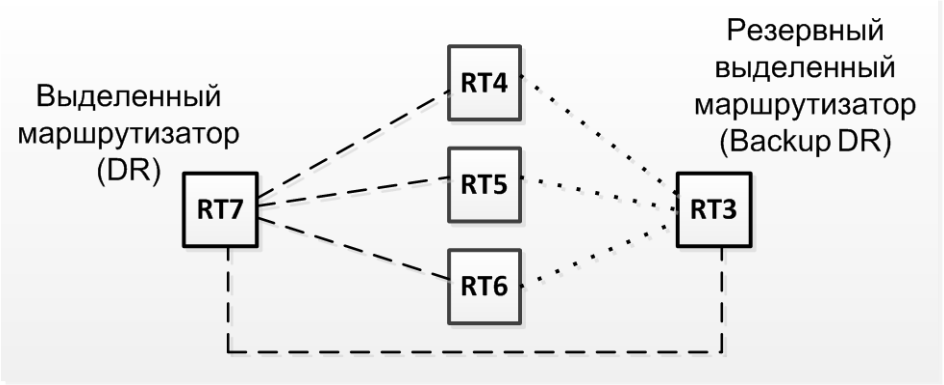


Рис. 5.12. Граф смежности для магистральной области

Служебные сообщения, которыми в процессе функционирования обмениваются маршрутизаторы, пересылаются в OSPF-пакетах. В качестве транспорта для пересылки OSPF-пакетов используется протокол IP, т. е. OSPF работает «поверх» IP. Протокол OSPF имеет в IP зарегистрированный в NIC идентификатор 89, который при инкапсуляции OSPF-пакетов в пакет IP прописывается в поле заголовка IP-пакета как «протокол верхнего уровня».

Все служебные пакеты OSPF передаются с использованием значения поля TOS заголовка IP-пакета «нормальное обслуживание» (биты флагов D, T, R, C установлены в 0000). Вместе с тем, пакеты OSPF должны иметь преимущество перед обычными пакетами данных IP как для приема, так и для передачи. Это реализуется путем установки соответствующего значения «Internetwork Control» в поле приоритета поля TOS заголовка IP-пакета.

Всего в протоколе OSPF предусмотрено пять типов OSPF-пакетов, наименование и назначение которых даны в табл. 5.7. Пакеты протокола OSPF (за исключением пакетов Hello) передаются только между смежными маршртутизаторами.

Каждый пакет OSPF начинается со стандартного заголовка размером 24 байта, содержащего всю информацию, требуемую для дальнейшей обработки пакета (рис. 5.13).

Заголовок OSPF-пакета включает следующие поля:

- номер версии протокола OSPF (в настоящее время действует версия 2);

- тип пакета;

Таблица 5.7

Типы OSPF-пакетов

Тип пакета	Название	Назначение
1	*Hello* — приветствие	Обнаружение и поддержка соседских отношений, определение DR или указание на DR (короткий пакет – 1 раз в 10 сек).
2	*Database Description* (DD) — описание базы данных	Синхронизация баз данных смежных маршрутизаторов.
3	*Link State Request* (LSR) — запрос состояния канала	Для запроса маршрутной информации у смежного маршрутизатора.
4	*Link State Update* (LSU) — обновление состояния канала	Передача маршрутной информации (анонсов LSA).
5	*Link State Ack* (LSAck) — подтверждение приема	Подтверждение приема маршрутной информации (анонсов LSA).

- размер пакета в байтах с учетом стандартного заголовка;
- идентификатор маршрутизатора-отправителя пакета;
- 32-битовый идентификатор области, к которой относится пакет (магистральная область помечается идентификатором 0.0.0.0);
- контрольная сумма, вычисляемая с учетом заголовка OSPF-пакета, но без учета 64-битового поля данных аутентификации;
- поле *Тип аутентификации* определяет используемую для пакета процедуру аутентификации;
- данные используемой схемы аутентификации.

Пакеты HELLO периодически (каждые 10 с) передаются во все интерфейсы маршрутизатора (соседним маршрутизаторам) для организации соседских отношений и контроля состояния связей с соседними маршртутизаторами. Наличие двухсторонней связи между соседями устанавливается, когда маршрутизатор видит себя в пакете Hello от соседа. Протокол Hello используется также для выбора или указания выделенного маршрутизатора DR для данной сети. Сообщения HELLO имеют небольшой объем, что не

приводит к существенному увеличению служебного трафика в сети при указанной частоте их передачи. На основании принимаемых от непосредственных соседей сообщений HELLO маршрутизатор формирует записи о состоянии связей со своими непосредственными соседями в базе данных о топологии сети.

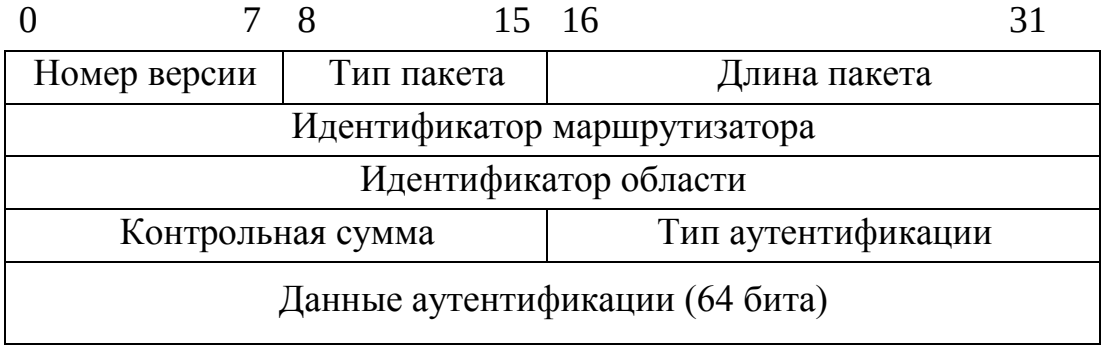


Рис. 5.13. Формат заголовка OSPF-пакета

Передача пакетов HELLO осуществляется по групповому адресу 225.0.0.5, причем все маршрутизаторы должны быть готовы принимать сообщения по этому адресу. Эти пакеты не должны пересылаться далее одного ретрансляционного участка, поэтому для предотвращения их дальнейшей пересылки время их жизни ограничивается значением поля TTL в заголовке IP-пакета, равном TTL = 1.

Формат пакета HELLO приведен на рис. 5.14.

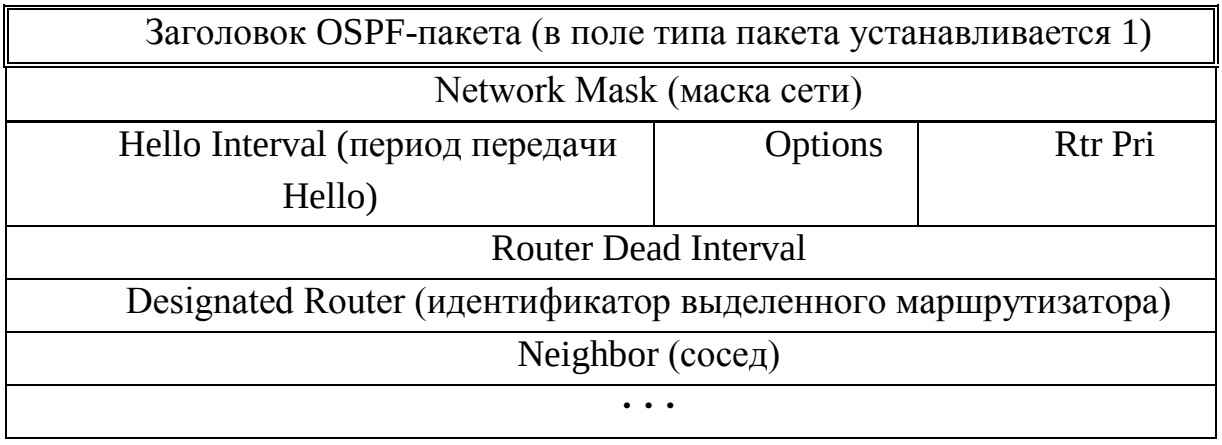


Рис. 5.14. Формат пакета HELLO

В поле *Network Mask* записывается маска сети, связанной с интерфейсом маршрутизатора. Например, сети класса B имеют маску 0xffffff00.

В поле *Hello Interval* указывается интервал времени между передачей пакетов HELLO в секундах.

В поле *Options* (опции) с помощью флагов (в RFC 2328 описаны 5 флагов) указываются поддерживаемые маршрутизатором дополнительные возможности. Это поле присутствует не только в пакетах HELLO, но и в пакетах DD и всех типах LSA. Использование этого поля в пакетах HELLO позволяет маршрутизатору отказаться от соседа при рассогласовании возможностей.

Rtr Pri (Router Priority) — значение приоритета маршрутизатора, используемое при выборе выделенного маршрутизатора DR (Backup DR) в сети или в области сети;

В поле *Router Dead Interval* задается интервал времени (в секундах) по истечении которого при отсутствии приема пакетов HELLO маршрутизатор считается неработоспособным (время исчисляется от момента приема последнего пакета Hello от соседнего маршрутизатора);

В поле *Designated Router* указывается идентификатор выделенного маршрутизатора DR для данной сети с точки зрения передающего маршрутизатора. DR идентифицируется по IP-адресу в сети. Если маршрутизатора DR нет, поле имеет значение 0.0.0.0.

В поле *Neighbor* приводятся идентификаторы маршрутизаторов, чьи пакеты Hello были недавно приняты (в течение Router Dead Interval).

После обнаружения соседей путем обмена пакетами HELLO и установления отношений смежности начинается процесс синхронизации баз данных смежных маршрутизаторов, реализуемый с помощью обмена сообщениями *Database Description* (DD). Пакеты DD описывают содержимое базы данных о каналах. Один из маршрутизаторов является ведущим (master), а второй — ведомым (slave). Ведущий маршрутизатор передает пакеты DD, которые подтверждаются пакетами DD от ведомого маршрутизатора. Отклики связываются с опросом через порядковые номера пакетов DD.

Формат пакетов DD (это OSPF-пакеты 2-го типа) приведен на рис. 5.15.

Пакет содержат список элементов, каждый из которых описывает часть базы данных о каналах маршрутизатора.

<table>
<tr><td colspan="10">Заголовок OSPF-пакета (в поле типа пакета устанавливается 2)</td></tr>
<tr><td colspan="3">Interface MTU</td><td colspan="3">Options</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>I</td><td>M</td><td>MS</td></tr>
<tr><td colspan="10">DD sequence number (порядковый номер пакета DD)</td></tr>
<tr><td colspan="10">LSA Header (заголовок LSA)</td></tr>
<tr><td colspan="10">. . .</td></tr>
</table>

Рис. 5.15. Формат пакета DD

В поле *Interface MTU* задается максимальный размер (в байтах) IP-пакета, который может быть передан через интерфейс без фрагментации.

В поле опций указываются дополнительные возможности маршрутизатора.

Флаги:

I — бит *Init*, устанавливаемый в 1 для первого (по порядку) пакета DD;

M — бит *More*, указывающий на присутствие других (последующих) пакетов DD;

MS — бит *Master/Slave*, определяющий отношения маршрутизаторов в процессе синхронизации баз данных (MS = 1 — ведущий, 0 — ведомый);

DD sequence number (порядковый номер пакета) используется для нумерации пакетов DD. Начальное значение (указывается флагом Init) должно быть уникальным. Далее порядковый номер DD увеличивается на 1 для каждого пакета вплоть до передачи всей базы данных.

Остальная часть пакета содержит список частей базы данных маршрутизатора. Каждая запись LSA (объявление о состоянии канала) в базе описывается заголовком LSA (см. ниже), содержащим все данные для уникальной идентификации LSA и его текущего экземпляра.

После обмена пакетами DD с соседом маршрутизатор может понять, что часть его базы данных устарела, т. е. у смежного маршрутизатора имеются более свежие записи LSA. Такой вывод маршрутизатор делает из анализа заголовков LSA, в которых указан больший возраст и порядковый номер текущего экземпляра LSA (см. ниже). Тогда маршрутизатор посылает смежному маршрутизатору запрос LSR (запрос состояния канала) на получение более современных фрагментов базы данных.

Пакеты LSR относятся к типу 3 и имеют формат, показанный на рис. 5.16.

Заголовок OSPF-пакета (в поле типа пакета устанавливается 3)
LS type (тип LSA)
Link State ID (идентификатор части сети, описываемой данным LSA)
Advertising Router (ID маршрутизатора, породившего LSA)
. . .

Рис. 5.16. Формат пакета DD

В поле *LS type* указывается тип запрашиваемого LSA. Всего имеется 5 типов объявлений состояния связей, которые будут рассмотрены ниже.

В поле *Link State ID* указывается идентификатор (ID) области сети, описываемой запрашиваемым LSA, а в поле *Advertising Router* — идентификатор маршрутизатора-источника данного LSA.

Маршрутная информация в виде объявлений состояния связей (LSA) передается в пакетах LSU (обновление состояния канала). Пакеты LSU относятся в OSPF к типу 4 и используются для лавинной рассылки объявлений состояния связей LSA. Каждый пакет LSU передает набор LSA на один интервал от точки их происхождения.

Каждый LSU-пакет начинается со стандартного 20-байтного заголовка, в котором в поле типа пакета указано значение 4 (рис. 5.17).

Заголовок OSPF-пакета (в поле типа пакета устанавливается 4)
LSA
LSA
. . .

Рис. 5.17. Формат пакета LSU

В поле *# LSAs* указывается число объявлений состояния связей LSA, включенных в этот пакет обновления. Далее в теле пакета LSU следуют объявления состояния связей LSA, передаваемые в данном пакете обновления.

Для обеспечения надежности процедуры лавинной рассылки отправленные LSA подтверждаются в пакетах LSAck. Если требуется повтор пе-

редачи некоторых LSA, они всегда передаются смежному маршрутизатору напрямую.

В протоколе OSPF предусмотрено 5 различных типов объявлений состояния связей, названия и описание которых приведены в табл. 5.8.

Таблица 5.8

Типы объявлений (анонсов) состояния связей LSA

Тип LSA	Название LSA	Описание LSA
1	Router-LSA	Генерируются всеми маршртутизаторами. Этот тип LSA описывает состояния интерфейсов маршрутизатора. Анонс рассылается в лавинном режиме внутри области.
2	Network-LSA	Генерируется выделенным маршрутизатором DR для широковещательных и NBMA-сетей. Этот тип LSA включает список маршрутизаторов, подключенных к сети. Рассылается в лавинном режиме внутри области.
3, 4	Summary-LSA	Генерируется граничными маршртутизаторами областей и рассылается в лавинном режиме в пределах связанной с LSA области. Каждый summary-LSA описывает маршрут к адресату за пределами данной области, но внутри данной автономной системы (междоменный маршрут). Тип 3 summary-LSA описывает маршруты в сети, а тип 4 — к граничным маршрутизаторам автономной системы.
5	AS-external-LSA	Генерируется граничными маршртутизаторами и рассылается по всей автономной системе. Каждый анонс AS-external-LSA описывает маршрут к адресатам в другой автономной системе.

Каждое объявление состояния связей (LSA) описывает часть маршрутного домена OSPF. Объявления router-LSA генерируются каждым маршрутизатором сети (области сети). В дополнение к этому при выборе маршрутизатора в качестве выделенного (DR), этот маршрутизатор порож-

дает сообщения network-LSA. Маршрутизаторы могут порождать и другие типы объявлений состояния связей LSAs.

Все LSA (за исключением LSA 5-го типа — AS-external-LSA) распространяются в сети путем лавинной рассылки внутри области сети (домена маршрутизации OSPF), не покидая пределы области. AS-external-LSA рассылаются по всей AS за исключением тупиковых областей. Рассылаемый набор LSA называют базой данных о состоянии каналов. Из базы данных о состоянии каналов каждый маршрутизатор создает дерево кратчайших путей с собой в качестве корня, что в конечном итоге дает таблицу маршрутизации.

Формат объявлений состояния связей (LSA) рассмотрим на примере объявления router-LSA (рис. 5.18). Каждое объявление состояния связей любого типа начинается с 20-байтного заголовка. Информации, содержащейся в заголовке, достаточно для уникальной идентификации LSA (поля *mun LSA, Link State ID* и *Advertising Router*). В области маршрутизации может одновременно существовать множество экземпляров LSA. Для определения последнего из них служат поля *LS age, LS sequence number* и *LS checksum* из заголовка LSA.

В поле *LS age* (возраст LSA) указывается время (в секундах) с момента генерации объявления. При генерации LSA поле имеет нулевое значение и должно увеличиваться на InfTransDelay при прохождении каждого интервала в процедуре лавинной рассылки. Параметр InfTransDelay представляет собой оценку времени передачи пакета через интерфейс (через один ретрансляционный участок). Для локальных сетей в RFC 2328 рекомендуется значение этого параметра выбирать равным 1 с. Время жизни экземпляра LSA ограничено максимальным значением MaxAge, которое в RFC 2328 рекомендуется выбирать равным 1 ч.

В поле опций указываются дополнительные возможности, которые поддерживаются в описываемой части домена маршрутизации.

Поле *mun LSA* определяет один из пяти типов маршрутных объявлений (см. табл. 5.8), которые могут пересылаться маршрутизатором в пакетах LSU. В примере на рис. 5.18 в этом поле указан тип LSA = 1, т. е. router-LSA.

Поле *Link State ID* идентифицирует часть сети, описываемую анонсом LSA. Содержимое поля зависит от типа LSA. Например, в network-LSA

поле *Link State ID* содержит IP-адрес интерфейса выделенного маршрутизатора DR для данной сети (по этому адресу определяется номер сети IP). В LSA типа 1 (router-LSA) в этом поле указывается идентификатор ID маршрутизатора, генерирующего данное объявление, а в LSA типа 4 (Summary-LSA) — идентификатор ID граничного маршрутизатора автономной системы.

Поле заголовка *Advertising Router* содержит идентификатор ID маршрутизатора, создавшего LSA. Для router-LSA значение этого поля совпадает со значением поля *Link State ID*.

<table>
<tr><td colspan="9">0 7 8 15 16 23 24 31</td></tr>
<tr><td colspan="8">LS age</td><td>Options</td><td>Tun LSA (=1 для router-LSA)</td></tr>
<tr><td colspan="9">Link State ID (идентификатор части сети, описываемой данным LSA)</td></tr>
<tr><td colspan="9">Advertising Router (ID маршрутизатора, породившего LSA)</td></tr>
<tr><td colspan="9">LS sequence number (порядковый номер LSA)</td></tr>
<tr><td colspan="8">Checksum (контр. сумма без LS age)</td><td>Length (размер LSA в байтах)</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>V</td><td>E</td><td>B</td><td>0</td><td># links (число каналов, описываемых в LSA)</td></tr>
<tr><td colspan="9">Link ID (идентификатор объекта, к которому подключен роутер – область сети)</td></tr>
<tr><td colspan="9">Link Data (IP-адрес маршрутизатор, породившего LSA)</td></tr>
<tr><td colspan="3">Type (тип канала)</td><td colspan="3"># TOS</td><td colspan="3">Metric (метрика канала)</td></tr>
<tr><td colspan="3">TOS</td><td colspan="3">0</td><td colspan="3">TOS metric</td></tr>
<tr><td colspan="9">. . .</td></tr>
<tr><td colspan="9">Link ID (идентификатор объекта, к которому подключен роутер — область сети)</td></tr>
<tr><td colspan="9">Link Data (IP-адрес маршрутизатор, породившего LSA)</td></tr>
<tr><td colspan="9">. . .</td></tr>
</table>

Рис. 5.18. Формат объявления состояния связей router-LSA

Порядковый номер LSA *LS sequence number* служит для обнаружения дубликатов и старых LSA и представляет собой 32- разрядное целое число со знаком в диапазоне $[-N+1, N-1]$, где $N = 2^{31}$. Значение $(-N + 1)$ являет-

ся минимальным значением порядкового номера (самым старым), которое используется при первой генерации LSA. При генерации новых экземпляров LSA порядковый номер увеличивается на 1 до достижения максимального значения ($N - 1$). При достижении максимального значения порядкового номера экземпляр объявления LSA удаляется из домена маршрутизации и генерируется новое LSA с минимальным значением порядкового номера. Для выполнения этой процедуры используется принудительное старение LSA. А именно, в поле *возраст LSA* (*LS age*) экземпляра LSA с максимальным значением порядкового номера ($N - 1$) устанавливается максимальное значение *LS age = MaxAge* и осуществляется повторная рассылка LSA с сохранением максимального порядкового номера. После этого маршрутизатор удаляет данный экземпляр LSA из базы данных и генерирует новый экземпляр с минимальным значением порядкового номера ($-N + 1$).

В поле *Checksum* записывается значение контрольной суммы, вычисляемой для всего содержимого LSA, включая заголовок, но без учета поля возраста LSA.

Поле *Length* — размер LSA в байтах с учетом 20-байтного заголовка LSA.

За заголовком следует тело сообщения LSA, которое содержит описания состояний каналов маршрутизатора в область.

Начинается оно с поля флагов, имеющее три младших значащих бита:

V — этот флаг устанавливается в тех случаях, когда маршрутизатор является конечной точкой виртуального канала, для которого описываемая область является транзитной (*V* – конечная точка виртуального канала);

E — этот флаг устанавливается для граничных маршрутизаторов автономной системы (E — external);

B — этот флаг устанавливается для граничных маршрутизаторов области (B — border).

В поле *# links* указывается число каналов, которое описывается в данном LSA (оно равно количеству каналов маршрутизатора в область).

Описание состояния каждого канала содержит следующие поля.

Type (тип канала) — канал в транзитную сеть, к другому маршрутизатору, в тупиковую сеть или виртуальный канал.

Link ID — идентификатор объекта, к которому подключен маршрутизатор. Содержимое этого поля зависит от типа канала (*Type*). При соединении с объектом, генерирующим LSA (другой маршрутизатор или транзит-

ная сеть) поле *Link ID* содержит значение *Link State ID* из LSA соседа. Во внутридоменных объявлениях *router-LSA* в этом поле указывается идентификатор выделенного в данной области маршрутизатора DR.

Link Data — данные канала. Содержимое этого поля также зависит от типа канала (поле *Type*). Для соединений с тупиковыми сетями поле *Link Data* содержит маску IP, а для остальных типов соединений (за исключением безадресных) — IP-адрес интерфейса маршрутизатора. Это поле обеспечивает последнюю часть информации, требуемой для построения таблицы маршрутов, когда рассчитывается IP-адрес следующего маршрутизатора (next hop).

 # TOS — число различных метрик TOS, заданных для данного соединения, без учета метрики канала по умолчанию. При отсутствии дополнительной метрики TOS это поле имеет значение 0.

Metric — метрика данного канала маршрутизатора. При использовании дополнительных метрик (кроме метрики по умолчанию) в анонс может также включаться дополнительная информация, связанная с TOS. Для этого задействуются поля *TOS* (Type of Service — тип сервиса), к которому относится метрика, и *TOS metric*, содержащее связанную с TOS метрическую информацию. Соответствие кодировок типов сервиса в протоколе OSPF и в поле TOS пакета IP показано в табл. 5.9.

Таблица 5.9

Представление TOS-IP в OSPF

Код OSPF	Значения TOS (RFC 1349)	Критерий качества обслуживания
0	0000	Нормальное обслуживание
2	0001	Минимальная стоимость
4	0010	максимальная надежность
8	0100	Максимальная пропускная способность
16	1000	Минимальная задержка

Далее в теле сообщения LSA следуют описания всех оставшихся каналов маршрутизатора.

Пример объявления состояния связей router-LSA маршрутизатора RT3 для области 1 показан на рис. 5.19. В данном примере предполагается, что

интерфейсы маршрутизатора к сетям N3 и N4 на рис. 5.10 имеют IP-адреса 192.1.1.3 и 192.1.5.3 соответственно. В качестве идентификатора Router ID маршрутизатора RT3 используется меньший из IP-адресов его интерфейсов (192.1.1.3). RT3 имеет два соединения с областью 1: одно в транзитную сеть 192.1.1.0 (N3) и другое в тупиковую сеть 192.1.5.0 (N4). Кроме того, RT3 является граничным маршрутизатором области 1. Транзитная сеть идентифицируется IP-адресом интерфейса выделенного маршрутизатора DR (т. е. Link ID = 192.1.1.4 — адрес интерфейса DR-маршрутизатора RT4 в сеть 192.1.1.0).

LS age = 0	;	= 0 при создании LSA
Options = (E-bit)	;	
LS type = 1	;	Router-LSA
Link State ID = 192.1.1.3	;	Router ID для RT3
Advertising Router = 192.1.1.3	;	Router ID для RT3
bit E = 0	;	не является граничным для AS
bit B = 1	;	граничный роутер области
#links = 2	;	LSA содержит описание 2-х каналов
Link ID = 192.1.1.4	;	IP-адрес интерфейса DR (RT4)
Link Data = 192.1.1.3	;	IP-адрес интерфейса RT3 в сеть
Type = 2	;	соединение с транзитной сетью
# TOS metrics = 0	;	
metric = 1	;	
Link ID = 192.1.5.0	;	IP-номер сети
Link Data = 0xffffff00	;	маска сети
Type = 3	;	соединение с тупиковой сетью
# TOS metrics = 0	;	
metric = 2	;	

Рис. 5.19. Пример объявления состояния связей router-LSA маршрутизатора RT3 для области 1

С описанием форматов объявлений состояния связей (LSA) других типов можно ознакомиться в документе RFC 2328.

5.5. ПРОТОКОЛЫ ВНЕШНЕЙ МАРШРУТИЗАЦИИ. ПРОТОКОЛ BGP

Для выбора маршрутов между автономными системами (рис. 5.20) используются протоколы внешней маршрутизации (протоколы внешних шлюзов), например, BGP. При этом внутренняя структура АС для протокола внешней маршрутизации является неразличимой, а маршрут представляет собой последовательность автономных систем на пути следования к АС назначения. Необходимо заметить, что АС могут быть транзитными (AS1 на рис. 5.20) и оконечными (AS2 и AS3 на рис. 5.20).

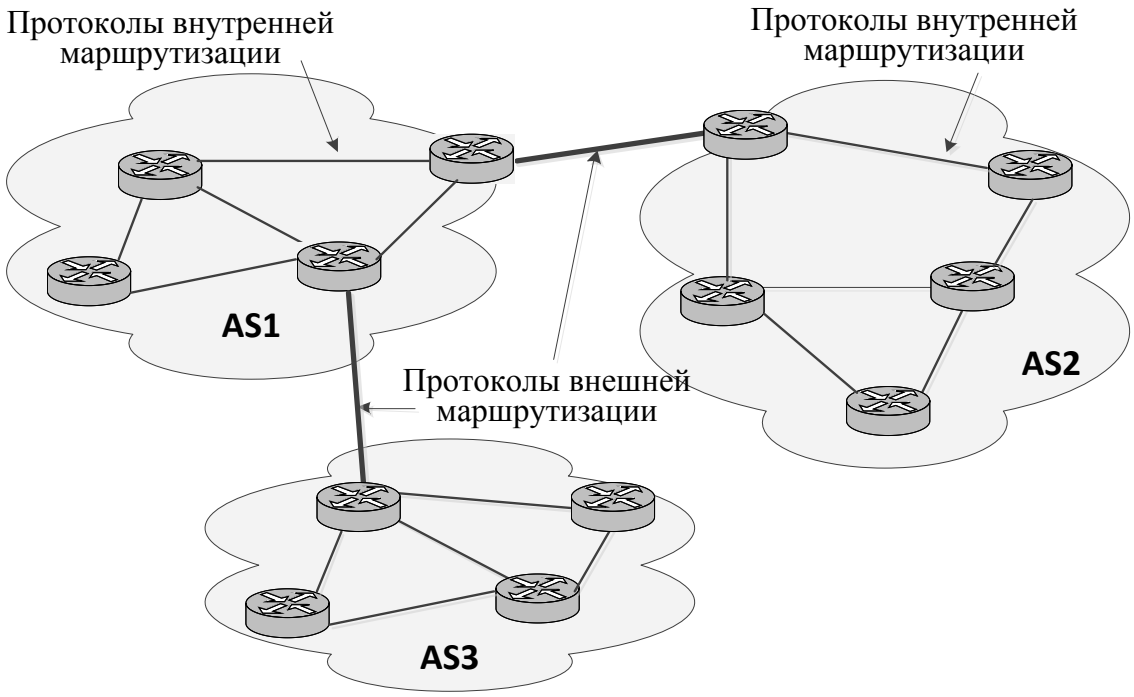


Рис. 5.20. Место протоколов внутренней и внешней маршрутизации

Протоколы внешней маршрутизации могут применяться не только для выбора маршрутов между АС в сети интернет, но и в любых крупных объединенных пакетных сетях.

Протокол BGP (Border Gateway Protocol — протокол граничного шлюза) — это протокол внешней маршрутизации, который был разработан для применения в объединенных сетях TCP/IP, но может применяться и в других пакетных сетях. В настоящее время действует версия протокола

BGP-4, описанная в RFC 1771. Данная версия протокола предусматривает возможность агрегации сетей с использованием технологии бесклассовой междоменной маршрутизации.

Дистанционно-векторная маршрутизация и маршрутизация на основе оценки состояния линий, применяемые в протоколах внутренней маршрутизации, не эффективны для решения задач внешней маршрутизации. В частности, в разных АС могут использоваться различные метрики, что затрудняет разработку работоспособного алгоритма дистанционно-векторной маршрутизации или маршрутизации на основе оценки состояния линий. Кроме того, при использовании протокола маршрутизации на основе состояния линий каждый маршрутизатор пересылает данные о своих линиях с соседями остальным маршрутизаторам сети путем лавинной рассылки. Рассылка этих данных всем маршрутизаторам в разных автономных системах может оказаться невыполнимой. Поэтому протокол BGP, в отличие от дистанционно-векторного протокола RIP и протокола OSPF, работающего на основе оценки состояния линий, использует концепцию маршрутно-векторной маршрутизации. Протокол BGP воспринимает объединенную сеть в виде графа, состоящего из АС (или сетей) в вершинах графа и магистральных линий (ребер графа) между ними.

Сравнение маршрутно-векторной и дистанционно-векторной маршрутизации приведено в табл. 5.9.

Протокол работает с помощью сообщений *Открытие* (*Open*), *Подтверждение* (*Keep alive*), *Обновление* (*Update*) и *Уведомление* (*Notification*), посылаемых по TCP-соединениям (порт 179). При обмене вышеназванными сообщениями между маршртутизаторами сети реализуются следующие основные процедуры протокола BGP-4:

- приобретение соседей;
- проверка доступности соседей;
- проверка доступности сетей.

Приобретение соседей. Соседи — это маршрутизаторы, присоединенные к одной и той же сети, или маршрутизаторы разных АС, согласные обмениваться маршрутной информацией по протоколу BGP. Согласие определяется администратором сети, а также загруженностью маршрутизатора. Процедура реализуется путем обмена сообщениями *Открытие* и *Подтверждение*.

Таблица 5.9

Сравнение маршрутно-векторной и дистанционно-векторной маршрутизации

Маршрутно-векторная маршрутизация	Дистанционно-векторная маршрутизация
Информация о маршруте включает адрес сети-назначения и перечень идентификаторов всех АС на пути к данной сети.	Информация о маршруте включает адрес сети-назначения без перечня узлов на пути к данной сети.
Информация о маршруте не включает метрики пути (расстояния, стоимости).	Информация о маршруте включает метрику пути.
Информация об обновлении маршрута передается только при изменениях в сети и только маршрутизаторам, объявленным как соседи.	Информация об обновлении маршрутной таблицы передается периодически соседним маршрутизаторам.
Возможность осуществления политики маршрутизации за счет явного указания перечня АС на пути к сети-назначения.	—

Проверка доступности соседей. Данная процедура реализуется периодической посылкой сообщения *Подтверждение* сотрудничества.

Проверка доступности сетей. Эта процедура реализуется рассылкой сообщений об обновлении базы данных доступных сетей и маршрутов к ним маршрутизаторам-соседям (сообщение *Обновление*). Каждый маршрутизатор поддерживает базу данных доступных ему сетей и маршрутов к ним. Когда в базе данных происходят изменения, маршрутизатор рассылает сообщения об обновлении другим маршрутизаторам, взаимодействующим с ним по протоколу BGP. Таким образом, протокол BGP не требует периодического обновления всей таблицы маршрутизации BGP.

Каждое сообщение протокола BGP начинается с 19-байтного заголовка (рис. 5.21), содержащего три поля:

- маркер — зарезервировано для аутентификации;

- длина сообщения в байтах;
- тип сообщения.

16 байт	2 байта	1 байт	
Маркер	Длина сообщения	Тип сообщения	Поле данных сообщения

Рис. 5.21. Общий формат сообщений протокола BGP

Максимальный размер сообщения составляет 4096 байтов. Минимальное сообщение (например, *Подтверждение*) содержит только BGP-заголовок (19 байтов).

19 байт	Заголовок
1 байт	Версия протокола
2 байта	Моя автономная система
2 байта	Время удержания
4 байта	Идентификатор BGP
1 байт	Длина дополнительных параметров
	Дополнительные параметры

Рис. 5.22. Формат сообщения *Открытие*

Чтобы приобрести соседа, маршрутизатор сначала устанавливает TCP-соединение с интересующим его другим BGP-маршрутизатором. Затем он посылает сообщение *Открытие* о предложении сотрудничества, минимальная длина которого составляет 29 байт (рис. 5.22). Это сообщение содержит следующие поля:

- версия (Version) — беззнаковое целое число, указывающее номер версии протокола BGP;
- моя АС (My Autonomous System) — номер автономной системы маршрутизатора-отправителя сообщения *Открытие*;
- время удержания (Hold Time) — максимальный интервал времени (в секундах) между передачей последовательных сообщений *Подтверждение* и/или *Обновление*, предлагаемый маршрутизатором-отправителем сообщения *Открытие*. Взаимодействующие маршрутизаторы выбирают меньшее

из предлагаемых ими друг другу значений времени удержания, которое должно быть не менее 3 с, если значение Hold Time не равно 0);

- идентификатор BGP (BGP Identifier) — IP-адрес маршрутизатора-отправителя сообщения OPEN, присвоенный этому узлу BGP;

- длина дополнительных параметров (Optional Parameters Length) — поле показывает размер в байтах поля дополнительных параметров, которое может следовать за ним, если *Optional Parameters Length* не равно 0;

- поле дополнительных параметров (Optional Parameters) — поле содержит список и описание дополнительных (значение) параметров.

В частности, последние два поля могут использоваться для решения задач аутентификации. Так, в RFC 1771 определен такой дополнительный параметр, как *Authentication Information* (тип 1, данные аутентификации). В этом случае в поле *Optional Parameters Length* записывается код аутентификации (*Authentication Code*), задающий механизм аутентификации, а поле дополнительных параметров содержит данные аутентификации (*Authentication Data*).

Сообщение *Подтверждение* состоит только из заголовка. Взаимодействующие маршрутизаторы посылают друг другу это сообщение не позднее, чем истечет выбранный интервал времени удержания. При установке Hold Time = 0 сообщения *Подтверждение* не передаются.

Сообщения *Обновление* используются для передачи маршрутной информации между узлами BGP. Данные из пакетов UPDATE используются для построения графа, описывающего связи между различными АС. Сообщения об обновлении могут содержать маршрутные данные двух типов:

- список отмененных (удаляемых) маршрутов, о которых ранее объявлял данный маршрутизатор;

- информацию о доступном маршруте через объединенную сеть, которая может быть добавлена в базы данных других узлов.

Сообщение *Обновление* может одновременно анонсировать доступный маршрут и отзывать группу недоступных более маршрутов. Формат его показан на рис. 5.23.

Информация об обновлении первого типа касается удаления одного или нескольких маршрутов. В поле длины списка отменяемых маршрутов указывается размер поля отмененных маршрутов (*Withdrawn Routes*) в байтах. Нулевое значение этого поля говорит об отсутствии отменяемых мар-

шрутов и поля отмененных маршрутов *Withdrawn Routes* в сообщении UPDATE.

19 байт	Заголовок
2 байта	Длина списка отмененных маршрутов
	Отмененные маршруты
2 байта	Полная длина списка атрибутов пути (TPAL)
	Атрибуты пути
	Информация о доступности сетевого уровня (NLRI)

Рис. 5.23. Формат сообщения *Обновление*

Поле отменяемых маршрутов имеет переменный размер и содержит список префиксов IP-адресов сетей, маршруты к которым исключаются из обслуживания. Это поле содержит два компонента: *Length* (1 байт) и *Prefix* (переменный размер). В подполе *Length* указывается длина префикса IP-адреса в битах. Нулевое значение подполя *Length* соответствует всем IP-адресам. Подполе *Prefix* (префикс) содержит префикс IP-адреса, за которым следует несколько битов, используемых для выравнивания по границе октета.

Информация первого типа об одном доступном маршруте через объединенную сеть описывается в трех полях: TPAL (*Total Path Attribute Length* — полная длина списка атрибутов пути), *Path Attributes* (атрибуты пути) и NLRI (*Network Layer Reachability Information* — информация о доступности сетевого уровня). В поле TPAL указывается общая длина следующего за ним поля атрибутов пути (*Path Attributes*) в байтах. Значение 0 говорит об отсутствии поля NLRI в данном сообщении UPDATE.

Поле атрибутов пути содержит список всех атрибутов конкретного маршрута, интерпретация которых зависит от значений флагов, также содержащихся в данном поле.

Стандартом определены следующие типы атрибутов:

- Origin (происхождение) — указывает на способ формирования информации о пути, т. е. о доступности сетевого уровня NLRI (с помощью протоколов внутренней (IGP) или внешней маршрутизации (EGP), или же другим способом);

- AS_Path — список (упорядоченный или неупорядоченный) всех автономных систем на данном маршруте;

- Next_Hop — IP-адрес граничного BGP-маршрутизатора, который следует использовать в качестве следующего маршрутизатора на пути к адресатам, перечисленным в поле NRLI;

- Multi_Exit_Disc — положительное число, определяющее степень предпочтения маршрута при наличии двух точек входа в АС;

- Local_Pref (локальные предпочтения) — используется для информирования других BGP-узлов о степени предпочтения объявляемого маршрута (этот атрибут не имеет значения для BGP-маршрутизаторов других автономных систем).

Атрибуты *Atomic_Aggregate* и *Aggregator* реализуют концепцию агрегирования маршрутов в иерархических сетях.

Поясним назначение атрибута *Multi_Exit_Disc*. Этот атрибут используется на внешних (между АС) каналах для выбора точки входа/выхода из данной АС в соседнюю АС при наличии нескольких точек входа/выхода в неё. Значение атрибута представляет собой метрику внутреннего маршрута от граничного маршрутизатора до адресата в соседней автономной системе. Значения данной метрики для маршрутов от всех точек входа в соседнюю автономную систему до адресата определяются применяемым в этой АС протоколом внутренней маршрутизации (например, OSPF). Граничный маршрутизатор соседней АС передает значения метрик внутренних маршрутов в данную АС, на основе чего BGP-маршрутизаторы данной АС могут осуществить выбор точки входа/выхода в соседнюю АС.

Поле информации о доступности сетевого уровня (NRLI) имеет переменный размер и включает список префиксов IP-адресов сетей, к которым можно получить доступ с помощью данного маршрута. Как и поле отмененных маршрутов поле NRLI содержит два компонента: *Length* (1 байт) и *Prefix* (переменный размер). В подполе *Length* указывается длина префикса IP-адреса в битах. Нулевое значение подполя *Length* соответствует всем IP-адресам. Подполе *Prefix* содержит префикс IP-адреса сетей, достижимых по данному маршруту, за которым следует несколько битов, используемых для выравнивания по границе октета.

В случае обнаружения ошибок передается сообщение *Уведомление*. Формат сообщения показан на рис. 5.25. Расшифровка кодов ошибок дана в табл. 5.10.

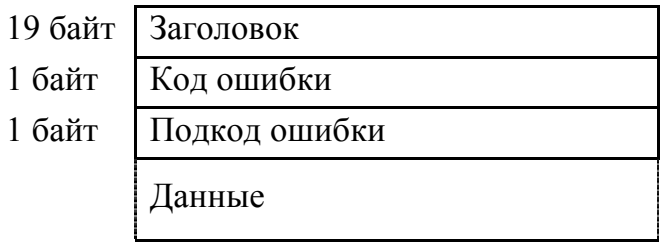


Рис. 5.25. Формат сообщения *Уведомление*

Таблица 5.10

Описание кодов ошибок в сообщении *Уведомление*

Код ошибки	Описание
1	Ошибка в заголовке сообщения
2	Ошибка в сообщении OPEN
3	Ошибка в сообщении UPDATE
4	Истекло время поддержания
5	Ошибка конечного автомата
6	Прерывание (для разрыва соединения)

Работу протокола BGP рассмотрим в упрощенном виде на примере объединенной сети, представленной на рис. 5.26. Сплошными стрелками показано взаимодействие (обмен маршрутной информацией) смежных маршрутизаторов по протоколу BGP, пунктирными стрелками – взаимодействие внутренних маршрутизаторов в каждой автономной системе по протоколу OSPF. Сначала маршрутизаторы внутри AS (например, R1, R2, R3 в AS1), обмениваясь между собой маршрутной информацией с помощью протокола OSPF, собирают полную информацию о внутренней топологии AS и формируют таблицы маршрутизации, которые в дальнейшем используются для выбора путей внутри AS. Затем маршрутизатор R1 с помощью протокола BGP может отправить маршрутное сообщение об об-

новлении маршрутизатору R4 в автономной системе AS2, содержащее следующие поля:

AS_Path – идентификатор автономной системы AS1;

Next Hop – IP-адрес маршрутизатора R4;

NLRI – список всех сетей в автономной системе AS1.

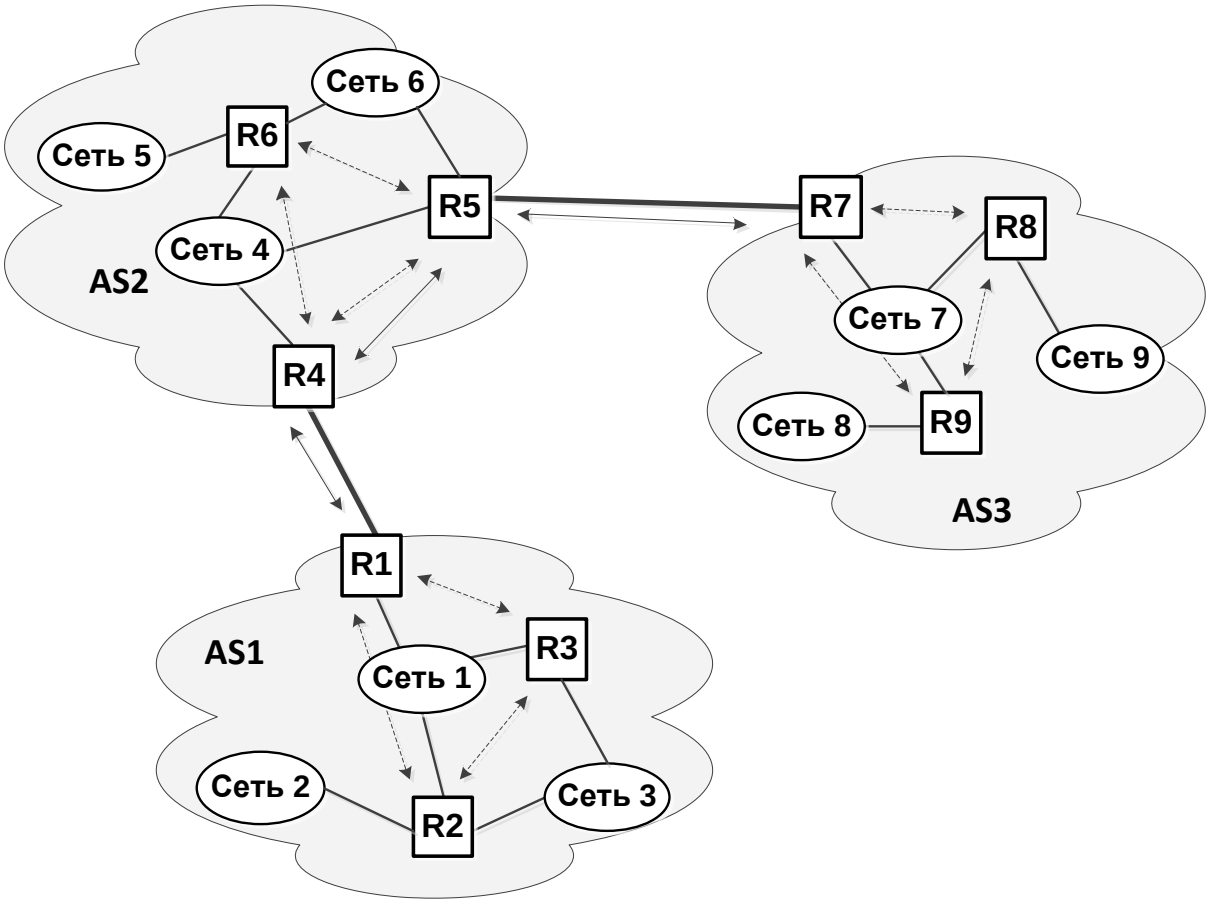


Рис. 5.26. Пример объединенной сети

Это сообщение информирует маршрутизатор R4 о том, что ко всем перечисленным в поле *NRLI* сетям можно получить доступ через маршрутизатор R1, и для этого необходимо пересечь одну автономную систему – AS1. Маршрутизатор R4 переправит полученное от R1 по протоколу BGP маршрутное сообщение смежному с ним граничному BGP-маршрутизатору R5 своей автономной системы (AS2), которое в поле *Next Hop* будет содержать IP-адрес маршрутизатора R4. У маршрутизатора R5 установлены соседские отношения с BGP-маршрутизатором R7 другой автономной системы – AS3. Он маршрутное сообщение об обновлении мар-

шрутной информации маршрутизатору R7, которое будет включать следующие поля:

AS_Path – идентификаторы автономных систем {AS1, AS2};

Next Hop – IP-адрес маршрутизатора R5;

NLRI – список всех сетей в автономной системе AS1.

Это сообщение информирует маршрутизатор R7 о том, что ко всем перечисленным в поле *NRLI* сетям можно получить доступ через маршрутизатор R5, и для этого необходимо пересечь две автономных системы – AS1 и AS2. Если у маршрутизатора R7 есть альтернативный маршрут к некоторым сетям, перечисленным в предлагаемом маршруте через R5, то он должен сделать выбор между этими маршрутами. Альтернативным, к примеру, мог бы быть маршрут через другую точку входа в AS2, либо через другую автономную систему, имеющую выход в AS1, чего в рассматриваемом примере нет. Выбор предпочтения при наличии нескольких объявленных маршрутов осуществлялся бы на основе сравнения метрик внутренних маршрутов, передаваемых в AS3 значением атрибута *Multi_Exit_Disc*.

После выбора предпочтительного маршрута маршрутизатор R7 помещает полученные маршрутные данные в свою базу данных маршрутизации и переправляет маршрутную информацию смежным с ним BGP-маршрутизаторам. Передаваемое им маршрутное сообщение будет содержать поле *AS_Path* с перечнем идентификаторов пересекаемых на данном маршруте автономных систем {AS1, AS2, AS3}.

Подобным образом маршрутная информация распространяется по всей объединенной сети, состоящей из множества автономных систем. При этом поле *AS_Path* используется для предотвращения зацикливания. А именно, если маршрутное сообщение об обновлении получено маршрутизатором из автономной системы, включенной в список *AS_Path* , то этот маршрутизатор не пересылает обновленную маршрутную информацию другим маршрутизаторам.

ВОПРОСЫ И ЗАДАНИЯ К ГЛАВЕ 5

1. Имеется ли вероятность того, что пакет будет доставлен по неверному адресу, если все маршрутизаторы и хосты сети работают нормально?

2. Какая разница между доменом и автономной системой?

3. Какие метрики маршрутов используются в протоколах маршрутизации?

4. Поясните отличия между дистанционно-векторной, маршрутно-векторной маршрутизацией и маршрутизацией на основе оценки состояния связей.

5. Какие механизмы предусмотрены в протоколе RIP для преодоления проблемы счета до бесконечности и ускорения распространения по сети маршрутной информации?

6. Составьте таблицу RIP-маршрутизации для маршрутизатора A на рис. 5.6. Предполагается, что все метрики интерфейсов маршрутизаторов в сети равны 1, а метрики интерфейсов из сетей в маршрутизаторы равны 0.

7. Поясните достоинства и недостатки иерархической маршрутизации.

8. Для чего в протоколе OSPF используется концепция выделенного маршрутизатора?

9. Каковы отличия между таблицами маршрутизации протоколов RIP и OSPF?

10. Какие типы маршрутов бывают в сетях с OSPF-маршрутизацией?

11. Являются ли идентичными базы данных OSPF-маршрутизаторов разных областей сети. Если нет, то почему?

12. Каким образом и какая маршрутная информация протокола OSPF передается в данную область из других областей сети и из других автономных систем?

13. Посылаются ли в протоколе BGP периодические обновления маршрутной информации, как это делает RIP?

14. Где в BGP переносится информация о номерах автономной системы?

6. АРХИТЕКТУРА, ПРОТОКОЛЫ И ПРИНЦИПЫ ПОСТРОЕНИЯ СЕТЕЙ MPLS

В настоящее время для построения транспортных сетей используются различные технологии и протоколы физического (PDH, SDH, WDM и др.), канального (Frame Relay — FR, PPP, 10 и 40 Gigabit Ethernet, MPLS), сетевого (IP) уровней, а также технологии, охватывающие несколько уровней ЭМВОС, такие, как X.25 и ATM. При этом могут использоваться различные варианты межуровневого сочетания перечисленных технологий. Например, IP/MPLS/10GE/SDH/DWDM, когда протокол IP работает поверх MPLS, пакеты MPLS переносятся в кадрах 10 Gigabit Ethernet, которые, в свою очередь, передаются на физическом уровне по каналам и трактам, образованным системами передачи синхронной цифровой иерархии (SDH) спектрального мультиплексирования (WDM). Выбор того или иного варианта определяется различными факторами. Для построения высокоскоростных магистральных сетей в последние годы широко используется *технология многопротокольной коммутации по меткам* (MPLS) в стеке IP/MPLS/10GE/SDH/DWDM [6, 8].

Независимо от применяемых технологий и протоколов для построения транспортных сетей важнейшими условиями являются обеспечение требуемых значений пропускной способности, устойчивости и качества передачи. В таких пакетных технологиях транспортных сетей, как X.25, Frame Relay, ATM, MPLS это в значительной мере достигается за счет использования техники виртуальных каналов.

6.1. ОБЩАЯ ХАРАКТЕРИСТИКА ТЕХНОЛОГИИ MPLS

MPLS (Multiprotocol Label Switching) — это технология коммутации пакетов в многопротокольных сетях, основанная на использовании меток. MPLS позиционируется как способ построения высокоскоростных магистралей.

Традиционно главными требованиями, предъявляемыми к технологиям магистральной сети, были высокая пропускная способность, малое значение задержки и хорошая масштабируемость. Архитектура MPLS обеспечивает построение магистральных сетей, имеющих практически неограниченные возможности масштабирования, повышенную скорость обработки

трафика и высокую гибкость с точки зрения организации дополнительных сервисов.

История создания технологии MPLS началась в середине 90-х гг. с попыток объединения достоинств технологий IP и ATM: гибкости, обеспечиваемой маршрутизацией пакетов в каждом узле IP-сети, и надежности доставки пакетов с малыми задержками, обеспечиваемой техникой виртуальных каналов, используемой в ATM. Первым продуктом на рынке стала IP-коммутация, разработанная компанией Ipsilon. Появились и другие разработки в этой области таких компаний, как Cisco Systems, IBM и др.

Основная идея состояла в том, чтобы обработку большей части трафика в узлах сети переместить с сетевого уровня вниз (по стеку протоколов OSI) и, тем самым, повысить производительность сетевых устройств и скорость продвижения пакетов в сети. В традиционных IP-сетях каждый маршрутизатор извлекает из поступившего пакета заголовок, в котором содержится IP-адрес назначения и информация о требуемом качестве обслуживания в поле TOS или DS-байт. После его обработки в соответствии с заданным качеством обслуживания и таблицей маршрутизации независимо от других маршрутизаторов осуществляется выбор дальнейшего пути следования данного пакета. Таким образом, каждый маршрутизатор IP-сети обрабатывает значительное количество информации, на что затрачивается соответствующее время.

Процедуру соотнесения пакета тому или иному классу обслуживания (FES — Forwarding Equivalence Classes, класс эквивалентного продвижения) и выбор пути следования пакета до адресата можно выполнять не во всех промежуточных узлах, а только на входе в сеть. При этом в промежуточных узлах сети можно производить обработку не всего IP-заголовка (20 байт), а короткой метки, присвоенной пакету, что позволяет существенно уменьшить время обработки пакетов в узлах и ускорить их продвижение по сети.

Архитектура устройства, осуществляющего продвижение данных по меткам и называемого в технологии MPLS «маршрутизатор, коммутирующий по меткам» (Label Switching Router — LSR), показана на рис. 6.1.

LSR, наряду с традиционными функциональными модулями IP-маршрутизатора (модуль протоколов маршрутизации, таблица маршрутизации, модуль продвижения данных), содержит дополнительные блоки,

обеспечивающие продвижение данных с помощью меток. Это — модуль сигнального протокола, осуществляющего прокладку пути коммутации по меткам, таблица коммутации и модуль продвижения данных по меткам.

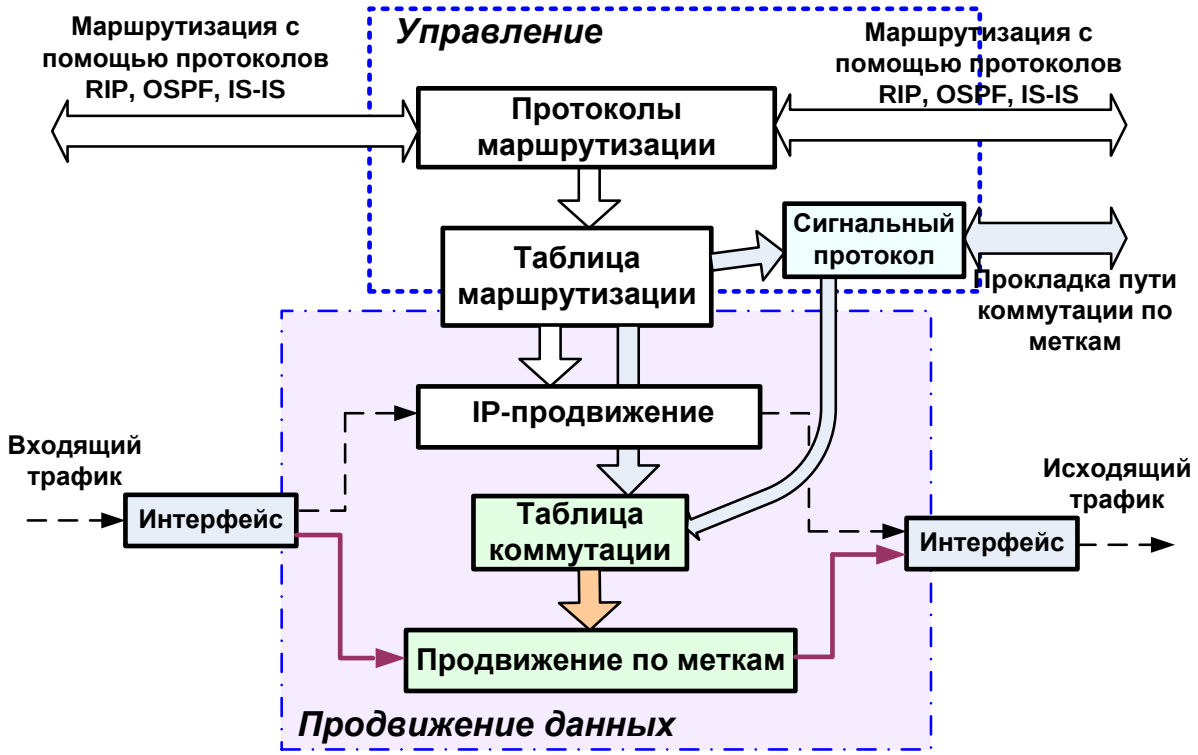


Рис. 6.1. Архитектура маршрутизатора, коммутирующего по меткам

Видно, что LSR объединяет функции IP-маршрутизатора и коммутатора по меткам, так как он поддерживает два механизма продвижения данных: дейтаграммный на основе IP-адресов и механизм виртуальных каналов (с установлением соединения). Протоколы маршрутизации используются для построения таблиц IP-маршрутизации, на основе которых создаются таблицы коммутации по меткам (MPLS-продвижения).

Технология многопротокольной коммутации по меткам занимает свое место между 2-м и 3-м уровнями (рис. 6.2) модели эталонной модели взаимодействия открытых систем.

Перечислим отличительные особенности и области применения MPLS.

1. MPLS — это технология, ориентированная на соединение (использует технику виртуальных каналов и путей), на основе чего обеспечивается

поддержка гарантированного качества обслуживания, в отличие от дейтаграммных сетей.

2. Имеет место многопротокольная поддержка.

3. Технология MPLS поддерживает построение виртуальных частных сетей.

4. Имеется возможность конструирования трафика (TE — Traffic Engineering).

Рис. 6.2. Место MPLS в модели OSI

Обеспечение гарантированного качества обслуживания на основе передачи с установлением соединения. Сеть, использующая механизм передачи без установления соединения, например, IP-сеть, не может обеспечить действительно гарантированного качества обслуживания. Известно, что для обеспечения качества обслуживания в IP-сетях разработаны и постоянно совершенствуются такие механизмы, как дифференцированное (difserv) и интегрированное (intserv) обслуживание. Однако, архитектура дифференцированных служб работает только с агрегированным трафиком от нескольких источников (классами трафика), не учитывая и не реализуя индивидуальных требований к качеству обслуживания. Архитектура интегрированных служб, использующая протокол RSVP резервирования ресурсов, напоминает подход с установлением соединения, но плохо приспособлена для работы в больших объединенных сетях. Для таких услуг, как голос и видео, требующих сетей с высокой предсказуемостью, подходы,

характерные для дифференцированного и интегрированного обслуживания, в сильно загруженных сетях оказываются неадекватными.

Напротив, ориентированная на соединение сеть обладает мощными средствами управления трафиком и предоставления гарантий обслуживания с различными уровнями качества. Архитектура MPLS накладывает на объединенную IP-сеть структуру, ориентированную на соединение и, таким образом формирует основу для обеспечения гарантий качества обслуживания различных видов трафика.

Многопротокольная поддержка. Архитектура MPLS может использоваться в сетях с различными сетевыми технологиями и технологиями канального уровня (IP, ATM, Frame Relay, Ethernet, PPP и др.). Она может использоваться как в «чистых» сетях IP, ATM, FR, так и в объединенной сети, построенной на базе двух, и более технологий. Именно в этой универсальности совместного использования с другими технологиями заключается многопротокольность архитектуры MPLS, полезная для решения задач оптимизации ресурсов и поддержки различных уровней качества в смешанных сетях.

Поддержка виртуальных частных сетей. Технология MPLS может эффективно использоваться для построения в объединенной сети виртуальных частных сетей на основе разграничения трафика (VPN-MPLS). В виртуальной частной MPLS-сети её трафик отделен от трафика других виртуальных частных сетей, что предоставляет гарантии безопасности и качества обслуживания. Делается это с применением техники виртуальных путей.

Конструирование (*инжиниринг*) *трафика.* Конструирование трафика — это совместное решение задач определения маршрутов прохождения потоков трафика по сети, оптимизации использования ресурсов сети и обеспечения требуемого качества обслуживания для каждого потока.

Протоколы IP-сетей поддерживают примитивные формы автоматизированного конструирования трафика. К примеру, протокол маршрутизации OSPF позволяет маршрутизаторам в целях балансирования нагрузки динамически менять маршруты пакетам данного адресата. Но, во-первых, маршрутизаторы IP-сети оперируют отдельными пакетами, выбирая кратчайшие пути для каждого пакета. Поэтому для пакетов различных потоков данному получателю или группе получателей одной подсети (локальной

сети) в объединенной сети, как правило, выбирается один маршрут, что ведет к неоптимальному распределению и использованию ресурсов сети и, возможно, к перегрузке на данном маршруте. Весь трафик между двумя конечными точками следует по одному маршруту, который может быть изменен только в случае перегрузки сети. Во-вторых, не обеспечивается обслуживание с разными уровнями качества.

Архитектура MPLS оперируют не только с отдельными пакетами, но и с потоками пакетов, у каждого из которых есть определенные требования к качеству обслуживания и предсказуемые потребности в ресурсах по передаче трафика. В MPLS-сети возможен выбор маршрутов для отдельных потоков, причем потоки, связывающие одну пару конечных точек, могут следовать по разным маршрутам. При возникновении перегрузки маршруты потоков в MPLS-сети могут быть разумно изменены с учетом требований к качеству обслуживания трафика и оптимального распределения и использования ресурсов сети. Конструирование трафика может существенно повысить пропускную способность сети и обеспечить требуемое качество обслуживания за счет предотвращения перегрузок.

6.2. АРХИТЕКТУРА MPLS-СЕТИ. ПРИНЦИП КОММУТАЦИИ ПО МЕТКАМ

Архитектура MPLS-сети, основные элементы которой представлены на рис. 6.3, описана в RFC 3031. На рис. 6.3 сеть MPLS взаимодействует с несколькими IP-сетями, которые могут не поддерживать технологию MPLS.

Сеть MPLS включает две области: *ядро сети*, в которой находятся магистральные маршрутизаторы, коммутирующие по меткам (LSR — Label Switch Router), а также *граничную область*, в которой находятся — пограничные маршрутизаторы, коммутирующие по меткам (LER — Label Edge Router).

Сеть MPLS может включать несколько доменов. MPLS-домен — это группа соединённых маршрутизаторов, осуществляющих коммутацию по меткам, находящихся под единым административным управлением и функционирующих в соответствии с единой политикой маршрутизации. MPLS домен образуется LSR-маршрутизаторами, а на границе домена размещаются устройства LER.

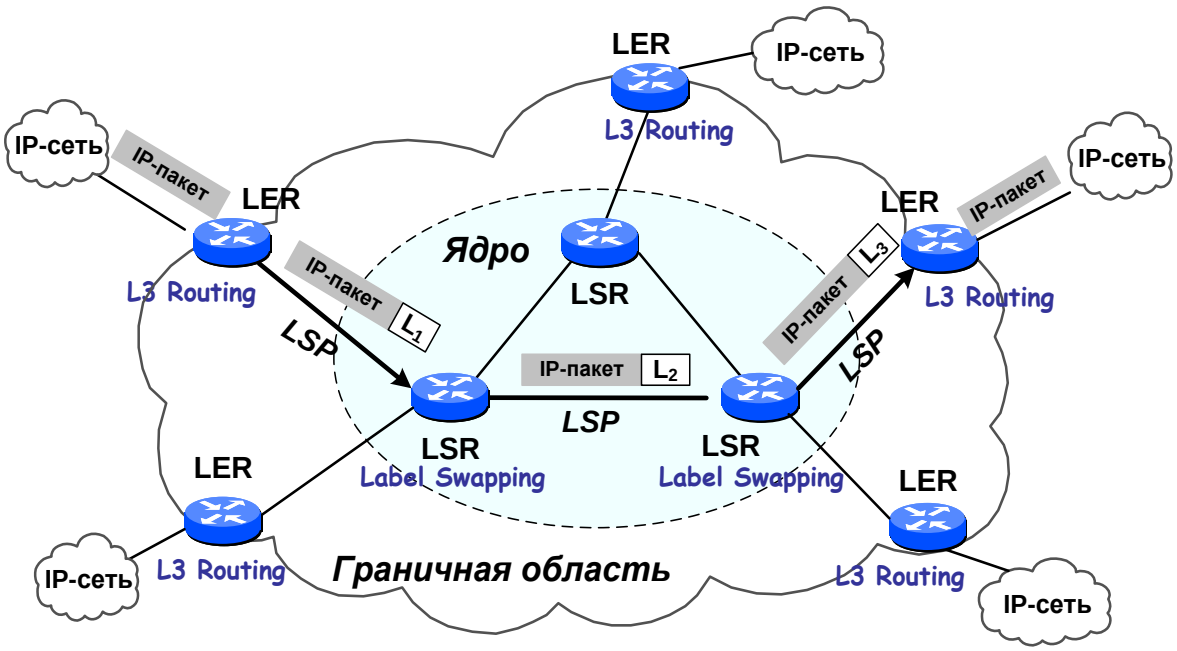


Рис. 6.3. Архитектура сети MPLS

Маршрутизаторы LER и LSR выполняют различные функции. Пограничный маршрутизатор LER с одной стороны взаимодействует с IP-сетью, как обычный маршрутизатор. С другой — с сетью MPLS, как маршрутизатор, коммутирующий по меткам. При поступлении на его вход IP-пакета со стороны IP-сети LER назначает ему метку, добавляет её к IP-пакету (рис. 6.3) и отправляет полученный MPLS-пакет в сеть MPLS в соответствии с таблицей коммутации, поддерживаемой в LER, по предварительно проложенному виртуальному *пути коммутации по меткам* (LSP — Label Switch Path).

Через промежуточные маршрутизаторы LSR продвижение MPLS-пакета к выходному пограничному маршрутизатору LER осуществляется не на основе IP-адреса назначения, а на основе короткой метки. При этом метка имеет локальное значение в пределах каждого ретрансляционного участка между двумя маршрутизаторами, коммутирующими по меткам. В каждом очередном на LSP-пути маршрутизаторе LSR производится смена значения метки (Label swapping). Таким образом, LSR в отличие от LER выполняет только продвижение MPLS-пакета по меткам в соответствии с хранящейся в нем таблицей коммутации.

Путь коммутации по меткам представляет собой однонаправленный виртуальный канал. Поэтому для установления соединения между вход-

ным и выходным MPLS-узлами (входным и выходным LER) необходимо создать два LSP — по одному в каждом направлении.

Пограничный маршрутизатор (LER) выходного MPLS-узла удаляет метку и отправляет IP-пакет в IP-сеть назначения в соответствии с поддерживаемой им таблицей IP-маршрутизации.

Виртуальный канал — это предварительно проложенный фиксированный маршрут следования пакетов, соединяющий конечные узлы в сети с коммутацией пакетов.

Каким образом поступающие на вход пограничного маршрутизатора IP-пакеты передаются в тот или иной LSP? Каждый поток IP-пакетов на входе LER соотносится с определенным FEC-классом (FEC — класс эквивалентного продвижения). FEC-класс — это поток пакетов, имеющих одни и те же требования к условиям их транспортировки. Все пакеты, принадлежащие одному FEC-классу продвигаются через MPLS-сеть по одному маршруту.

FEC-класс определяется по одному или нескольким признакам, указанным сетевым администратором, среди которых могут быть:

- IP-адрес назначения/отправителя (для каждого префикса сети назначения создается свой FEC-класс);
- код дифференцированной службы;
- номера портов отправителя/получателя;
- признак виртуальной сети (для каждой VPN создается отдельный FEC-класс);
- MAC-адрес назначения;
- тип приложения (речевому трафику и трафику передачи данных назначаются разные FEC-классы).

Для каждого FEC-класса предварительно устанавливается отдельный LSP, а также параметры качества обслуживания вдоль этого пути, такие, как объем выделяемых ресурсов, политика организации очередей и отбрасывания пакетов на каждом LSR-маршрутизаторе для данного LSP (пакетов данного FEC-класса). Указанные задачи решаются с применением протоколов двух типов:

- протокола внутренней маршрутизации (к примеру, OSPF-TE — расширение OSPF для MPLS);

- протокола распределения меток LDP (Label Distribution Protocol) или усовершенствованной версии протокола резервирования ресурсов — RSVP-TE (расширение RSVP для MPLS).

Протокол маршрутизации определяет топологию сети, производит текущую оценку связей и строит таблицу маршрутизации. Протокол распределения меток с использованием таблицы маршрутизации осуществляет прокладку путей LSP (распределение меток между соседними LSR) и резервирование ресурсов промежуточных маршрутизаторов (LSR) на пути LSP для данного FEC-класса. Заметим, что сетевой оператор может явно указать маршруты и назначить им соответствующие метки.

На рис. 6.4 более детально иллюстрируется назначение, обработка меток и продвижение MPLS-пакета по LSP.

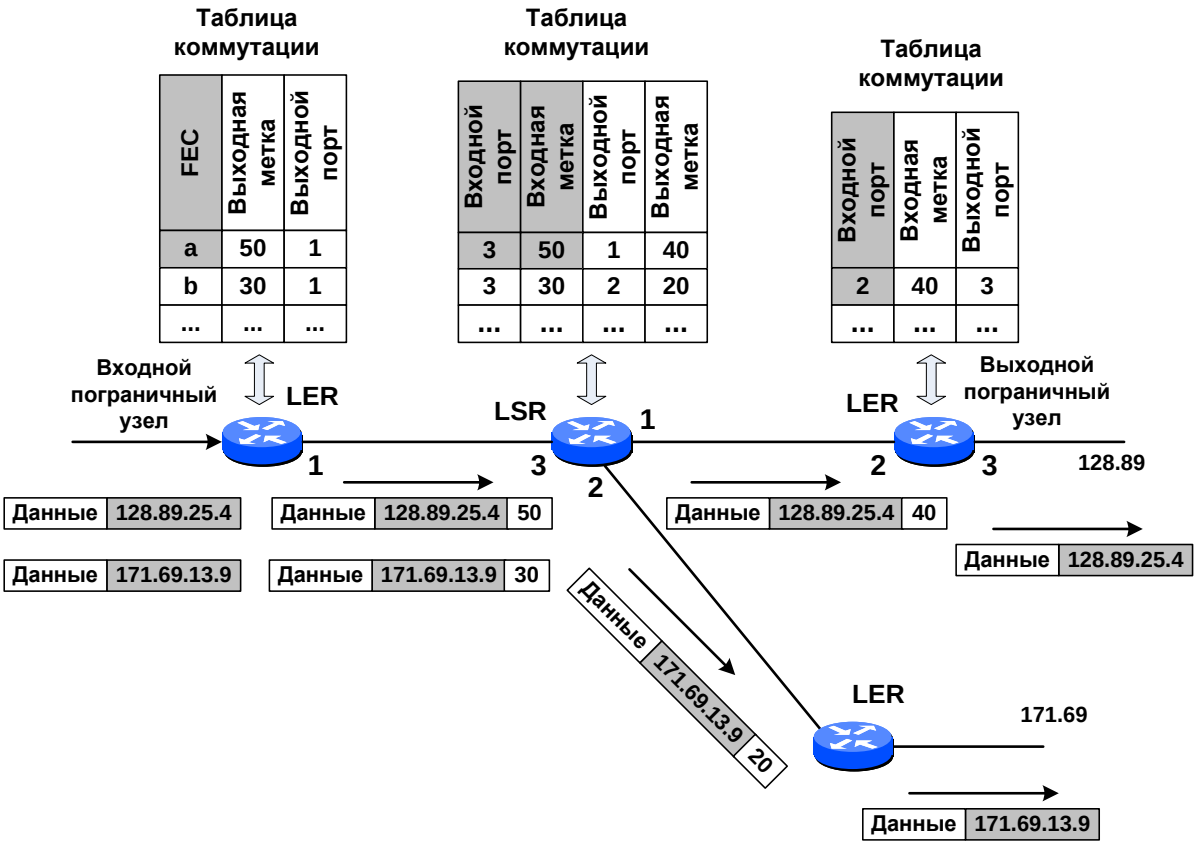


Рис. 6.4. Продвижение MPLS-пакетов

1. При поступлении IP-пакета в MPLS-домен он обрабатывается во входном пограничном маршрутизаторе (LER). LER соотносит пакет с определенным FEC-классом и на основе этого класса назначает пакету опре-

деленный LSP-путь, добавляет к пакету соответствующую метку и продвигает его к следующему LSR на LSP-пути. Если для данного FEC-класса ещё не существует путь LSP, то пограничный маршрутизатор, взаимодействуя с другими LSR маршрутизаторами, должен проложить новый LSP.

2. Каждый последующий LSR-маршрутизатор на данном LSP-пути получив меченый MPLS-пакет выполняет замену метки и переправляет этот пакет следующему LSR-маршрутизатру на LSP-пути.

3. Каждый LSR-маршрутизатор в MPLS-домене поддерживает таблицу продвижения данных (таблицу коммутации) для LSP-путей, проходящих через данный LSR-маршрутизатор. Когда прибывает помеченный пакет, LSR-маршрутизатор просматривает таблицу коммутации, чтобы определить следующий ретрансляционный участок (номер выходного порта) и новое значение метки. Далее LSR-марщрутизатор удаляет из пакета входную метку, присоединяет к нему соответствующую выходную метку и продвигает пакет на соответствующий выходной интерфейс.

4. Пограничный LER выходного MPLS-узла удаляет метку, анализирует заголовок IP-пакета и переправляет IP-пакет по IP-сети получателю.

Рассмотрим структуру заголовка MPLS-пакета (метки).

Заголовок MPLS-пакета имеет длину 32 бита и состоит из следующих полей (рис. 6.5):

- метка (20 бит) — локальное значение метки на данном ретрансляционном участке LSP, по которому осуществляется продвижение пакета;

- класс услуги (3 бита) — это поле было зарезервировано дл экспериментов, но может использоваться для указания класса трафика, дисциплину его обслуживания в очередях, содержать информацию дифференцированных служб;

- S-бит — признак дна стека меток, устанавливаемый в единицу для самой ранней записи в стеке и в 0 — для всех остальных более поздних (понятие стека меток дается ниже);

- время жизни (8 бит) — это поле дублирует соответствующее поле IP-пакета и ограничивает время пребывания MPLS-пакета в сети максимальным количеством пройденных ретрансляционных участков или времени пребывания пакета в сети.

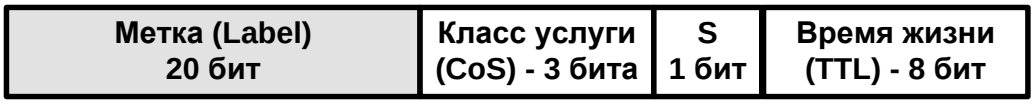


Рис. 6.5. Формат заголовка MPLS-пакета

Как уже указывалось ранее, многопротокольная технология MPLS может взаимодействовать с различными технологиями, используемыми на канальном уровне, к примеру, технологиями канального уровня PPP, Ethernet и др. (рис. 6.6, *а*) или сетевыми технологиями, используемыми под IP/MPLS на канальном уровне, такими, как ATM, Frame Relay (рис. 6.6, *б, в*).

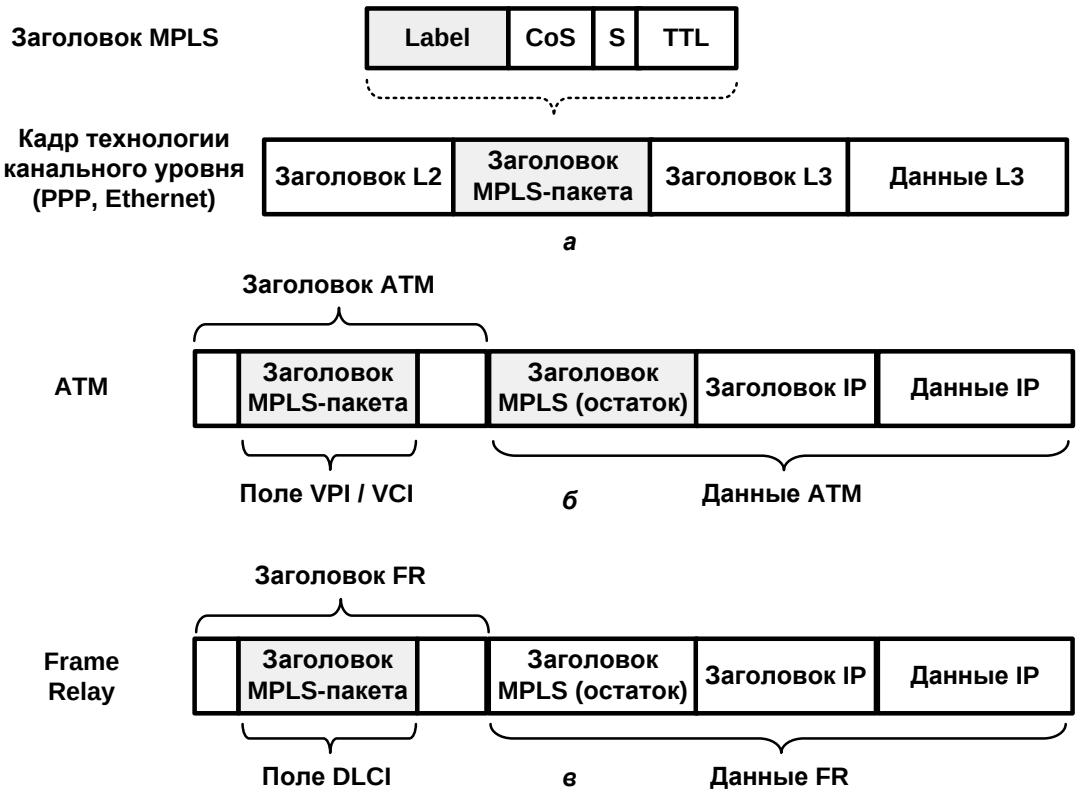


Рис. 6.6. Инкапсуляция заголовка MPLS в кадры технологий, используемых на канальном уровне

Если архитектура MPLS используется поверх технологий канального уровня PPP или Ethernet (к примеру, IP/MPLS/PPP или IP/MPLS/Ethernet), то заголовок MPLS встраивается между заголовком кадра технологии ка-

нального уровня (L2) и заголовком пакета технологии сетевого уровня (L3), к примеру, заголовком IP-пакета (рис. 6.6, *а*). Продвижение кадра в MPLS-сети происходит не на основе адресной информации, содержащейся в заголовке L2 (в случае Ethernet — это MAC-адрес), а на основе значения метки заголовка MPLS-пакета.

Если MPLS используется поверх сетевых технологий ATM или Frame Relay, работающих в данном случае на канальном уровне, то часть MPLS-заголовка (значение метки) размещается полях идентификаторов виртуальных каналов: VPI/VCI (20 бит) и DLCI (до 23-х бит), уже имеющихся в ячейках ATM и в кадрах FR соответственно. Остальная часть заголовка MPLS размещается между заголовком ATM (FR) и заголовком IP-пакета. Продвижение пакетов осуществляется на основе адресной информации в полях VPI/VCI (ATM) или DLCI (FR). В обоих случаях поле TTL времени жизни пакета, а для ATM — ещё и поля CoS и S-бит, остаются невидимыми для коммутатора. Детали обработки этой ситуации описаны в спецификации MPLS.

Далее рассмотрим организацию стека меток

Стек меток представляет собой одну из важнейших особенностей архитектуры MPLS. Применение стека меток позволяет создавать иерархическую систему агрегированных путей LSP, называемых *туннелями*, с любым количеством уровней иерархии. Эти туннели могут охватывать несколько сетевых сегментов (доменов).

MPLS-пакет может переносить несколько меток (рис. 6.7), организованных в виде стека LIFO (Last in First out, последним прибыл — первым обслужен). Обработка пакета всегда осуществляется по верхней метке. Метка, находящаяся на дне стека (ближайшая к IP-пакету), имеет признак S = 1.

Рис. 6.7. Формат MPLS-пакета со стеком меток

На любом LSR-маршрутизаторе метка может быть добавлена к стеку операцией *Push* (помещение в стек), заменена новой (*Swap*) или удалена из стека операцией *Pop* (выталкивание верхней метки).

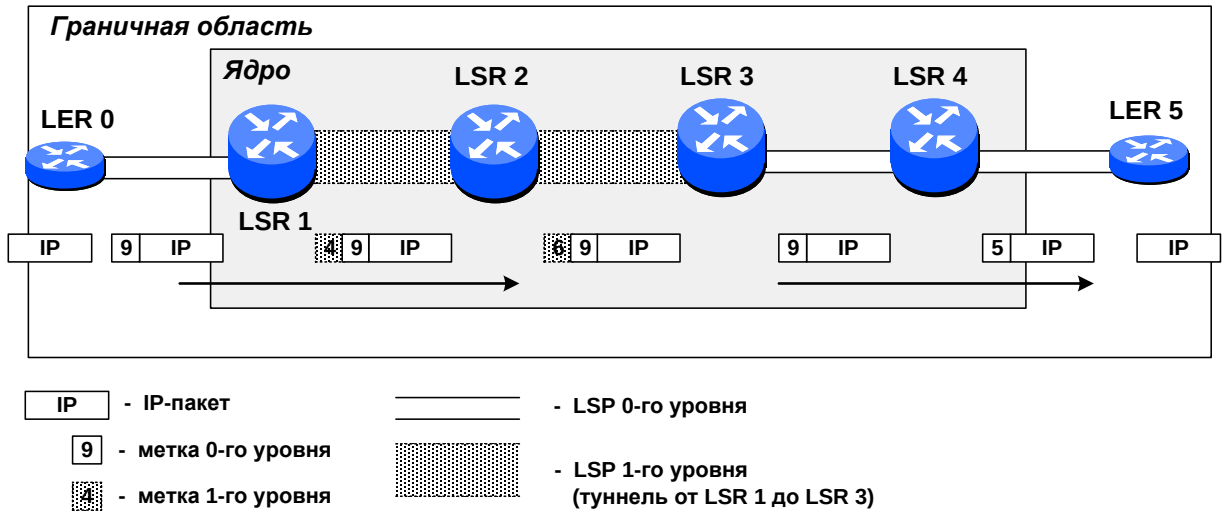


Рис. 6.8. Стеки меток и образование туннелей в технологии MPLS

Принцип образования туннеля на основе стекирования меток в одном MPLS-домене иллюстрируется на рис. 6.8. Двухуровневый туннель (LSP 1-го уровня) создан между маршртутизаторами LSR1 и LSR3 и может содержать несколько LSP 0-го уровня (на рис. 6.8 показан один). При поступлении на вход LSR1 пакета MPLS с одной меткой 9 к нему добавляется метка со значением 4, которая становится верхней в стеке.

Далее в туннеле продвижение пакета осуществляется на основе верхней метки. В маршрутизаторе LSR2 её значение заменяется на новое — 6. При этом значение нижней метки не изменяется (9). В конце туннеля в маршрутизаторе LSR3 верхняя метка удаляется (выталкивается) и на её место становится нижняя метка, имеющая значение 9. В случае продвижения пакета через несколько MPLS-доменов туннели могут быть созданы в каждом домене.

Такой подход к образованию туннелей (агрегированию LSP) в технологии MPLS похож на используемый двухуровневый стек в технологии ATM (виртуальные каналы внутри виртуальных путей), однако архитектура MPLS не имеет ограничений на количество уровней стека.

Организация меток в виде стека обеспечивает значительную гибкость при планировании и администрировании MPLS-сетей. Путем агрегирования LSP можно получить небольшое количество туннелей, а следовательно, сократить размер таблиц коммутации, упростить задачи управления ресурсами и масштабирования ядра сети.

Рассмотрим особенности обработки поля времени жизни MPLS-пакета (или счетчика ретрансляционных участков) в стеке меток. Обычно в объединенной IP-сети значение этого поля уменьшается на единицу на каждом маршрутизаторе, и когда счетчик обнуляется, пакет отбрасывается. Это делается для того, чтобы избежать слишком долгого пребывания IP-пакета в сети из-за неверной маршрутизации. Поскольку LSR-маршрутизатор не анализирует IP-заголовок, поле времени жизни включается в заголовок MPLS-пакета. Правила обработки поля времени жизни в MPLS-заголовке следующие.

1. Когда IP-пакет прибывает на входной пограничный маршрутизатор LER MPLS-домена, в стек пакета помещается один MPLS-заголовок (метка). Значение поля времени жизни этого MPLS-заголовка устанавливается равным значению поля времени жизни IP-заголовка после его обработки модулем IP-маршрутизации LER (т. е. уменьшенному на единицу).

2. Когда MPLS-пакет прибывает на промежуточный LSR-маршрутизатор MPLS-домена, значение поля времени жизни в метке (MPLS-заголовке), находящейся на вершине стека, уменьшается на единицу:

- если получившееся значение времени жизни нулевое, то MPLS-пакет либо просто отбрасывается, либо передается «обычному» сетевому уровню для обработки ошибки (например, протоколу ICMP для формирования сообщения об ошибке;

- если получившееся значение времени жизни положительное, то оно помещается в поле времени жизни метки, находящейся на вершине стека, после чего MPLS-пакет переправляется дальше. Значения полей времени жизни в MPLS-заголовках, находящихся не на вершине стека, не изменяются и на ход обработки пакета не влияют.

3. Когда MPLS-пакет прибывает на выходной пограничный маршрутизатор MPLS-домена, значение поля времени жизни уменьшается на еди-

ницу, после чего оставшаяся в стеке метка извлекается и стек становится пустым:

- если получившееся значение времени жизни нулевое, то IP-пакет либо просто отбрасывается, либо передается «обычному» сетевому уровню для обработки ошибки;

- если получившееся значение времени жизни положительное, то оно помещается в поле времени жизни IP-заголовка, после чего IP-пакет, переправляется дальше посредством обычной IP-маршрутизации. При этом контрольная сумма IP-заголовка пересчитывается заново.

Выбор маршрутов в MPLS-сетях означает выбор и установление LSP-пути для потоков конкретного FEC-класса. Существующая архитектура MPLS поддерживает два вида выбора маршрута:

1) маршрутизация «шаг-за-шагом» (на каждом ретрансляционном участке);

2) явная маршрутизация.

При маршрутизации на уровне ретрансляционных участков каждый узел (LSR-маршрутизатор) независимо выбирает следующий ретрансляционный участок для каждого FEC-класса. Это обычный режим для существующих IPv4-сетей. Такой вариант маршрутизации предполагает для выбора маршрута использование обычного протокола маршрутизации, например OSPF. После этого сигнальным протоколом осуществляется прокладка пути LSP на основе использования записей в маршрутных таблицах узлов. В отличие от маршрутизации в IP-сетях здесь после установления LSP-путей проявляются такие преимущества архитектуры MPLS, как быстрая коммутация по меткам, возможность агрегирования путей на основе стека меток, дифференцированная обработка пакетов различных FEC-классов, следующих по одному и тому же маршруту. Однако из-за ограниченности метрик, используемых в типичных протоколах маршрутизации, маршрутизация по ретрансляционным участкам не поддерживает конструирование (инжиниринг) трафика, проведение какой-либо политики качества обслуживания, безопасность и т.д.

При явной маршрутизации, каждый LSR не выбирает следующий шаг (ретрансляционный участок) независимо. Один LSR, обычно входной узел для данного FEC-класса специфицирует несколько (или все) промежуточные LSR в LSP-пути.

Если один LSR специфицирует все LSR на LSP-пути, то говорят о *жесткой явной маршрутизации*. Если один LSR специфицирует только некоторые LSR на LSP-пути, то говорят о *гибкой явной маршрутизации*.

Последовательность явно маршрутизированных LSR может быть выбрана при конфигурации (т.е. заранее), или динамически одним узлом. Например, входной узел может использовать информацию о топологии и качестве обслуживания в MPLS-домене, полученную из маршрутной базы данных, для того чтобы вычислить весь путь для дерева с корнем в выходном узле.

Явная маршрутизация позволяет использовать все преимущества архитектуры MPLS, включая возможность инжиниринга трафика и проведение определенной политики качества обслуживания.

Конечным итогом маршрутизации в MPLS-домене является прокладка LSP-путей для каждого FEC-класса. Установление LSP-пути эквивалентно определению локальных значений меток MPLS-пакета на каждом ретрансляционном участке вдоль всего маршрута следования MPLS-пакета. Указанная задача решается с помощью протоколов распределения меток. Архитектура MPLS не зависит от конкретного протокола, поэтому в сети для распределения меток могут применяться разные протоколы сетевой сигнализации, в качестве которых могут использоваться следующие протоколы:

- LDP (Label Distribution Protocol) — протокол распределения меток;
- RSVP-TE — расширение для MPLS-протокола резервирования ресурсов;
- BGP-TE — усовершенствованная для MPLS версия протокола маршрутизации BGP-4 и др.

Назначение метки для данного FEC-класса на каждом ретрансляционном участке производится «нижним по течению» LSR-маршрутизатором. Информация о назначенной метке передается соседнему узлу, расположенному «выше по течению». Распространение информации о значении присвоенной метки может быть инициировано запросом от верхнего устройства LSR (downstream on-demand) либо осуществляться спонтанно (unsolicited downstream).

Существуют два режима распределения меток: независимый и упорядоченный. Первый предусматривает возможность уведомления верхнего узла о значении присвоенной данным LSR метки до того, как этот LSR по-

лучит сообщение о назначенной для данного класса метке от своего нижнего соседа. Второй режим разрешает высылать подобное уведомление только после получения таких сведений от нижнего LSR, за исключением случая, когда маршрутизатор LSR является выходным для этого FEC.

Рассмотрим создание LSP-пути (заполнение таблиц коммутации) на примере протокола распределения меток LDP. Протокол LDP позволяет автоматически создавать LSP-пути в MPLS-сети на основе сформированных в каждом узле таблиц маршрутизации с помощью протоколов внутренней маршрутизации (к примеру, OSPF).

Предположим, что выбран упорядоченный режим распределения меток LSP со спонтанным распространением сведений между LSR о назначенных метках (рис. 6.9).

На стадии *A* каждое из устройств сети MPLS строит базу топологической информации, применяя один из протоколов маршрутизации (на рисунке — протокол OSPF).

На стадии *B* маршрутизаторы LSR применяют процедуру нахождения соседних устройств и устанавливают с ними сеансы LDP. Соседями считаются LSR-устройства с поддержкой протокола LDP, непосредственно соединенные между собой или логически (при туннельной передаче).

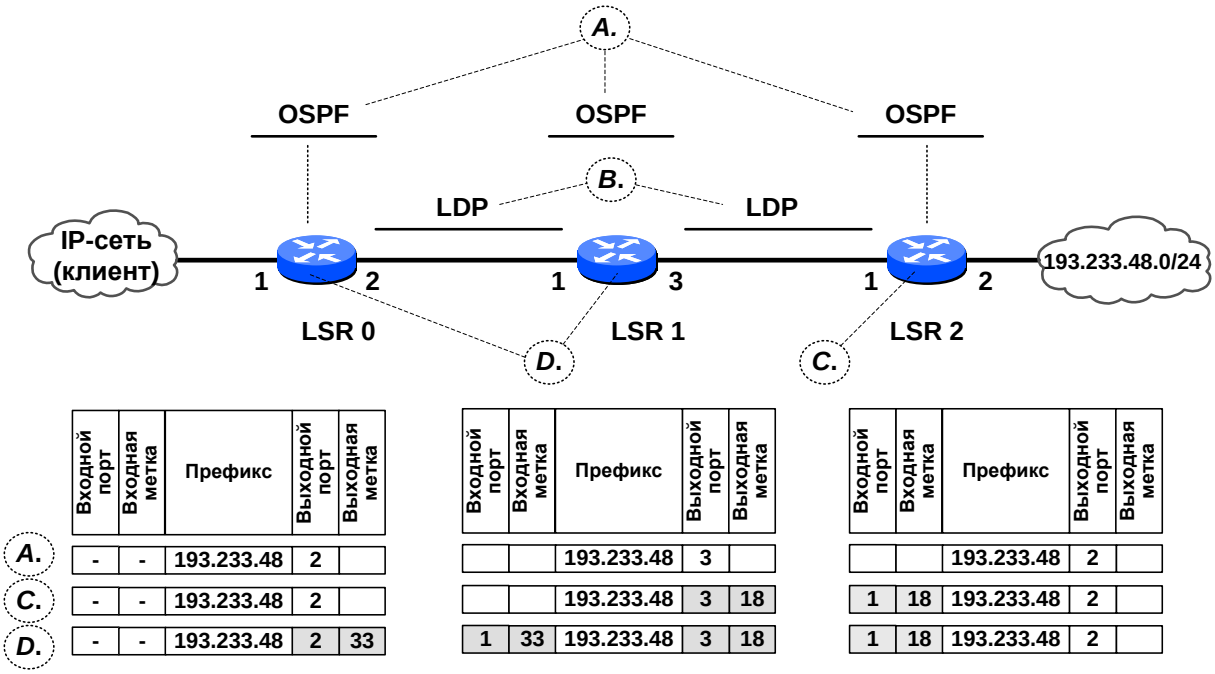


Рис. 6.9. Установление LSP-пути с помощью протокола LDP

Определение соседей осуществляется путем многоадресной рассылки сообщений *Hello* транспортного протокола UDP на порт 646 UDP по общеизвестному групповому IP-адресу 224.0.0.2. Сообщения *Hello* содержат информацию о промежутке времени, в течение которого соседние LSR должны обмениваться сообщениями (по умолчанию — 15 или 45 с), и некоторую другую служебную информацию (транспортные адреса LSR).

После идентификации соседей производится установление сессии LDP между соседними LSR, в ходе которого завершается установление соседских отношений, производится обмен и согласование параметров LDP-сессии, включая согласование пространств используемых меток. LDP-сообщения при установлении сессии и её функционировании передаются поверх TCP (порт 646). В результате открывается транспортное TCP-соединение между модулями LDP взаимодействующих LSR, по которому передается информация о назначенных метках и служебная информация (тестовые сообщения *Keepalive* для проверки работоспособности устройств).

На стадии *C* LSR 2 на основе анализа собственных таблиц маршрутизации обнаруживает, что он является выходным LSR для пути, ведущего к IP-сети 193.233.48.0. Тогда LSR 2 ассоциирует класс FEC с пакетами, адрес получателя которых соответствует префиксу данной сети, и присваивает этому классу случайное значение метки — в рассматриваемом случае 18. После этого протокол LDP уведомляет верхний маршрутизатор LSR 1 о том, что потоку, адресованному сети с префиксом 193.233.48, присвоена метка 18. LSR 1 помещает это значение в поле выходной метки своей таблицы коммутации.

На стадии *D* устройство LSR 1, которому известно значение метки для потока, адресованного на префикс 193.233.48, присваивает собственное значение метки данному FEC и уведомляет верхнего соседа LSR 0 об этой метке. LSR 0 записывает полученную информацию в свою таблицу коммутации. После завершения данного процесса все готово для передачи пакетов из сети «клиента» в сеть с адресом 193.233.48.0, т. е. по выбранному пути LSP.

Протокол LDP должен реагировать как на появление новой записи FEC в таблице маршрутизации, так на исчезновение записи FEC из табли-

цы маршрутизации. Во втором случае предусмотрено два режима удержания меток (Label Retention Mode): консервативный (Conservative Label Retention Mode) и свободный (Liberal Label Retention Mode).

При использовании консервативного режима удержания меток при уничтожении маршрута на FEC метка удаляется. Для восстановления LSP необходимо, чтобы метка была заново выделена соседним нижним LSR-маршрутизатором. Если используется свободный режим удержания меток, то при уничтожении маршрута на FEC метка не удаляется, а лишь помечается как неактивная. И в случае, если маршрут на FEC восстанавливается через тот же нижний LSR, метка не запрашивается, а используется старая, статус которой меняется на активный.

Сочетания режимов распределения меток (независимый или упорядоченный), распространения информации о метках (по запросу или без запроса — спонтанно) и удержания меток (консервативный или свободный) определяют множество вариантов работы протокола LDP. В данной LDP-сессии может использоваться один из вариантов работы LDP-протокола.

ВОПРОСЫ И ЗАДАНИЯ К ГЛАВЕ 6

1. В чем состоят отличия архитектуры и функций, реализуемых IP-маршрутизатором и MPLS-маршрутизатором, коммутирующим по меткам?
2. Какие механизмы продвижения данных поддерживает MPLS-маршрутизатор, коммутирующий по меткам?
3. Перечислите отличительные особенности и области применения технологии MPLS.
4. В чем состоит многопротокольность MPLS?
5. Что такое конструирование трафика?
6. Опишите архитектуру сети MPLS. Каковы отличия функций, реализуемых LSR и LER?
7. Дайте пояснение термина «виртуальный канал».
8. Опишите механизм продвижения пакетов по меткам.
9. Что такое «стек меток», «туннель»?
10. Поясните особенности обработки поля времени жизни MPLS-пакета.
11. Каким образом в технологии MPLS определяются требования к качеству обслуживания пакетов?
12. Какие виды выбора маршрута поддерживаются технологией MPLS?
13. Опишите процедуру установления пути с помощью протокола LDP.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Столингс В*. Компьютерные системы передачи данных : [пер. с англ.] / В. Столингс. — М.: Вильямс, 2002. — 928.

2. *Столингс В*. Беспроводные линии связи и сети: [пер. с англ.] / В. Столингс. — СПб.: Вильямс, 2003. — 640 с.

3. Телекоммуникационные системы и сети: учебное пособие. В 3-х томах. Т. 1 / В. П. Шувалов [и др.]; под ред. проф. В. П. Шувалова. — М. : Горячая линия — Телеком, 2004. — 672 с.

4. Широкополосные беспроводные сети передачи информации / В. М. Вишневский [и др.]. — М.: Техносфера, 2005. — 592 с.

5. *Шахнович И.В*. Современные технологии беспроводной связи / И. В. Шахнович. — М.: Техносфера, 2006. — 288 с.

6. *Олифер В.Г*. Компьютерные сети. Принципы, технологии, протоколы / В. Г. Олифер, Н. А. Олифер. — СПб: Питер, 2006. — 960 с.

7. *Битнер В.И.* Нормирование качества телекоммуникационных услуг / В. И. Битнер, Г. Н. Попов. Учебное пособие. Под ред. профессора В.П. Шувалова. — М. : Горячая линия — Телеком, 2004. — 312 с.

8. *Таненбаум Э.* Компьютерные сети / Э. Таненбаум. — СПб. : Питер, 2003. — 992 с.

Рашич Валерий Остаевич
Фуртиков Валентин Михайлович
Рашич Андрей Валерьевич

СЕТЕВЫЕ ТЕЛЕКОММУНИКАЦИОННЫЕ ПРОТОКОЛЫ

Учебное пособие

Лицензия ЛР № 020593 от 07.08.97
Налоговая льгота – Общероссийский классификатор продукции
ОК 005-93, т. 2; 953005 – учебная литература

Подписано к печати Формат 60x84/16. Печать цифровая.
Усл. печ. л. 14,19 . Уч.-изд. л. . Тираж экз. Заказ

Отпечатано с готового оригинал-макета, предоставленного авторами,
в Цифровом типографском центре
Издательства Политехнического университета.
195251, Санкт-Петербург, Политехническая ул., 29.
Тел.: (812) 550-40-14.
Тел./факс: (812) 297-57-76.