А.Б. Гольдштейн, Б.С. Гольдштейн

MPLS

ТЕХНОЛОГИЯ И ПРОТОКОЛЫ

А. Б. Гольдштейн, Б. С. Гольдштейн

Технология и протоколы MPLS

«БХВ — Санкт-Петербург»
2005

УДК 621.39
004.724
ББК 32.973.202
32.81
С59

Гольдштейн А. Б., Гольдштейн Б. С.
Технология и протоколы MPLS
СПб.: БХВ — Санкт-Петербург, 2005. — 304 с.: ил.

ISBN 5-8206-0126-2

Технология MPLS представляет собой синтез всего самого лучшего из технологий уровня 2 (ATM, Frame Relay, Ethernet) и маршрутизации уровня 3 пакетных сетей. Изучение рассмотренных в книге протоколов MPLS позволит читателю оценить, является ли MPLS именно тем инструментом, с помощью которого сегодняшний хаос неуправляемой передачи пакетов по IP-сети может быть превращен в стройный, эффективно функционирующий механизм для сети связи следующего поколения NGN. На этот вопрос читатель сможет ответить сам, прочитав книгу, а также узнать про классы эквивалентности FEC, метки, протоколы LDP, CR-LDP, RSVP, RSVP-TE, OSPF, BGP-4, IS-IS, трафик-инжиниринг, GMPLS и многое другое.

Для технических специалистов, занятых разработкой и эксплуатацией сетей связи, студентов старших курсов и аспирантов соответствующих специальностей, для всех, кто интересуется современными инфокоммуникациями.

Научно-техническое издание

ISBN 5-8206-0126-2

© Гольдштейн А. Б., Гольдштейн Б. С., 2005

Alexander Goldstein, Boris Goldstein.
MPLS Technology & Protocols – BHV, St.Petersburg, 2005.

The MPLS technology is a synthesis of all the best from the Layer 2 technologies (ATM, Frame Relay, Ethernet) and Layer 3 packet routing. The study of the MPLS protocols discussed in the book will allow the reader to evaluate whether MPLS is that very tool by means of which the today's chaos of uncontrolled packet transmission over the IP network may be transformed into a well-shaped and effective mechanism for the Next Generation Network. The reader will be able to answer this question himself/herself after reading the book, and also to get knowledge of Forwarding Equivalence Classes (FEC), labels, LDP, CR-LDP, RSVP, RSVP-TE, OSPF, BGP-4, and IS-IS protocols, traffic engineering, GMPLS, etc.

The book is intended for engineers involved in the development and operation of telecommunication networks. It will be a valuable reference source for university students of senior years and post-graduates studying in these areas. It will be helpful for all those who are interested in state-of-the-art infocommunication technologies.

Technical edition

Copyright © A. Goldstein, B. Goldstein 2005

Содержание

Предисловие

«Когда я назначаю кого-то на лидирующую позицию, то имею девяносто девять недовольных и одного неблагодарного», говорил Людовик XIV. Существование, как минимум, девяносто девяти претендентов на позицию главного механизма обеспечения качества обслуживания QoS (Quality of Service) в сетях связи следующего поколения NGN (Next Generation Network) обусловлено лавинообразным ростом числа пользователей IP-сетей и соответствующим увеличением мультимедийного трафика. Изначально же IP-технологии были ориентированы на передачу данных простейших приложений, для чего было вполне достаточно программных маршрутизаторов. По мере появления новых мультимедийных приложений типа IP-телефонии [20], требующих более высоких скоростей передачи и поддержки более широкой полосы пропускания, возникла потребность в создании технологий и устройств, обеспечивающих возможность быстрой коммутации на уровне 2 (уровне звена данных) и уровне 3 (сетевом уровне) аппаратными средствами. Появились устройства коммутации на уровне 2, решающие проблему узких мест коммутации в среде LAN, а также новые маршрутизаторы, улучшающие ситуацию с маршрутизацией на уровне 3 путем перевода в быстродействующую аппаратную реализацию процедур просмотра маршрутных таблиц для пересылки пакетов. Достигли своего апогея наиболее перспективные технологии 90-х годов Frame Relay и ATM. Появились разнообразные средства обеспечения качества обслуживания DiffServ и IntServ, рассматриваемый в этой книге протокол резервирования RSVP и многое другое.

И все же, все эти девяносто девять претендентов на наиболее эффективное решение проблемы сетевого QoS при передаче мультимедийной информации в реальном времени с учетом таких показателей, как задержка, дрожание фазы, перегрузка и т.п. фактически уже уступили занявшей лидирующую позицию многопротокольной коммутации по меткам MPLS (MultiProtocol Label Switching).

MPLS является весьма изящным и универсальным решением проблем QoS, стоящих перед сегодняшними пакетными сетями, решением, которое обеспечивает скорость передачи, масштабируемость, оптимизацию распределения трафика и эффективную маршрутизацию (на основе показателей QoS) в пакетных сетях IP, ATM и Frame Relay. Лидерство MPLS обусловлено, по мнению авторов, удачно выбранной позицией, позволяющей оптимальным образом отображать сквозной трафик третьего уровня от исходящего сетевого узла (маршрутизатора) к входящему узлу в трафик между соседними узлами на втором уровне сетевой иерархии. Т.о., MPLS, являясь гибридом уровней 2 и 3 семиуровневой модели OSI, собрала вместе лучшее из двух миров: уровня 2 и уровня 3, мира ATM и мира IP.

Некоторая «неблагодарность» в ответ на столь значительное внимание к технологии MPLS проявилась в возникшей путанице относительно того, что такое MPLS, к какому уровню OSI она относится, что эта технология может и для чего она предназначена. Такая путаница обусловлена, в частности, тем, что технология MPLS не использует какой-либо единый формат для транспорта пакетов, а базируется на нескольких протоколах (и соответствующих им стандартах), каждый из которых решает отдельную проблему. Попытка распутать этот клубок и каким-то образом структурировать описания протоколов MPLS как раз и предпринята в этой книге и, прежде всего, в ее первой главе.

Вторая глава посвящена собственно меткам. Сами по себе, метки не являются чем-то новым. Технологии X.25, ATM, Frame Relay и, до некоторой степени, TDM используют инкапсуляцию с помощью меток в течение многих лет. Хороший пример такой инкапсуляции в TDM при передаче сообщений протокола DSS1 ISDN через универсальный интерфейс сети доступа V5.2 был рассмотрен в одной из предыдущих книг этой серии, посвященной протоколам сети доступа [21]. Но только в MPLS эта концепция реализуется в общем виде, т.е. метки не привязаны к какому-либо конкретному протоколу второго уровня. Метка MPLS представляет собой число, уникальным образом идентифицирующее некоторую совокупность передаваемых пакетов. Метка имеет только локальное значение, т. е. по мере следования пакетов вдоль маршрута ее необходимо изменять. О самих метках, о распределении меток, о коммутации по меткам, о соответствующих протоколах сигнализации и маршрутизации, о методах инжиниринга трафика, о VPN, о туннелях MPLS и о многом другом рассказывается в этой книге.

Книга построена таким образом, что в ней можно условно выделить три части. В первых главах представлено описание технологии MPLS и рассматриваются:

- принципы MPLS, сильные и слабые стороны этой технологии,
- краткая история MPLS, основы архитектуры,
- классы эквивалентности пересылки FEC и коммутируемые по меткам тракты LSP,
- структура метки, методы распределения меток и коммутации по меткам.

Задача этих глав — ввести ключевые определения и термины, относящиеся к MPLS, и объяснить технологии, в результате слияния которых и была создана MPLS. Сюда входит рассмотрение технологий маршрутизации и коммутации, ранних исследований в области коммутации ячеек и тегов, а также других важных технологий, которые являются ключом к пониманию эволюции технологии MPLS.

Задача следующей, наиболее объемистой части книги — объяснить базовую технологию и протоколы MPLS, включая протоколы сигнализации, механизмы распределения меток и коммутации по меткам, и содержит следующие главы:

- протокол LDP и распределение меток,
- протокол RSVP и идеи инжиниринга трафика,
- протокол OSPF и метрики маршрутизации,
- протокол BGP и пограничные шлюзы,
- протокол IS-IS и внутризоновая маршрутизация.

Уже простое перечисление глав показывает, что эта книга могла бы быть издана в *серии «Телекоммуникационные протоколы»*, как первоначально и планировалось. Но книга, которая получилась и которую вы держите в руках, существенно отличается от вышедших в серии «Телекоммуникационные протоколы» книг, в первую очередь, тем, что отнюдь не является справочником. Это вполне объяснимо: справочники по телекоммуникационным протоколам написаны для гораздо более «зрелых» технологий ОКС7 [22,24], V5 [23], R1.5 [25]. В данной же книге представлен не столько справочник, сколько, в определенном смысле, путеводитель по технологии MPLS, чрезвычайно молодой, но, тем не менее, уже заслужившей место в «Энциклопедии современных инфокоммуникаций».

Это изменение подхода к книге обусловило и то, что в третьей ее части обсуждаются направления текущих и будущих работ в области технологии и протоколов MPLS, роль MPLS в конвергенции сетей и услуг связи, перспективные вопросы MPLS:

- инжиниринг трафика в MPLS,
- туннели MPLS и виртуальные частные сети VPN,
- MPLambdaS и GMPLS,
- перспективы технологии MPLS.

Заключительные главы книги также требуют краткого предварительного комментария. Первоначально MPLS рассматривалась как технология, которая сумела бы радикально улучшить маршрутизацию IP-пакетов на уровне 3 модели OSI. До самого последнего времени главным был тот аргумент, что благодаря применению MPLS значительно возрастает производительность сети, т.к. анализ коротких меток фиксированной длины выполняется намного быстрее, чем анализ длинных IP-заголовков сетевого уровня при традиционной маршрутизации IP-пакетов. Однако этот аргумент уже сегодня становится менее актуальным в связи с появлением аппаратных решений на интегральных схемах прикладной ориентации ASIC, на программируемых в процессе эксплуатации матрицах FPGA и т.п. Анализ заголовков IP-пакетов в самом ближайшем будущем не будет являться узким местом для производительности терабитовых (а вскоре — и петабитовых!) маршрутизаторов традиционных IP-сетей.

С другой стороны, на первый план вышла проблематика качества обслуживания QoS, возрастает интерес к новым приложениям, таким как использование MPLS в оптических сетях коммутации, инжиниринг трафика TE, речь поверх MPLS, виртуальные частные сети VPN, что и обусловило как подбор материала, так и структуру третьей части книги.

Список литературы весьма обширен и содержит 125 наименований. И даже в этот список вошло далеко не все, что уже написано про MPLS, но поскольку перечисленные источники сами содержат литературные ссылки, наш перечень будет нетрудно расширить. Представляются также полезными предметный указатель и список использованных в книге аббревиатур, составляющий основу глоссария.

Авторы пользуются случаем выразить свою признательность профессору Г.Г. Яновскому, впервые обратившему внимание авторов на эту проблематику, причем тогда, когда вышеприведенные сентенции о перспективности MPLS были отнюдь не так очевидны как сегодня. Кроме того, книга была бы заметно хуже, если бы не научное редактирование, выполненное В.А. Соколовым. Весьма полезными были и стимулирующие дискуссии с коллегами из ЛОНИИС, СПбГУТ, ГК Экран, НТЦ Протей, Уралсвязьинформ, МТУ-информ/Комстар, Ленсвязь, ЮТК, АДЭ и др. Значительная помощь в работе над книгой была оказана студентами старших курсов СПбГУТ им. проф. М.А. Бонч-Бруевича — А. Атциком и его коллегами, материалы дипломных проектов которых приведены в качестве некоторых сценариев и примеров в ряде глав книги.

Все это, безусловно, содействовало решению главной задачи, которую ставили перед собой авторы: показать, что многопротокольная коммутация по меткам — это именно тот инструмент, с помощью которого сегодняшний хаос неуправляемой передачи пакетов по IP-сетям может быть превращен в стройный, эффективно функционирующий механизм для NGN.

Глава 1
Основы MPLS

1.1. Технология MPLS

Технология *многопротокольной коммутации по меткам MPLS* явилась результатом слияния нескольких сходных технологий, которые были изобретены в середине 1990-х годов. Наиболее известная из них (хотя и не первая, увидевшая широкий свет) была названа ее изобретателями — компанией Ipsilon — *IP Switching.* Ранее компания Toshiba уже описала похожий механизм — *Cell Switching Router* (*CSR*), а вскоре были обнародованы сведения и о некоторых других технологиях, среди которых отметим *Tag Switching* (Cisco Systems) и *ARIS* (IBM). Эти механизмы имеют ряд общих черт. Все они используют для пересылки пакетов простой метод замены меток и разработанную для Интернет структуру управления, т.е. IP-адреса и стандартные протоколы маршрутизации, например OSPF и BGP. Мы кратко рассмотрим эти технологии в следующем параграфе.

Возрастающий интерес к коммутации с использованием меток привел к созданию специальной рабочей группы *IETF*, целью которой было выработать на основе вышеупомянутых механизмов общий стандарт. Не желая давать группе название, которое соответствовало бы продукту какой-либо одной компании, IETF остановила свой выбор на нейтральном (правда, слегка громоздком) названии: *многопротокольная коммутация по меткам*.

Можно встретить другие переводы словосочетания label switching: «коммутация меток», «коммутация на базе (или, что то же самое, — на основе) меток». Первый из этих переводов просто неверен — ведь коммутируются не метки, а пакеты. Второй несколько искажает действительное положение вещей — базу (или основу) коммутации составляют далеко не только метки.

Перед пересылкой принятого пакета на следующий участок маршрута устройство коммутации по меткам обычно вставляет в этот пакет, некоторую новую метку вместо той, которая в нем содержалась. Эта операция называется *заменой меток (label swapping).* Для того чтобы определить, куда пересылать пакеты,

устройство коммутации по меткам, которое называется *маршрутизатором LSR (Label Switching Router)*, использует стандартные протоколы управления IP-сетью.

Многопротокольной (Multi-Protocol) коммутацией MPLS называется потому, что ее средства применимы к *любому* протоколу сетевого уровня, т.е. MPLS — это своего рода инкапсулирующий протокол, способный транспортировать информацию множества протоколов низших уровней модели OSI, как это показано на рис. 1.1.

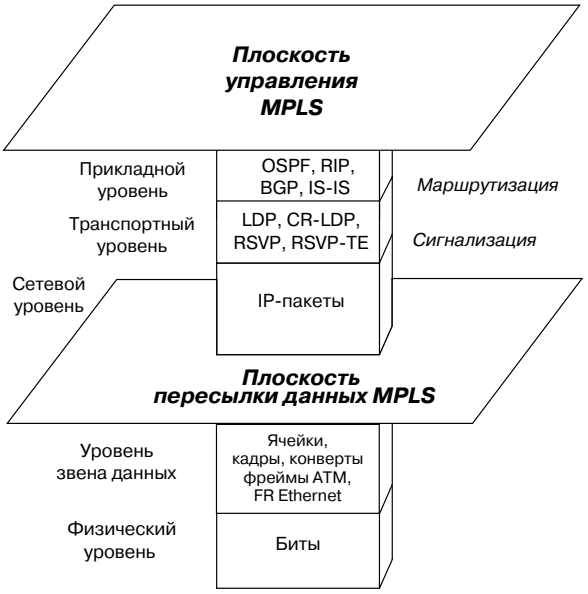


Рис. 1.1. Плоскости MPLS

Напомним, что первый, *физический уровень* (physical layer) содержит функции, обеспечивающие использование физической среды для двусторонней *передачи битов* (с такой достоверностью, какую обеспечивает эта среда) по прямому тракту, связывающему два узла сети. Второй уровень — *уровень звена данных* (data link layer) — содержит функции, обеспечивающие формирование в этом тракте надежного логического звена связи, по которому происходит двусторонний обмен *информационными блоками* между названными узлами; при этом путем обнаружения и исправления ошибок гарантируется заданная достоверность передачи. Третий, *сетевой уровень* содержит функции, обеспечивающие транспортировку информационных блоков от отправителя к получателю через несколько узлов сети по подходящему маршруту транспортировки, который составляется из звеньев второго уровня. Общая идея протоколов всех уровней (кроме физического) состоит в том, что информационный блок каждого уровня содержит заголовок и информационное поле, и в том, что блок протокола вышележащего

уровня помещается в информационное поле блока протокола расположенного сразу под ним нижележащего уровня.

Представленная на рис. 1.1 плоскость пересылки данных MPLS не образует полноценного уровня, она «вклинивается» в сетях IP, ATM или Frame Relay между 2-м и 3-м уровнями модели OSI, оставаясь независимой от этих уровней. Можно сказать, что одновременное функционирование MPLS на сетевом уровне и на уровне звена данных приводит к образованию так называемого *уровня 2.5*, где, собственно, и выполняется коммутация по меткам.

1.2. Предыстория MPLS

Теперь, когда, более или менее, перечислены мотивы, побудившие создателей различных технологий коммутации по меткам взяться в свое время за эту работу, имеет смысл обратиться к последовательности событий, которые привели, в конечном итоге, к сегодняшней MPLS.

С начала 1990-х годов мощными двигателями развития технологии коммутации по меткам были проблемы обеспечения совместимости протоколов IP и ATM. Поэтому на историю MPLS значительное влияние оказала сложившаяся тогда ситуация вокруг проблемы маршрутизации пакетов IP по сетям ATM. Попытки развивать в этом направлении стандарты протоколов ATM предпринимались еще раньше — в конце 1980-х годов. Уже тогда неоправданно преувеличенные перспективы технологии ATM обусловили начало разработки механизма переноса IP-пакетов по сетям ATM. Этой проблемой занялись сразу несколько рабочих групп в составе комитета IETF, а на рубеже 1993/94 годов ими были опубликованы два важных документа серии RFC.

Первый стандарт, посвященный IP-поверх-ATM (IPoATM), описан в RFC 1483 и касается простой проблемы: каким образом инкапсулировать IP-дейтаграммы (и пакеты других протоколов) в канал ATM. Второй стандарт, описанный в RFC 1577, определяет классический механизм передачи IP-пакетов по сети ATM и протокол преобразования адресов ATMARP. Классический механизм предполагает, что маршрутизаторы пакетов IP и хосты, находящиеся в одной и той же подсети (т.е. сетевые и подсетевые составляющие их адресов одинаковы), могут взаимодействовать через эту подсеть. Если они находятся в разных подсетях, то для пересылки пакета от подсети отправителя к подсети получателя необходим еще один или несколько маршрутизаторов. При определении классической модели IP-поверх-ATM было признано, что IP-устройства могут подключаться к общей сети ATM (например, к крупной сети ATM, принадлежащей оператору сети общего пользования) и при

этом находиться в подсетях, которые в эту общую сеть не входят. Таким образом, была предложена идея логической подсети IP (LIS), которая состоит из совокупности хостов и маршрутизаторов IP-пакетов, подключенных к общей сети ATM и имеющих общий адрес с IP-сетью и с подсетью.

Документ RFC 1577 специфицирует только взаимодействие внутри LIS и предполагает, что для передачи пакета из одной LIS в другую он должен проходить через маршрутизатор, подключенный к обеим LIS. Прежде всего, в связи с классической моделью нужно отметить следующее. Она подразумевает, что два IP-устройства, подключенных к одной и той же сети ATM, но принадлежащих разным подсетям LIS, не смогут использовать для обмена IP-дейтаграммами единый виртуальный канал VC, проходящий через сеть ATM. Вместо этого они будут вынуждены передавать пакеты через отдельный маршрутизатор. Такой механизм многим специалистам казался тогда непривлекательным, особенно, в связи с тем, что на тот период времени производительность серийно выпускавшихся коммутаторов ATM значительно превышала производительность маршрутизаторов. Несмотря на то, что возможным вариантом решения этой проблемы могло бы стать введение правила, согласно которому сеть ATM должна представлять собой одну LIS, такой вариант часто трудно реализовать по чисто организационным причинам. Например, маловероятно, что две совершенно разные организации, подключенные к одной сети ATM общего пользования, захотят использовать для своих IP-адресов общее адресное пространство. Далее, после определения общей концепции LIS, в RFC 1577 определен ключевой механизм управления организацией связи между двумя IP-устройствами, входящими в одну подсеть LIS, — протокол ATMARP. Используется протокол преобразования адресов ARP в традиционной локальной сети, позволяющий IP-устройствам получать адресную информацию, которая необходима для организации связи, например, адреса оборудования локальной сети Ethernet. Аналогичным образом, ATMARP позволяет двум IP-устройствам узнать ATM-адреса друг друга. В протокол ATMARP было введено новое понятие ARP-сервера подсети LIS, который преобразует IP-адреса в адреса оборудования сети ATM для данной LIS. Каждое IP-устройство подсети LIS регистрируется на ARP-сервере и получает его адрес в сети ATM и его IP-адрес. После этого любое устройство LIS может запросить у ARP-сервера преобразование IP-адреса в адрес сети ATM, а уже снабженное адресом сети ATM устройство сможет создать виртуальный канал к этому адресу, используя сигнализацию ATM, и затем передать свои данные.

Решение проблемы создания виртуального канала ATM к IP-устройству другой LIS, отсутствующее в RFC 1577, взяла на себя рабочая группа ROLC в составе IETF. Предложенный ею механизм — *протокол Next Hop Resolution Protocol (NHRP)*. Протокол NHRP позволяет IP-устройству одной логической подсети IP узнать ATM-адрес другого IP-устройства, с которым ему нужно установить связь, с помощью специального *сервера следующей пересылки (Next Hop Server)* и организовать виртуальный канал связи с этим устройством, используя сигнализацию ATM.

Однако во всех этих предшествовавших MPLS работах не подвергался сомнению базовый принцип: маршрутизаторы выполняют функции маршрутизации, а коммутаторы выполняют функции коммутации, и устройства этих двух типов всегда функционируют порознь. При этом имеются в виду не только IP-маршрутизаторы и ATM-коммутаторы, но и само правило разделения функций уровней 3 и 2 между различными технологиями и устройствами. Дальнейшая история соответствует высказыванию, которое приписывают Альберту Эйнштейну: «Все давно знают, что то-то и то-то совершенно невоможно. Но вот находится невежда, который этого не знает, и он-то и совершает открытие».

В роли такого «невежды» выступила компания *Toshiba,* практически впервые подвергшая сомнению этот принцип и в 1994 году анонсировавшая *маршрутизатор коммутации ячеек CSR (Cell Switching Router)*. В архитектуре CSR реализована идея управления коммутационным полем ATM-коммутатора с помощью протоколов IP, а не протоколов сигнализации сети ATM типа Q.2931. Подобный подход, будучи доведенным до логического завершения, смог бы свести на нет необходимость использования практически всей сигнализации ATM и всех функций мэппинга между IP и ATM. Этот подход позволяет совместно использовать традиционные коммутаторы ATM и оборудование CSR; например, CSR могут обеспечить взаимосвязь между подсетями LIS, устраняя необходимость в протоколе NHRP. Проект CSR был представлен на обсуждение рабочей группы комитета IETF в 1994 году, а немного позже, в начале 1995 года, — на технической сессии BOF комитета IETF, однако интерес к этой проблеме тогда был довольно низким.

Компании Ipsilon (в настоящее время — подразделение фирмы Nokia), благодаря более полным техническим спецификациям *IP Switching* и наличию готового продукта IP switch, обычно приписывается честь создания первой по-настоящему формализованной концепции MPLS-подобной коммутации по меткам в сетях IP, получившей значительно большее признание, нежели технология CSR. Сам IP Switch состоял из ATM-коммутатора и контроллера IP-коммутатора, который выполнял функции управления. Контроллер IP-коммутатора фактически являлся отдельным устройством, содержащим функциональные объекты маршрутизации и пересылки данных.

Среди рассматриваемых в этом параграфе технологий в пользу технологии IP Switching можно привести существенный аргумент: IP Switching позволяет устройству, обладающему функциональными возможностями ATM-коммутатора, выполнять также и работу маршрутизатора, а мэппингу между IP и ATM — вообще не использовать управляющие протоколы ATM. В мае 1996 года вышел документ RFC 1953 «Ipsilon Flow Management Protocol Specification for IPv4. Version 1.0», а в августе того же года — RFC 1987 «Ipsilon's General Switch Management Protocol Specification. Version 1.1». Эти публикации позволили компании Ipsilon официально заявить, что их технология является открытой, поскольку использованные в ней базовые протоколы общедоступны. Коммутация по меткам в IP Switching фактически основывалась на классификации потоков по таким параметрам, как IP-адрес и номер порта отправителя, IP-адрес и номер порта получателя, тип протокола. Потоки классифицировались как устойчивые (пересылка файлов по протоколу FTP, трафик HTTP, Telnet и др.) и как кратковременные (обращения к системе имен доменов DNS, сообщения протокола SNMP и протокола сетевого времени NTP). *Протокол Ipsilon Flow Management Protocol (IFMP)* относил трафик к тому или иному классу и помогал создавать виртуальный канал, требующийся для пересылки трафика этого класса, а для того чтобы конфигурировать коммутационное поле ATM-коммутатора, использовался *протокол Generic Switch Management Protocol (GSMP)*, который позволял добавлением внешнего контроллера превратить практически каждый коммутатор ATM в коммутатор пакетов IP. Механизм IP Switching был чрезвычайно важным шагом, поскольку предлагал жизнеспособную парадигму замены меток, а также вводил метод классификации IP-трафика.

Вскоре компания *Cisco Systems* анонсировала свой вариант технологии коммутации по меткам под названием *коммутация по тегам (Tag Switching),* которая существенно отошла от двух рассмотренных выше технологий IP Switching и CSR. В частности, для создания таблиц пересылки в коммутаторе она не опиралась на поток трафика данных и, к тому же, была специфицирована для ряда технологий уровня 2, отличных от ATM. Таким образом, технология Tag Switching оказалась намного ближе к окончательной концепции MPLS, чем механизм IP Switching. Более того, по мнению авторов, технология MPLS в значительной степени вышла из механизма Tag Switching. Сам так называемый *тег*, т.е. фиксированное количество битов, используемых для адресации, во многом аналогичен метке MPLS. Механизм Tag Switching предназначался для совместной работы с рядом протоколов нижних уровней и включал в себя *протокол распределения тегов (Tag Distribution Protocol, TDP)*. Как и в MPLS, механизм Tag Switching поддерживал образование стека тегов. Кроме более быстрого поиска адреса, новые маршрутизаторы могли обслуживать вызовы по-разному в зависимости от требуемого

качества обслуживания (при передаче речи, видео и изображений). К тому же, все маршрутизаторы производства Cisco, в которых был реализован механизм Tag Switching, были позже модернизированы и смогли поддерживать MPLS.

Как и Ipsilon, Cisco Systems выпустила RFC, в котором была описана предлагаемая технология. Однако, в отличие от Ipsilon, компания Cisco объявила о своем намерении провести стандартизацию технологии Tag Switching через IETF. В связи с этим было выпущено большое число проектов Интернет-стандартов, описывающих разные аспекты технологии Tag Switching, включая функционирование в сети ATM, с протоколами PPP и каналами 802.3, поддержку многоадресной маршрутизации, а также функций резервирования ресурсов с помощью протокола RSVP.

Практически сразу же после того, как Cisco опубликовала информацию о технологии Tag Switching и объявила об ее предполагаемой стандартизации в IETF, от корпорации IBM поступили проекты Интернет-стандартов, в которых предлагалась другая технология коммутации по меткам — *Aggregate Route-based IP Switching (ARIS).* Механизм ARIS предназначался для использования с ATM- и FR-коммутаторами, а также с устройствами коммутации на уровне 2 в локальных сетях. Устройство, в котором был реализован механизм ARIS, получило название *ARIS Integrated Switch Router (ISR)*. Технология ARIS имеет больше общих черт с технологией Tag Switching, нежели с другими уже упоминавшимися технологиями, — в обеих для создания таблиц пересылки используется трафик управляющей информации, а не трафик данных, — но при этом технология ARIS имеет некоторые существенные отличия от Tag Switching. Основное отличие состоит в том, что ARIS основан на маршрутах, а не на потоках. Маршруты в домене ARIS строятся на базе выходного узла. Конфигурируются выходные узлы домена ARIS, а затем от них распространяются маршруты в сторону входных узлов. Выходной узел может быть задан рядом идентификаторов: префиксом получателя протокола IPv4, IP-адресом выходного маршрутизатора, идентификатором маршрутизатора OSPF или идентификатором пары многоадресной передачи. Маршруты устанавливаются независимо от потоков пакетов. Многие из идей технологии ARIS перешли в окончательный стандарт MPLS.

Еще одной предшествовавшей MPLS технологией является *IP Navigator,* предложенная компанией Cascade. Cascade была затем куплена компанией Ascend, которая, в свою очередь, стала частью Lucent Technologies. В технологии IP Navigator были использованы многие идеи коммутации в IP-сетях, разработанные ранее компаниями Toshiba, Ipsilon, Cisco и IBM.

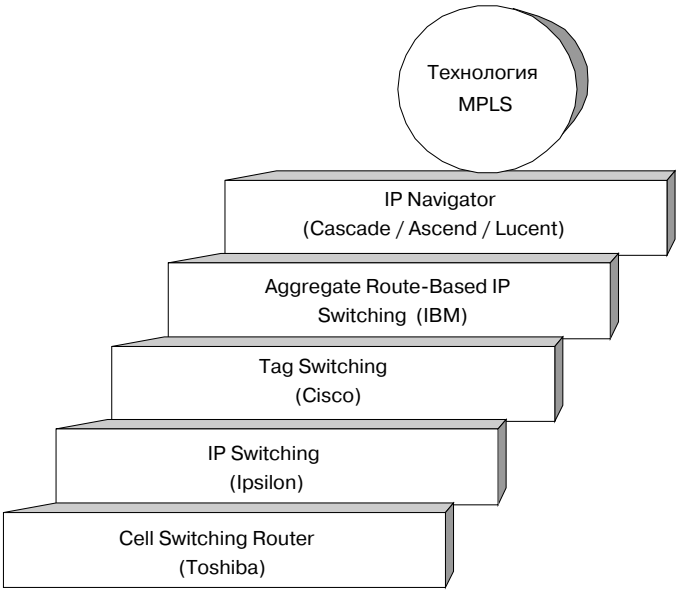


Рис. 1.2. Эволюция технологии

После публикации первой серии проектов стандартов Tag Switching 9-13 декабря 1996 года в Сан-Диего, Калифорния, состоялась рекордная за всю историю IETF по посещаемости сессия BOF, на которой Cisco Systems, IBM и Toshiba провели презентации своих технологий. Такой интерес, а также тот факт, что столь много ведущих компаний разработали во многом близкие технические предложения для решения проблемы, позволили сделать очевидный вывод о необходимости создать для стандартизации механизма коммутации по меткам специальную группу. В апреле 1997 года в Мемпфисе, Тенниси, состоялось первое заседание этой рабочей группы MPLS WG. Само название *Multiprotocol Label Switching* было принято, в первую очередь, по той, уже упомянутой нами причине, что названия *IP Switching* и *Tag Switching* ассоциировались с продуктами, выпускаемыми конкретными компаниями, и требовался нейтральный термин. Появившиеся вслед за этим документы RFC по технологии и протоколам MPLS приведены в табл. 1.1.

Таблица 1.1. Документы серии RFC, разработанные комитетом IETF по MPLS

RFC	Описание
RFC 2702	Requirements for Traffic Engineering over MPLS — определяет возможности управления трафиком в сети MPLS и алгоритмы эффективных и надежных сетевых операций в домене MPLS. Эти алгоритмы могут применяться для оптимизации использования сетевых ресурсов и для улучшения рабочих характеристик, связанных с передаваемым трафиком
RFC 3031	MPLS Architecture — специфицирует архитектуру многопротокольной коммутации по меткам MPLS
RFC 3032	MPLS Label Stack Encoding — специфицирует кодирование стека меток, а также правила и процедуры обработки разных полей стека меток, которые использует маршрутизатор LSR для передачи снабженных метками пакетов по звеньям данных протокола двухточечной связи PPP, звеньям данных локальной вычислительной сети и, возможно, по другим звеньям данных
RFC 3033	The Assignment of the Information Field and Protocol Identifier in the Q.2941 Generic Identifier and Q.2957 User-to-User Signaling for the IP — специфицирует назначение информационного поля и идентификатора протокола в общем идентификаторе Q.2941 и сигнализации «пользователь-пользователь» по Q.2957 для IP
RFC 3034	Use of Label Switching on Frame Relay Networks Specification — определяет модель и типовые механизмы использования MPLS в сетях Frame Relay. Расширяет и уточняет составляющие архитектуры MPLS и протокола распределения меток LDP в плане их использования в сетях Frame Relay
RFC 3035	MPLS using LDP and ATM Virtual Channel (VC) Switching — специфицирует процедуры, используемые при распределении меток к/от маршрутизаторов ATM-LSR, когда эти метки представляют классы эквивалентности пересылки (FEC), для которых алгоритмами маршрутизации сетевого уровня определены маршруты «по участкам». Специфицирует также инкапсуляцию MPLS, которая используется при передаче снабженных метками пакетов к/от маршрутизаторов ATM-LSR
RFC 3036	LDP Specification — определяет набор процедур протокола LDP, посредством которого LSR распределяют метки для пересылки пакетов MPLS
RFC 3037	LDP Applicability — описывает применимость протокола LDP
RFC 3038	Virtual Channel ID (VCID) Notification over ATM link for LDP — специфицирует процедуры обмена значениями идентификатора виртуального канала VCID между смежными маршрутизаторами ATM-LSR
RFC 3107	Carrying Label Information in BGP-4 — специфицирует способ, посредством которого информация о привязке метки к FEC для определенного маршрута вкладывается в то же сообщение протокола BGP, которое используется для рассылки информации о самом маршруте. Когда для выбора определенного маршрута используется протокол BGP, он может также использоваться для передачи метки MPLS, которая назначена для этого маршрута

Архитектура MPLS специфицирована в документе RFC 3031 «Multiprotocol Label Switching Architecture». Сегодня вопросами MPLS продолжают заниматься рабочие группы IETF (Routing Area Working Group — рабочая группа по маршрутизации — и MPLS Working Group — рабочая группа по MPLS) и в ATM Forum (Traffic Management Working Group — рабочая группа по управлению трафиком — и ATM-IP Collaboration Working Group — рабочая группа по совместной работе сетей ATM и IP). Основные идеи и результаты работы этих групп уже упоминались в предыдущем параграфе и будут рассмотрены в книге далее.

1.3. Классы эквивалентности пересылки FEC

Рабочими группами, упомянутыми в предыдущем параграфе, определены три основных элемента технологии MPLS: **FEC** — Forwarding Equivalency Class — класс эквивалентности пересылки, **LSR** — Label Switching Router — маршрутизатор коммутации по меткам и **LSP** — Label Switched Path — коммутируемый по меткам тракт. Начнем с классов эквивалентности пересылки.

При традиционной транспортировке пакета через сеть с использованием протокола уровня 3, не предусматривающего создания виртуальных соединений, каждый маршрутизатор на пути следования пакета самостоятельно принимает решение о том, к какому маршрутизатору переслать этот пакет дальше (способ транспортировки *hop-by-hop*). Иначе говоря, в каждом маршрутизаторе на пути следования пакета анализируется его заголовок и выполняется алгоритм сетевого уровня. Здесь и далее используется английское слово *hop* — прыжок, скачок, — а в терминах маршрутизации — одна пересылка. Под пересылкой пакета понимается его передача к ближайшему маршрутизатору из тех, что расположены на возможном пути следования этого пакета, т.е. слово «пересылка» используется как эквивалент английского слова *forwarding*.

В заголовке пакета содержится гораздо больше информации, чем нужно для того, чтобы выбрать следующий маршрутизатор. Этот выбор можно организовать проще — путем выполнения двух функций. Одна из них состоит в разделении всего множества прибывающих пакетов на классы, которые называются *классами эквивалентности пересылки FECs (Forwarding Equivalence Classes)*. Вторая ставит в соответствие каждому FEC определенное «направление» пересылки (слово «направление» написано в кавычках потому, что в сети используется режим hop-by-hop, и разные пакеты одного и того же FEC могут пересылаться к разным маршрутизаторам, то есть физические направления пересылки могут быть разными). С точки зрения выбора следующего маршрутизатора все пакеты, принадлежащие одному FEC, неразличимы.

Идея классов эквивалентности более универсальна, чем MPLS. При традиционной IP-маршрутизации тот или иной маршрутизатор тоже может считать, что два пакета принадлежат одному и тому же условному классу эквивалентности, если в его таблицах маршрутизации используется некий адресный префикс, идентифицирующий направление, в котором предполагаемые маршруты транспортировки этих двух пакетов совпадают наиболее долго. По мере продвижения пакета по сети каждый следующий маршрутизатор анализирует его заголовок и приписывает этот пакет к такому из собственных, (определенных только в данном маршрутизаторе) классов эквивалентности, который соответствует тому же направлению.

При использовании же многопротокольной коммутации по меткам MPLS пакет приписывается к определенному классу FEC только один раз, когда он попадает в сеть. Этому FEC присваивается *метка* — идентификатор фиксированной длины, передаваемый вместе с пакетом, когда тот пересылается к следующему маршрутизатору. Благодаря этому в остальных маршрутизаторах заголовок сетевого уровня не анализируется. *Метка*, установленная пограничным маршрутизатором при входе пакета в MPLS-сеть, используется как указатель входа таблицы, которая определяет очередной маршрутизатор для пересылки к нему пакета, а также новую метку для FEC, к которому относится пакет.

Таким образом, *класс эквивалентности пересылки FEC* является формой представления группы пакетов с одинаковыми требованиями к направлению их передачи, т.е. все пакеты в такой группе обрабатываются в маршрутизаторе одинаково и одинаково следуют к пункту назначения. Примером FEC могут служить все IP-пакеты с адресами пунктов назначения, соответствующими некоторому префиксу, например, 212.18.6. Возможны также FEC на основе префикса адреса и еще какого-нибудь поля IP-заголовка, например, тип обслуживания (ToS). Каждый маршрутизатор сети MPLS создает таблицу, с помощью которой определяет, каким образом должен пересылаться пакет. Эта таблица, которая называется *информационной базой меток LIB*, содержит используемое множество меток и для каждой из них — *привязку «FEC-метка»*. Метки, используемые маршрутизатором LSR при привязке «FEC-метка», подразделяются на следующие категории:

- *на платформной основе,* когда значения меток уникальны по всему тракту LSP; метки выбираются из общего пула меток, и никакие две метки, распределяемые по разным интерфейсам, не имеют одинаковых значений;
- *на интерфейсной основе,* когда значения меток связаны с интерфейсами: для каждого интерфейса определяется отдельный пул меток, из которого для этого интерфейса и выбираются метки. При этом метки, назначаемые для разных интерфейсов, могут быть одинаковыми.

Понятия «метка» и «база данных LIB» будут рассмотрены более подробно в главе 2, а сейчас важно отметить, что значение метки, как правило, изменяется по мере продвижения пакета по сети.

Метод пересылки пакетов на основе привязки «FEC-метка», принятый в MPLS, имеет ряд преимуществ перед методами, основанными на анализе заголовка блоков сетевого уровня. В частности, пересылку по методу MPLS могут выполнять маршрутизаторы, которые способны читать и заменять метки, но при этом либо вообще не способны анализировать заголовки блоков сетевого уровня, либо не способны делать это достаточно быстро.

Так или иначе, действия маршрутизатора LSR зависят от значения метки, которую он принимает от предшествующего LSR. Фактически, действия, выполняемые LSR, специфицированы в *Next Hop Level Forwarding Entry (NHLFE)*, который указывает следующий участок, операцию, которая должна быть выполнена со стеком меток (стек меток подробнее будет также рассмотрен в главе 2) и кодирование, которое следует использовать для стека в исходящем тракте. Выполняемая со стеком операция может состоять в том, что LSR должен изменить метку на вершине стека. Эта операция может потребовать, чтобы LSR просто вытолкнул верхнюю метку из стека, или вытолкнул и заменил ее новой, или просто поместил новую метку над той, которая до этого была верхней, ничего не выталкивая и не заменяя). Следующим участком для обрабатываемого пакета с метками может оказаться и тот же самый LSR. В этом случае LSR выталкивает верхнюю метку стека и пересылает пакет самому себе. В этот момент пакет может иметь еще одну метку, которую следует анализировать, или оказаться без меток, т.е. исходным пакетом IP. В последнем случае пакет пересылается в соответствии со стандартной маршрутизацией IP.

Если маршрутизатор обнаруживает, что он оказался предпоследним LSR в тракте, то он должен удалить весь стек и передать пакет в последний LSR. Благодаря этому минимизируется объем обработки, которую должен выполнить последний LSR. То, каким образом LSR определяет, что он в данном тракте предпоследний, является задачей распределения меток и используемого для этого протокола распределения меток. Но прежде поясним, что же представляет собой этот тракт.

1.4. Коммутируемые по меткам тракты LSP

При рассмотрении классов FEC в параграфе 1.3 отмечалось, что путь следования потока пакетов в сети MPLS определяется тем FEC, который установлен для этого потока во входном LSR. Такой путь носит название *коммутируемого по меткам тракта LSP* (*Label-Switched Path)* и идентифицируется последовательностью меток в LSR, расположенных на пути следования потока от отправителя к получателю.

LSP организуются либо перед передачей данных (*с управлением от программы*), либо при обнаружении определенного потока данных (*управляемые данными*).

Метки в LSP назначаются с помощью *протокола распределения меток LDP (Label Distribution Protocol)*, рассматриваемого в главе 3, причем существуют разные способы такого распределения на основе данных вспомогательных протоколов, в частности, рассмат-

риваемого в главе 4 протокола RSVP-TE. Подготавливают процесс распределения меток протоколы маршрутизации, такие как OSPF, IS-IS или BGP, рассматриваемые в главах 5, 6 и 7, соответственно. С помощью этих протоколов маршрутизации создается «древо» сети, на которое «развешиваются» метки).

Главная задача распределения меток — это организация и обслуживание трактов LSP, в том числе, определение каждой привязки «FEC-метка» в каждом LSR тракта LSP. Маршрутизатор LSR использует протокол распределения меток, чтобы информировать о привязке «FEC-метка» вышестоящий LSR. Нижестоящий LSR может непосредственно сообщать о привязке «метка-FEC» вышестоящему LSR, что называется привязкой *по инициативе нижестоящего* (*unsolicited downstream*). Кроме того, возможно извещение о привязке, передаваемое *нижестоящим по требованию* (*downstream on demand*), когда вышестоящий LSR запрашивают привязку у нижестоящего LSR. Организуемый LSP всегда является односторонним. Трафик обратного направления идет по другому LSP. Технология MPLS поддерживает следующие два варианта создания LSP:

- *последовательная маршрутизация по участкам маршрута (hop-by-hop routing)* — каждый LSR самостоятельно выбирает следующий участок маршрута для данного FEC. Эта методология сходна с той, что применяется сейчас в IP-сетях. LSR использует имеющиеся протоколы маршрутизации, такие, например, как OSPF;
- *явная маршрутизация (ER)* — сходна с методом маршрутизации со стороны отправителя. Входной LSR (т.е. LSR, от которого исходит поток данных в сети MPLS) специфицирует цепочку узлов, через которые проходит ER-LSP. Специфицированный тракт может оказаться не оптимальным. Вдоль тракта могут резервироваться ресурсы для обеспечения заданного QoS трафика данных. Это облегчает оптимальное распределение трафика по всей сети и позволяет предоставлять дифференцированное обслуживание потокам трафика разных классов, сформированных на основе принятых правил и методов управления сетью.

Рассмотрим логически завершенный (и, в определенном смысле, автономный) домен сети MPLS, изображенный на рис. 1.3. Завершенность этого домена выражается в том, что он имеет вполне определенную замкнутую границу, вдоль которой размещено четыре так называемых *пограничных узла MPLS* (MPLS edge nodes или, как их еще иногда называют, LER — Label Edge Router), обозначенных на рис. 1.3 как LSR1, LSR5, LSR6, LSR7. Помимо этих узлов, внутри домена сети MPLS (когда это не вызывает двоякого толкования, мы будем для удобства называть его просто MPLS-сетью) имеется множество маршрутизаторов, каждый из которых имеет с остальными маршрутизаторами (в том числе и с пограничными узлами)

либо прямые, либо коммутируемые связи. В последнем случае коммутация, необходимая для создания такой связи, производится другими маршрутизаторами из этого множества, которые не обязательно являются пограничными узлами MPLS и могут не иметь функций LSR. Более того, некоторые коммутируемые связи между LSR могут проходить через подсети, встроенные в рассматриваемую MPLS-сеть. Они, разумеется, не показаны в примере на рис. 1.3, где изображены только три внутренних маршрутизатора LSR2, LSR3 и LSR4.

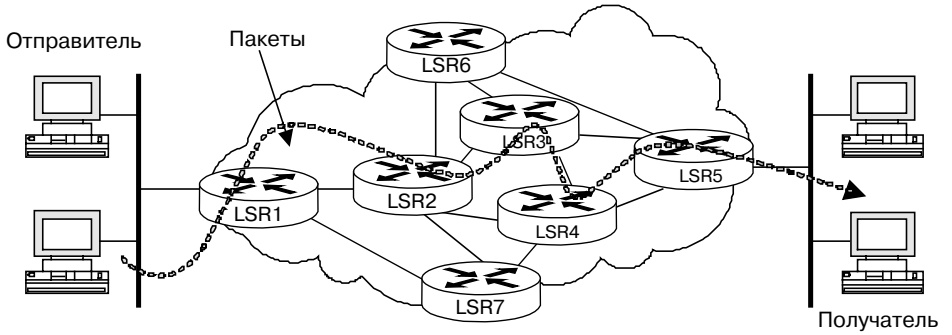


Рис. 1.3. Пример домена MPLS-сети

Напомним, что на рис. 1.3 изображен лишь упрощенный домен MPLS-сети. Пакеты, поступающие в него, могут приходить как непосредственно от отправителей (что показано на рис. 1.3), так и из смежной сети, которая может быть MPLS-сетью более высокого уровня (то есть содержать в себе рассматриваемый домен). Эти пакеты принимаются пограничным узлом MPLS (в данном случае LSR1), который является по отношению к этим пакетам *входным MPLS-узлом*. Пакеты, направляемые сетью в другую смежную сеть, передаются туда другим пограничным узлом, который является по отношению к этим пакетам *выходным MPLS-узлом* (в данном случае LSR5). В общем случае, все пакеты, транспортируемые через MPLS-сеть от входного MPLS-узла *LSR1* к выходному MPLS-узлу *LSR5*, принадлежат одному FEC и следуют по одному и тому же виртуальному *коммутируемому по меткам тракту LSP*, который может проходить через несколько LSR и маршрутизаторов без функций LSR.

Таким образом, в MPLS-сети имеются маршрутизаторы двух типов: пограничные LSR и транзитные LSR. Пограничные маршрутизаторы LSR в ряде случаев включают в себя шлюзы интерфейсов сетей разных видов (например, Frame Relay, ATM или Ethernet) и пересылают их трафик в MPLS-сеть после организации трактов LSP, а также распределяют трафик обратного направления при выходе его из MPLS-сети. К этому следует добавить, что любой MPLS-совместимый маршрутизатор должен быть способен при-

нимать в любом своем интерфейсе пакет со вставленной меткой, отыскивать ее в таблице коммутации, вставлять новую метку в соответствующем формате и затем отправлять пакет через другой интерфейс. Иными словами, пограничный LSR может коммутировать пакет с меткой от любого интерфейса к любому другому интерфейсу с заменой метки. Такой подход гораздо гибче, чем в случае ATM, так как он не ограничен исключительно каналами передачи ячеек. Пограничные маршрутизаторы выполняют основную роль в процессе назначения и удаления меток, когда трафик поступает в MPLS-сеть или выходит из нее.

При этом полезно отметить, что любой транзитный LSR способен принимать пакеты без меток, т.е. с обычными IP-заголовками. Довольно часто встречающееся в литературе утверждение, что внутри домена MPLS пакеты между транзитными LSR маршрутизируются только по меткам, не совсем верно. Для обычного MPLS-трафика это действительно так, но служебные сообщения передаются с использованием IP-заголовков. К обсуждению разделения пакетов с IP-заголовками и пакетов с метками в сети MPLS мы вернемся в главе 11, где попробуем проанализировать сегодняшние решения производителей для сетей MPLS.

К выходному узлу LSR5 (рис. 1.3) поступают потоки пакетов от нескольких входных узлов (от LSR1, LSR6 и LSR7). В промежуточных маршрутизаторах некоторые из этих потоков могут «сливаться», то есть объединяться в один общий поток пакетов, которые приобретают в этой точке слияния общий FEC. Таким образом, множество трактов LSP, идущих к одному выходному узлу, образует ветвящееся дерево, корень которого находится в этом выходном узле.

Каждый из четырех пограничных узлов выполняет, в общем случае, функции и входного, и выходного узла, так что в изображенной на рисунке MPLS-сети существует четыре дерева такого рода, которые вместе содержат $4\times(4-1)=12$ трактов LSP. Ясно, что через один промежуточный маршрутизатор LSR может проходить несколько LSP, в том числе, LSP, принадлежащих разным деревьям. Если учесть, к тому же, что физическая топология сети отличается от топологии виртуальной сети LSP (и еще раз вспомнить про режим hop-by-hop), то станет ясно, что на практике могут возникать случаи «закольцовывания» путей прохождения пакетов, и, следовательно, в MPLS-сетях нужно предусматривать меры обнаружения и/или предотвращения таких случаев. В главе 3, посвященной протоколу LDP, достаточно внимательно рассматриваются средства борьбы с такого рода петлями. К рис. 1.3 мы еще вернемся в следующих главах книги, а сейчас, в заключительном параграфе главы, попробуем свести воедино основные рассмотренные в ней понятия.

1.5. Основные понятия

Итак, MPLS может рассматриваться как совокупность технологий, которые, работая совместно, обеспечивают доставку пакетов от отправителя к получателю контролируемым, эффективным и предсказуемым способом. В MPLS для пересылки пакетов используются рассмотренные выше коммутируемые по меткам тракты LSP, которые были организованы с помощью рассматриваемых в следующих главах протоколов маршрутизации и сигнализации уровня 3. Основные специальные термины MPLS сведены в табл. 1.2.

Таблица 1.2. Основные термины MPLS

Понятие	Пояснение
FEC — Forwarding Equivalence Class) класс эквивалентности пересылки	Множество пакетов, которые пересылаются одинаково, например, с целью обеспечить заданное QoS
Label — метка	Короткий идентификатор фиксированной длины, определяющий принадлежность пакета тому или иному FEC
Label swapping — замена меток	Замена метки принятого узлом сети MPLS пакета новой меткой, связанной с тем же FEC, при пересылке этого пакета к нижестоящему узлу
LER — (MPLS edge router — пограничный узел сети MPLS)	Пограничный узел сети MPLS, который соединяет домен MPLS с узлом, находящимся вне этого домена
Loop detection — выявление закольцованных маршрутов	Метод, позволяющий обнаружить, что пакет прошел через узел более одного раза
Loop prevention — предотвращение образования закольцованных маршрутов	Метод выявления и устранения закольцованных маршрутов
LSP — (Label Switched Path) коммутируемый по меткам тракт	Приходящий через один или более LSR тракт, по которому следуют пакеты одного и того же FEC
ER — LSP — (explicitly routed LSP) — LSP с явно заданным маршрутом	Тракт LSP, который организован способом, отличным от традиционной маршрутизации пакетов IP
LSR — (Label Switching Router) маршрутизатор коммутации по меткам	Маршрутизатор, способный пересылать пакеты по технологии MPLS
MPLS domain — домен MPLS	Совокупность узлов MPLS, между которыми существуют непрерывные LSP
MPLS egress node — выходной узел сети MPLS	Последний MPLS-узел в LSP, направляющий исходный пакет к адресату, который находится вне MPLS-сети
MPLS ingress node — входной узел сети MPLS	Первый MPLS-узел в LSP, принимающий исходный пакет и помещающий в него метку MPLS

В дополнение к приведенным в таблице рассмотрим еще некоторые базовые понятия, уже упоминавшиеся выше и весьма важные для понимания принципов работы технологии MPLS, такие как *пересылка, коммутация* и *маршрутизация*.

Маршрутизация — это выбор маршрута или того его элемента, который ведет к ближайшему узлу, входящему в этот маршрут, как правило, функция уровня 3 модели OSI. Очень важно правильно воспринять это понятие, потому что технология MPLS дополняет его общепринятую трактовку и «вклинивается» между сетевым уровнем 3 и уровнем звена данных 2. Маршрутизация в традиционном смысле, без MPLS, представляет собой процесс определения следующего участка, по которому должен пойти пакет в направлении получателя, путем анализа заголовка сетевого уровня. Процесс маршрутизации в каждом маршрутизаторе использует различные протоколы и алгоритмы маршрутизации для отыскания маршрутов и создания таблицы пересылки, которая используется уже в плоскости пересылки данных, как это показано на рис. 1.1.

Коммутация — это выбор исходящего порта в соответствии с результатом маршрутизации и создание связи между входящим и выбранным исходящим портами, т.е. создание внутри узла условий (можно сказать, внутриузлового пути) для отправки пакета по уже выбранному маршруту. Как правило, это — функция уровня 2 модели OSI. В традиционном смысле *коммутатор* представляет собой устройство, которое принимает пакеты во входных портах, анализирует информацию заголовка уровня 2 (звена данных), использует свои внутренние таблицы коммутации, чтобы создать условия для отправки пакетов к надлежащим выходным портам. Обычно коммутаторы работают быстрее маршрутизаторов, но имеют меньше функциональных возможностей. Добавление в коммутатор функций MPLS превращает его в LSR.

Пересылка — это использование созданных посредством коммутации условий (внутриузлового пути) для того, чтобы передать пакет из входящего порта по упомянутому маршруту через выбранный при коммутации исходящий порт.

Это несколько упрощенные определения, но они являются хорошей отправной точкой для обсуждения в следующей главе того, что же такое метки MPLS, как именно они распределяются и как по ним производится коммутация.

Глава 2
Метки и функционирование MPLS

2.1. Коммутация по меткам

Из самого определения MPLS видно, что метки — основа основ этой технологии. Именно с метками выполняются процедуры их распределения по маршрутизаторам LSR и процедуры создания трактов LSP, по которым будут следовать пакеты MPLS. После распределения меток и создания трактов LSP может выполняться главная функция MPLS — пересылка снабженных метками пакетов по сети MPLS. Помимо этой функции должны решаться и вспомогательные задачи, связанные с метками, а именно, контроль времени сохранения меток, упорядочение меток и обработка ошибок.

В предыдущей главе *домен сети MPLS* определен как совокупность узлов LSR, между которыми созданы непрерывные LSP. Там же, на рис. 1.3 представлен такой домен. Рассмотрим подробнее основные шаги алгоритма, который выполняется в отношении пакетов данных в нем. Для этого скорректируем рис. 1.3, приведя его к виду рис. 2.1. На базе рис. 2.1 и подводя итоги главы 1, перечислим основные шаги, которые необходимы, чтобы обеспечить прохождение пакета данных через домен MLPS:

Шаг 1. Создание и распределение меток. До начала передачи через сеть MPLS пакетов трафика любого вида маршрутизаторы LSR устанавливают соответствие между метками и FEC в своих таблицах. В следующей главе будет показано, как нижестоящие маршрутизаторы с помощью сигнализации LDP, использующей транспортный протокол TCP, производят распределение меток и привязку их к классам FEC. Кроме того, производится согласование характеристик трафика и функциональных возможностей MPLS.

Напомним, что значения меток могут выбираться и рассылаться либо заранее, до передачи данных (кривая 2 на рис. 2.1), либо генерироваться как пакеты, принадлежащие поступающему в сеть MPLS определенному потоку данных или трафику определенного класса (кривая 3). Эти два подхода к назначению меток, как отмечалось в предыдущей главе, называются, соответственно, назначением *с управлением от программы* и назначением, *управляемым трафиком* (данными). Но после того как домен коммутации по меткам сконфигурирован для обслуживания пакетного трафика, пересылаемого через MPLS-сеть с помощью меток, все пакеты обрабатываются одинаково.

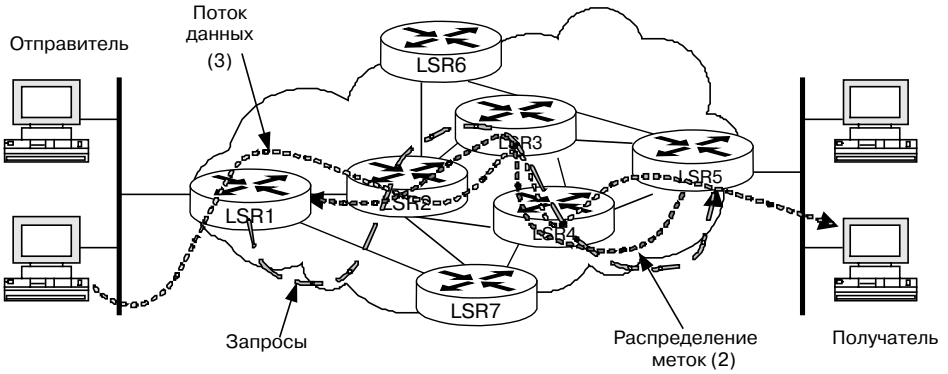


Рис. 2.1. Коммутация по меткам в MPLS-сети

Шаг 2. Создание таблицы в каждом LSR. При получении данных о привязке меток к FEC каждый маршрутизатор LSR создает записи в таблице LIB. Содержимое таблицы отражает соответствие между метками и FEC и ставит в соответствие каждой паре «входной интерфейс, входящая метка» пару «выходной интерфейс, исходящая метка». При любом новом согласовании привязки меток к FEC записи в таблице обновляются. Структура таблицы будет рассмотрена далее в этой главе. Следует напомнить, что таблицы меток, согласно которым каждый пакет направляется по соответствующему тракту LSP, всегда должны быть заданы до того, как пакет начнет свой путь по сети. Кроме того, коммутируемый по меткам тракт — всегда односторонний. Если необходимо, чтобы пакетный трафик между двумя пограничными LSR проходил и в противоположном направлении, следует создать два тракта.

Шаг 3. Создание коммутируемого по меткам тракта LSP. Как показано линией 3 на рис. 2.1, тракты LSP создаются в направлении, обратном созданию записей в таблицах LIB. Сами LSP и процедура их создания уже были кратко рассмотрены в параграфе 1.4. Напомним, что каждый LSR получает метку от нижестоящего маршрутизатора. LSP создается путем последовательной маршрутизации по

участкам, а если требуется оптимизация распределения трафика, для определения тракта используется протокол CR-LDP, гарантирующий выполнение требований к QoS/CoS, или протокол RSVP-TE.

Шаг 4. Табличный поиск и инкапсуляция метки в пакет. Входной маршрутизатор (LSR1 на рис. 2.1), определив, какому FEC принадлежит принятый им извне пакет, использует таблицу LIB, чтобы отыскать нужную привязку «FEC-метка», и инкапсулирует эту метку способом, соответствующим применяемой на уровне 2 технологии, как это будет показано ниже.

Шаг 5. Пересылка пакета. Рассмотрим прохождение пакета от входного маршрутизатора LSR1 к выходному маршрутизатору LSR5. Отметим, что LSR1 может не иметь метки для этого пакета. В таком случае он находит следующий маршрутизатор по IP-адресу. Пусть следующим маршрутизатором для LSR1 является LSR2. Маршрутизатор LSR1 инициирует запрос метки от LSR2. Полученную метку LSR1 вставляет в пакет и пересылает его к LSR2. Каждый следующий LSR (в данном случае — LSR3 и LSR4) анализирует метку, содержащуюся в принятом пакете, заменяет ее исходящей меткой и пересылает пакет дальше. Когда пакет достигает LSR5, тот удаляет метку из пакета, поскольку пакет покидает домен MPLS, и доставляет пакет адресату. Тракт LSP, по которому проходит пакет, показан прерывистыми линиями 3.

Рассмотрим подробнее используемый на *шаге 5* алгоритм замены меток при пересылке пакетов через домен MPLS. Получив пакет, маршрутизатор LSR извлекает из него метку и использует ее в качестве индекса в своей таблице пересылки. Как только найдена запись, в которой значение входящей метки равно значению метки, извлеченной из пакета, маршрутизатор, согласно подзаписи этой записи, заменяет входящую метку в пакете исходящей меткой и пересылает пакет через выходной интерфейс, указанный в подзаписи, к следующему LSR, также указанному в этой подзаписи. Если подзапись указывает определенную выходную очередь, маршрутизатор ставит пакет именно в эту очередь. Простота алгоритма пересылки пакетов, используемого в MPLS, обуславливает простую и экономичную его реализацию в аппаратном обеспечении, что, в свою очередь, позволяет повысить производительность пересылки без использования дорогостоящей аппаратуры.

Если LSR поддерживает не одну, а несколько таблиц (по одной для каждого из своих интерфейсов), то единственное изменение алгоритма состоит в том, что после получения пакета LSR предварительно выбирает ту таблицу, которая будет использоваться для обработки пакета. Выбор таблицы производится согласно идентификатору интерфейса, через который пакет был получен.

Таким образом, метка, переносимая в составе пакета, всегда передает семантику пересылки, потому что она однозначно определяет нужную запись в таблице, которую ведет LSR, и потому что эта запись содержит информацию о том, куда пересылать пакет. В качестве опции метка может также передавать семантику резервирования ресурсов, поскольку определяемая ею запись может содержать информацию о том, какие ресурсы будет использовать пакет, например, ту выходную очередь, в которую он должен ставиться. Когда метка переносится в заголовке ATM или Frame Relay, она должна передавать семантику как пересылки, так и резервирования ресурсов. Когда метка переносится в специальном заголовке, информация о том, какие ресурсы будут доступны пакету, может кодироваться как часть этого заголовка, а не переноситься меткой, которая служит в этом случае только для пересылки. Еще один возможный вариант состоит в совместном использовании для кодирования этой информации как метки, так и «не меточной» части специального заголовка. И, разумеется, даже при использовании специального заголовка метка может передавать и семантику пересылки, и семантику резервирования ресурсов.

Теперь рассмотрим алгоритм шага 5 на примере рис. 2.2.

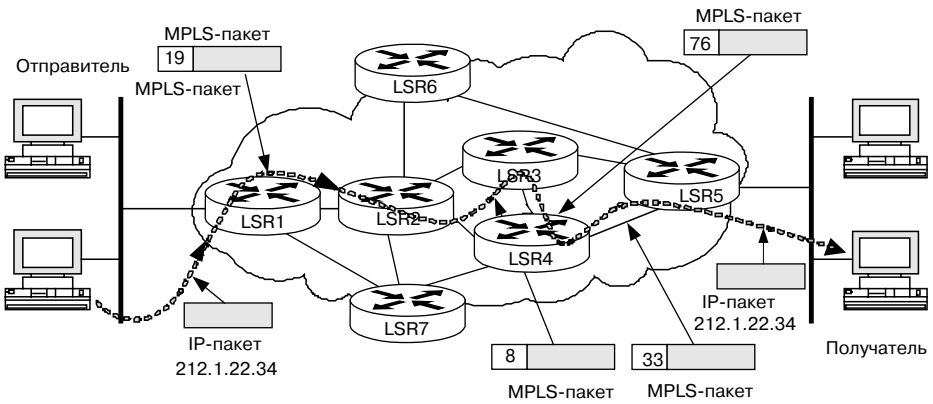


Рис. 2.2. Прохождение трафика в домене MPLS

Входной пограничный маршрутизатор LSR1 распознает, что поступивший к нему извне IP-пакет уровня 3 с адресом 212.1.22.34 должен быть, согласно FEC этого пакета, передан по LSP 1-2-3-4-5, добавляет к пакету MPLS-метку 19 и пересылает его к транзитному маршрутизатору LSR2, где с помощью таблицы пересылки входящая метка 19 заменяется исходящей меткой 8, и пакет передается по тому же LSP дальше, к LSR3. Транзитный LSR3 заменяет входящую метку 8, которую имеет поступивший к нему пакет, исходящей меткой 76. Затем транзитный LSR3 использует соответствующий выходной интерфейс для передачи пакета к другому транзитному

маршрутизатору LSR4, который выполняет аналогичную процедуру с пакетом, поступившим через его входной интерфейс с меткой 76: снабжает этот пакет новой меткой 33 и передает его к выходному пограничному маршрутизатору LSR5. В маршрутизаторе LSR5 метка 33 удаляется, и пакет пересылается к получателю с адресом 212.1.22.34 с помощью традиционной маршрутизации на сетевом уровне, покидая изображенный на рис. 2.2 (а также на рис. 2.1 и на рис. 1.3) домен MPLS.

Таким образом, использование меток является основным механизмом переноса трафика через сеть MPLS. Метки вводятся в пакеты при их входе в сеть MPLS, заменяются новыми метками по мере продвижения пакетов от узла к узлу и удаляются на выходе пакетов из этой сети.

Из описанного примера видно, что механизм замены меток имеет ряд преимуществ перед механизмом маршрутизации по участкам, который используется в традиционных IP-маршрутизаторах. Он более прост и эффективен. Анализ заголовка пакета выполняется только один раз — во входном LSR1. Замена меток внутри домена MPLS выполняется быстро, поскольку LSR просто распознает метку и заменяет ее новой. Выходной LSR5 определяет, что пакет находится на границе домена, удаляет метку из пакета и пересылает его в домен получателя уже на основе другой информации — заголовка IP-пакета сетевого уровня (рис. 2.1). Хотя технология MPLS — это не просто замена значений меток, но понятие метки и алгоритм их замены меток являются основой MPLS.

2.2. Структура метки

Метка представляет собой короткий элемент фиксированной длины, используемый для локальной идентификации класса эквивалентности пересылки FEC. По значению метки пакета в каждом узле маршрута, по которому он передается, определяется его принадлежность определенному FEC.

Структура метки MPLS представлена на рис. 2.3. Ее длина составляет 32 бита (4 байта): 12 битов — заголовок и 20 битов — значение метки. Заголовок метки состоит из трех полей: 3-битового поля Exp, которое может служить для обозначения класса обслуживания, S-бита признака «дна» стека и 8-битового поля «время жизни» TTL (Time-to-Live).

20-битовое *поле метки* содержит значение MPLS-метки, которое может быть любым числом в диапазоне от 0 до 2^{20}-1, за исключением резервных значений (0, 1, 2, 3 и др.), определением использования которых занимается рабочая группа MPLS в составе комитета IETF.

Рис. 2.3. Структура MPLS-метки

Теперь рассмотрим содержимое полей заголовка метки.

Поле *экспериментальных битов (EXP)* содержит три бита, которые резервированы для дальнейших исследований и экспериментирования. В настоящее время проводится работа, направленная на создание согласованного стандарта использования этих битов для поддержки дифференцированного обслуживания разнотипного трафика и идентификации класса обслуживания. Первоначально поле так и называлось — «Класс обслуживания» (CoS), и это название до сих пор встречается в литературе. При предоставлении дифференцированных услуг MPLS-сети это поле может указывать определенный класс обслуживания, например, аналогичный классам DiffServ. Чтобы обеспечить сквозное качество IP-услуг, на границе MPLS-сети можно скопировать в поле CoS поле IP-приоритета с учетом того, что поле CoS в MPLS-заголовке содержит всего 3 бита, и в нем может передаваться только 3-битовое поле IP-приоритета, а 6-битовое поле кода дифференцированной услуги (Differentiated Service Code Point — DSCP) — нет. При необходимости информация CoS может передаваться и в виде одной из меток MPLS-стека, т.к. поле метки имеет размер 20 битов и способно вместить как поле IP-приоритета, так поле DSCP.

Бит S является средством поддержки иерархической структуры стека меток MPLS. В заголовке последней (т.е. самой глубокой или нижней) метки бит S=1, а во всех остальных метках в стеке бит S=0. Подробнее стек меток рассматривается в следующем параграфе.

Поле *времени жизни (TTL)* в заголовке MPLS работает аналогично полю TTL в IP-дейтаграмме; это поле является механизмом, предотвращающим возможность бесконечной циркуляции пакетов по сети вследствие образования закольцованных маршрутов. Байт TTL находится в конце заголовка метки и, как представлено на рис. 2.3, занимает биты 24 — 31. Диапазон значений этого поля составляет от 0 до 255. Поясним его назначение подробнее.

Как уже упоминалось, протокол MPLS поддерживает два способа определения маршрута, по которому будет следовать пакет. Первый из них аналогичен способу hop-by-hop, используемому сегодня в IP-сетях, и предполагает, что каждый маршрутизатор самостоя-

тельно выбирает, куда переслать принятый им пакет, так что маршрут, по которому транспортируется пакет, оказывается, вообще говоря, случайным.

Второй способ основан на том, что маршрутизаторы на пути следования пакета действуют не самостоятельно, а в соответствии с инструкциями, полученными от одного из LSR тракта LSP (обычно — от верхнего LSR, что позволяет совместить процедуру «раздачи» этих инструкций с процедурой распределения меток). Таким образом, маршрут следования пакетов однозначно определяется заранее.

Первый способ имеет ряд преимуществ, связанных со сравнительной простотой его реализации, но не исключает феномен «закольцовывания» маршрутов, для борьбы с которым и используется поле TTL. Входной LSR помещает в это поле максимальное число LSR, через которые имеет право пройти пакет, а каждый маршрутизатор, через который пакет проходит, уменьшает это число на единицу. Если пакет не предназначен узлу, где он оказался в тот момент, когда его поле TTL приняло нулевое значение, этот пакет просто отбрасывается.

Таким образом, значение TTL, установленное на границе MPLS-сети, уменьшается по мере прохождения пакетом каждого следующего LSR. Во время введения в пакет метки поле TTL протокола IP по умолчанию копируется в поле TTL MPLS. Это позволяет служебной программе *traceroute* показывать все промежуточные узлы сети MPLS в том случае, когда при достижении точки назначения пакет проходит домен MPLS. Если нужно, чтобы при входе в MPLS-сеть поле TTL протокола IP не копировалось в поле TTL MPLS, используется команда **no mpls ip propagate-ttl**. В этом случае значение поля TTL MPLS устанавливается равным 255, и служебная программа *traceroute* не показывает промежуточные узлы в MPLS-сети, а отображает весь домен MPLS как один IP-переход.

2.3. Стек меток MPLS

Спецификация кодирования стека меток MPLS определена документом RFC 3032 «MPLS Label Stack Encoding», написанным Эриком Розеном, Яковом Рехтером, Даниэлем Таппеном, Дино Фариначчи, Ги Федорковым, Тони Ли и Алексом Конта и опубликованным в январе 2001 года. Этот документ содержит детальную информацию о метках MPLS и о том, как они используются с различными сетевыми технологиями, а также определяет ключевое для технологии MPLS понятие — *стек меток*. Возможность иметь в пакете более одной метки в виде стека позволяет создавать иерархию меток, что открывает дорогу многим приложениям MPLS.

MPLS может выполнять со стеком следующие операции: помещать метку в стек, удалять метку из стека и заменять метку. Эти операции могут использоваться для слияния и разветвления информационных потоков. Операция *помещения метки в стек (push operation)* добавляет новую метку поверх стека, а операция *удаления метки из стека (pop operation)* удаляет верхнюю метку стека.

Функциональные возможности стека MPLS позволяют объединять несколько LSP в один. К стеку меток каждого из этих LSP сверху добавляется общая метка, в результате чего образуется агрегированный тракт MPLS. В точке окончания такого тракта он разветвляется на составляющие его индивидуальные LSP. Так могут объединяться тракты, имеющие общую часть маршрута. Следовательно, MPLS способна обеспечивать иерархическую пересылку, что может стать важной и востребованной функциональной возможностью. При ее использовании не нужно переносить глобальную маршрутную информацию, что делает сеть MPLS более стабильной и масштабируемой, чем сеть с традиционной маршрутизацией.

Согласно рассматриваемым в следующем параграфе правилам инкапсуляции меток, за меткой MPLS в пакете всегда должен следовать заголовок сетевого уровня. Так как MPLS начинает работу с верхнего уровня стека, этот стек используется по принципу LIFO «последним пришел, первым ушел».

Пример четырехуровневого стека меток представлен на рис. 2.4. Здесь заголовок MPLS №1 был первым заголовком MPLS, помещенным в пакет, затем в него были помещены заголовки №2, №3 и, наконец, — заголовок №4. Коммутация по меткам всегда использует верхнюю метку стека, и метки удаляются из пакета так, как это определено выходным узлом для каждого LSP, по которому следует пакет. Рассмотренный в предыдущем параграфе бит S имеет значение 1 в нижней метке стека и 0 — во всех остальных метках. Это позволяет привязывать префикс к нескольким меткам, другими словами — к *стеку меток (Label Stack)*. Каждая метка стека имеет собственные значения поля EXP, S-бита и поля TTL.

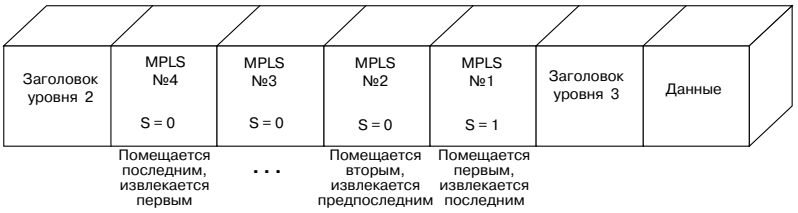


Рис. 2.4. Пример четырехуровневого стека меток MPLS

2.4. Инкапсуляция меток

При обсуждении в главе 1 «многопротокольности» технологии MPLS была подчеркнута возможность использовать эту технологию с самыми разными протоколами уровней 2 и 3. Там же говорилось, что MPLS образует уровень, который «вставляется» между сетевым уровнем и уровнем звена данных и позволяет разнообразным протоколам как того, так и другого уровня функционировать совместно.

При использовании протоколов коммутации на уровне звена данных, таких как ATM и Frame Relay, верхняя MPLS-метка вписывается в поле идентификаторов этих протоколов. Далее будет показано, как при использовании ATM для размещения MPLS-метки используется поле VPI/VCI, а при использовании Frame Relay — поле DLCI. В тех случаях, когда MPLS обеспечивает пересылку IP-пакетов сетевого уровня и когда технология уровня звена данных не поддерживает собственное поле меток, MPLS-заголовок должен *инкапсулироваться* между заголовками уровня звена данных и сетевого уровня.

Механизм инкапсуляции переносит один или более протоколов верхних уровней внутри полезной нагрузки дейтаграммы инкапсулирующего протокола. В сущности, вводится новый заголовок, который делает из инкапсулированного заголовка и поля данных новое поле данных. Общая модель инкапсуляции представлена на рис. 2.5, где подразумевается, что инкапсуляция MPLS может быть использована с любой технологией уровня 2. Метка MPLS может быть помещена в существующий формат заголовка уровня 2, как в случае ATM или FR, или вписана в специальный заголовок MPLS, как в случае Ethernet или PPP. Во всех случаях любые дополнительные метки находятся между верхней меткой стека и IP-заголовком уровня 3.

Показанный на рис. 2.5 заголовок MPLS часто называют *shim header* (заголовком-клином), подчеркивая в метафорической форме тот факт, что этот заголовок «уровня 2.5» вклинивается в пакете между заголовками уровня звена данных и сетевого уровня.

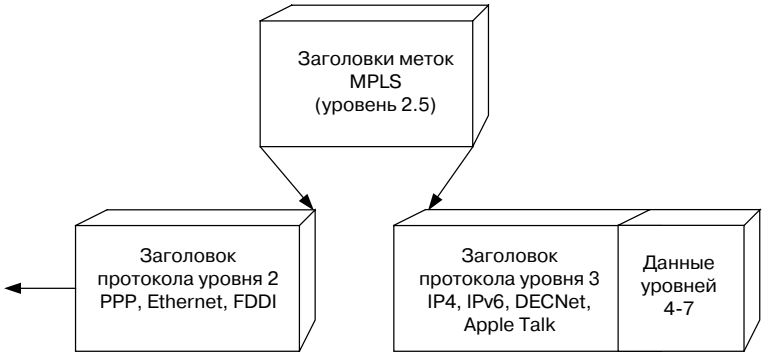


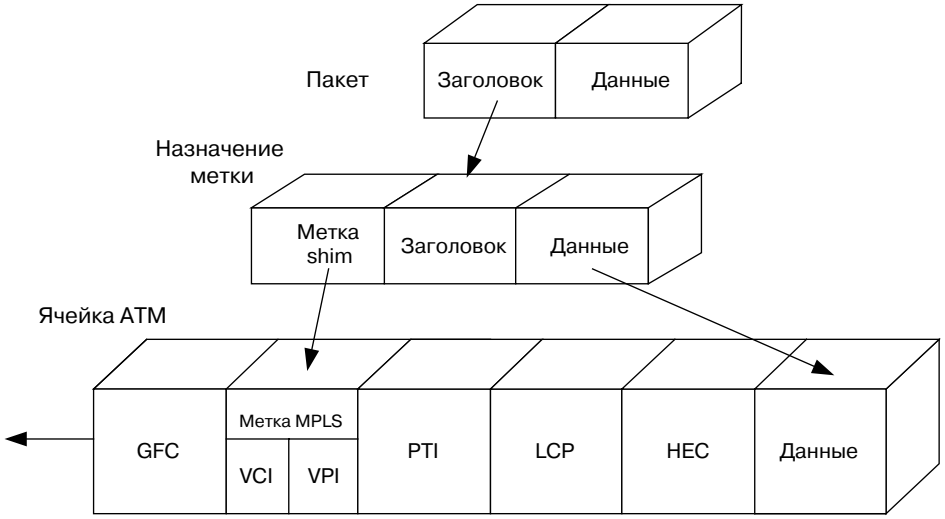
Рис. 2.5. Принцип инкапсуляции заголовка MPLS

Итак, одной из самых сильных сторон технологии MPLS (и потому отраженной в ее названии) является то, что она может использоваться совместно с различными протоколами уровня 2. Среди этих протоколов — ATM, Frame Relay, PPP и Ethernet, FDDI и другие, предусмотренные документами по MPLS.

Покажем, как метка может вписываться в заголовок уровня звена данных (VCI/VPI для сети ATM, DLCI для сети Frame Relay и т.п.) или «вставляться» между заголовками уровня звена данных и сетевого уровня. С самого начала рабочая группа IETF MPLS решила, что во всех случаях, когда это возможно, MPLS должна использовать имеющиеся форматы. По этой причине информация метки MPLS может передаваться в пакете несколькими разными методами:

- как часть заголовка второго уровня ATM, когда информация метки передается в идентификаторах виртуального канала VCI и виртуального тракта VPI, что показано на рис. 2.6;
- как часть кадра AAL5 уровня адаптации ATM (ATM Adaptation Layer 5) перед сегментацией и сборкой SAR (Segmentation and Reassembly), что выполняется в ATM-окружении в случае, когда эта информация содержит данные о стеке меток (несколько полей MPLS-меток);
- как часть заголовка второго уровня Frame Relay, когда информация метки передается в идентификаторах DLCI (Data Link Connection Identifier), что изображено на рис. 2.7;
- как новая 4-байтовая метка, называемая клином или прокладкой (shim), которая вставляется между заголовками второго и третьего уровней, что показано на рис. 2.5, — во всех остальных случаях.

Размещение метки MPLS в заголовке ATM представлено на рис. 2.6.

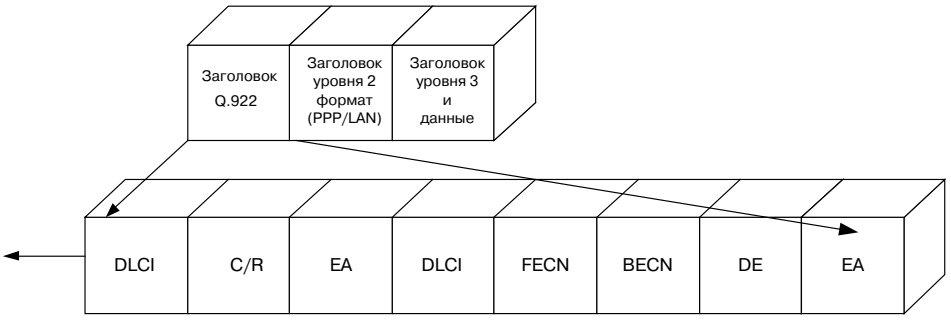


Заголовок ATM (5 байтов):
GFC – поле общего управления потоком (4 бита) для передачи информации о перегрузке
VCI – поле идентификатора виртуального канала (16 битов)
VPI – поле идентификатора виртуального тракта (8 битов)
PTI – поле идентификатора типа полезной нагрузки (3 бита): пользовательские данные или трафик техобслуживания
CLP – поле приоритета ячейки (1 бит): низкий или высокий приоритет
HEC – поле контроля ошибок в заголовке (8 битов) для исправления единичных ошибок или обнаружения множественных ошибок в заголовке ячейки

Рис. 2.6. MPLS-метка, передаваемая в полях VPI/VCI заголовка ATM

Использование MPLS поверх ATM сейчас довольно распространено, особенно для транспортировки по сетям ATM трафика IP. ATM-коммутаторы, которые конфигурированы для поддержки MPLS (ATM-LSR), выполняют протоколы маршрутизации TCP/IP и используют пересылку данных в ATM фиксированной длины 53 байта. Внутри этих ATM-LSR верхняя метка MPLS помещается в поля VCI/VPI заголовка ячейки ATM, а данные о стеке меток MPLS помещаются в поле данных ячеек ATM.

MPLS в сетях Frame Relay была развернута рядом крупных поставщиков услуг и до сих пор широко используется. Подобно ATM, FR-коммутаторы, поддерживающие MPLS, используют протоколы маршрутизации TCP/IP для пересылки данных под управлением FR. При использовании FR текущая метка помещается в поле идентификатора канала звена данных DLCI в заголовке FR. Любые дополнительные записи в стек меток MPLS переносятся после заголовка FR, но до заголовка сетевого уровня, содержащегося в поле данных кадра FR. Стандарт MPLS позволяет FR использовать адрес Q.922 длиной либо 2 октета, либо 4 октета. Формат представлен на рис. 2.7.



Примечание: Длина поля DLCI может составлять 10,17 или 23 бита

Рис. 2.7. Размещение метки MPLS в кадре FR

В отношении ячеек ATM и кадров Frame Relay договорились использовать для MPLS имеющиеся форматы заголовков, а во всех остальных случаях — собственную метку MPLS — «прокладку» между заголовками второго и третьего уровней. Отсюда видно, что MPLS позволяет создавать новые форматы меток без изменения протоколов маршрутизации, а потому распространение этой технологии на вновь появляющиеся виды оптического транспорта, такие как компактное мультиплексирование с разделением по длине волны DWDM (Dense Wave Division Multiplexing) и оптическая коммутация, представляет собой относительно простую задачу.

Принцип, представленный на рис. 2.5, подходит для каналов типа «точка-точка» (Point-to-Point — PPP) и для локальных сетей Ethernet (всех типов). Подобным методом можно передать одну MPLS-метку или стек меток.

Протокол PPP фактически представляет собой семейство родственных протоколов IETF, используемое для передачи многопротокольных дейтаграмм по двухточечным каналам связи. PPP определяет метод инкапсуляции дейтаграмм разных протоколов сетевого уровня, протокола управления звеном данных LCP и набора протоколов управления сетью NCP. Для использования MPLS с управлением коммутацией по меткам через звено PPP был определен специальный протокол, который управляет передачей пакетов MPLS по каналу PPP. Этот протокол обозначается аббревиатурой MPLSCP. Полю пакета протокола PPP присваивается специальное значение, которое указывает, что этот пакет содержит управляющий пакет MPLSCP (шестнадцатеричное число 8281). Когда по каналу PPP передаются пакеты данных MPLS, этому полю присваивается шестнадцатеричное значение 0281 в случае одноадресной передачи пакетов MPLS и шестнадцатеричное значение 0283 в случае многоадресной рассылки пакетов MPLS. Формат показан на рис. 2.8.

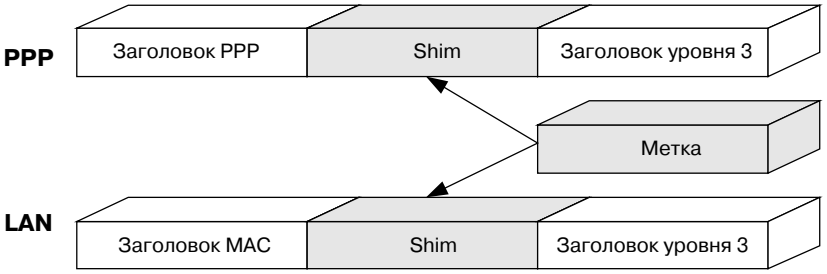


Рис. 2.8. Формат для введения MPLS-метки в пакет PPP и в кадр Ethernet

Когда пакеты *MPLS передаются по Ethernet*, в каждом кадре Ethernet переносится только один снабженный меткой пакет. Метка помещается между заголовком уровня звена данных и заголовком сетевого уровня. Использование MPLS в сетях Ethernet, особенно, в городских сетях, является еще одной перспективной возможностью MPLS. В стандарт Ethernet вносятся изменения, позволяющие увеличить скорость и дальность передачи Ethernet-пакетов. В настоящее время начинают использоваться Ethernet-интерфейсы на скоростях 10 Гбит/с, а в скором времени появятся Ethernet-интерфейсы и на еще больших скоростях.

Значения поля Ethertype в заголовке Ethernet — шестнадцатеричное 8847 для одноадресной передачи и шестнадцатеричное 8848 для многоадресной рассылки. Эти значения могут использоваться либо со стандартным форматом Ethernet, либо с форматом 802.3 LLC/SNAP.

К сожалению, добавление MPLS-метки или стека меток к 1492-байтовому пакету может привести к необходимости его фрагментации. При передаче пакетов MTU-размера (Maximum Transmission Unit — максимально возможный размер передаваемого блока данных) с MPLS-меткой протокол управления передачей TCP определяет, что в MPLS-сети нужно произвести фрагментацию. Однако следует отметить, что многие Ethernet-каналы поддерживают передачу 1500-байтовых или 1508-байтовых пакетов. Более того, в большинстве сетей пакеты с метками обычно передаются по ATM- или PPP-каналам, а не по сегментам локальных сетей.

Итак, метка может быть помещена в пакет разными способами — вписываться в специальный заголовок, помещаемый либо между заголовками уровня 2 и уровня 3, либо в свободное и доступное поле заголовка одного из этих двух уровней, если, конечно, таковое имеется. Очевидно, что вопрос о том, куда следует помещать заголовок, содержащий метку, должен согласовываться между объектами, ее использующими.

Несмотря на то, что в документе по MPLS специфицирована работа MPLS с рядом протоколов сетевого уровня (а теоретически, поскольку применяется специальный заголовок MPLS, — с любым сетевым протоколом), первоначально она стала использоваться с текущей версией протокола IP, т.е. с протоколом IPv4. Продолжается работа над протоколом IPv6, и этот протокол медленно, но верно реализуется на практике. Как и в случае IPv4, IPv6 помещает заголовок MPLS перед заголовком сетевого уровня и, в зависимости от используемого протокола уровня 2, либо в виде специального заголовка, либо внутри заголовка уровня звена данных.

Первые разработчики MPLS в список поддерживаемых протоколов включили протоколы IPX, Apple Talk, DECNet и CLNP, хотя в настоящее время к ним не проявляется большого интереса. Значительная часть текущих работ в области MPLS сконцентрирована на технологиях более низких уровней и на использовании оптических средств передачи пакетов по сети IP. Так что не будет ошибкой сказать, что IP станет преобладающим протоколом сетевого уровня, используемым с MPLS.

2.5. Таблицы пересылки

В главе 1 отмечалось, что когда пакет MPLS попадает в маршрутизатор коммутации по меткам LSR, этот маршрутизатор просматривает имеющуюся у него таблицу с так называемой *информационной базой меток LIB (Label Information Base)* для того, чтобы принять решение о дальнейшей обработке пакета (рис. 2.9). Эту информационную базу иногда называют также *Next Hop Label Forwarding Entry (NHLFE)*, и согласно RFC 3031 в нее обычно входит следующая информация:

- операция, которую надо произвести со стеком меток пакета (заменить верхнюю метку стека, удалить верхнюю метку, поместить поверх стека новую метку);
- следующий маршрутизатор в LSP, причем «следующим» может быть тот же самый LSR;
- используемая при передаче пакета инкапсуляция на уровне звена данных;
- способ кодирования стека меток при передаче пакета;
- другая информация, относящаяся к пересылке пакета.

Содержащая эту информацию таблица пересылки, которую ведет каждый LSR, как, впрочем, и любые другие таблицы, представляет собой последовательность записей. Каждая *запись* таблицы пересылки LSR состоит из входящей метки и одной или более подзаписей, причем каждая *подзапись* содержит значение исходящей

метки, идентификатор выходного интерфейса и адрес следующего маршрутизатора в LSP. Пример простой таблицы пересылки LIB представлен на рис. 2.9.

Входящая метка	Первая подзапись	Вторая подзапись
Значение входящей метки	Исходящая метка	Исходящая метка
	Выходной интерфейс	Выходной интерфейс
	Адрес следующего LSR	Адрес следующего LSR

Рис. 2.9. Запись в таблице пересылки LIB

Разные подзаписи внутри одной записи могут иметь либо одинаковые, либо разные значения исходящих меток. Более одной подзаписи бывает нужно для поддержки многоадресной рассылки пакета, когда пакет, который поступил к одному входящему интерфейсу, должен затем рассылаться через несколько исходящих интерфейсов. Обращение к таблице записей производится по значению входящей метки, т.е. по входящей метке *k* происходит обращение к *k-ой* записи в таблице.

Как уже отмечалось, в дополнение к информации, которая указывает, куда должен пересылаться пакет — следующий участок маршрута или следующий маршрутизатор, — запись в таблице может также содержать информацию, указывающую, какие ресурсы имеет возможность использовать пакет, например, определенную выходную очередь.

LSR может поддерживать либо одну общую таблицу, либо отдельные таблицы для каждого из своих интерфейсов. В первом варианте обработка пакета определяется исключительно меткой, переносимой в пакете. Во втором варианте обработка пакета определяется не только меткой, но и интерфейсом, к которому поступил пакет. LSR может использовать либо первый вариант, либо второй вариант, либо их сочетание.

Заметим, что в «традиционной» архитектуре систем коммутации пакетов разные режимы маршрутизации — одноадресная, многоадресная, с поддержкой QoS и др. — требуют различных алгоритмов пересылки. К примеру, при одноадресной пересылке пакетов используется алгоритм самого длинного совпадения (*longest match*) подряд идущих элементов сетевого адреса получателя. В MPLS же важнейшим свойством алгоритма пересылки является то, что всю информацию, необходимую для того, чтобы переслать пакет и чтобы принять решение о том, какие ресурсы может использовать пакет, LSR может получить всего однократным обращением к памяти. Это возможно благодаря тому, что:

- запись в таблице на рис. 2.9 содержит всю информацию, необходимую для пересылки пакета и для решения вопроса о ресурсах,
- метка, переносимая в пакете, непосредственно указывает индекс той записи в таблице, которая должна использоваться для пересылки пакета.

Возможность получать информацию (нужную как для пересылки, так и для резервирования ресурсов) путем однократного обращения к памяти и обеспечивает быструю пересылку пакетов в MPLS.

2.6. Привязка «метка-FEC»

На рис. 2.9 видно, что каждая запись в таблице пересылки, которую ведет LSR, содержит одну *входящую* метку и одну или более *исходящих* меток. В соответствии с этими двумя типами меток обеспечивается два типа привязки меток к FEC. Первый тип — метка для привязки выбирается и назначается в LSR локально. Такую привязку мы будем называть *локальной*. Второй тип — LSR получает от некоторого другого LSR информацию о привязке метки, которая соответствует привязке, созданной на этом другом LSR. Такую привязку мы будем называть *удаленной*.

Средства управления коммутацией по меткам используют для заполнения таблиц пересылки как локальную, так и удаленную привязку меток к FEC. Это может делаться в двух вариантах: *upstream* и *downstream*. Первый — это когда метки из локальной привязки используются как входящие метки, а метки из удаленной привязки используются как исходящие метки. Второй вариант — прямо противоположный, т.е. метки из локальной привязки используются как исходящие метки, а метки из удаленной привязки — как входящие метки. Рассмотрим каждый из этих вариантов.

Первый вариант называется *привязкой метки к FEC «снизу» (downstream label binding)*, потому что в этом случае привязка переносимой пакетом метки к тому FEC, которому принадлежит этот пакет, создается нижестоящим LSR, т.е. LSR, расположенным ближе к адресату пакета, чем LSR, который помещает метку в пакет. Отметим, что при привязке «снизу» пакеты, которые переносят определенную метку, передаются в направлении, противоположном направлению передачи информации о привязке этой метки к FEC.

Второй вариант называется *привязкой метки к FEC «сверху» (upstream label binding)*, потому что в этом случае привязка переносимой пакетом метки к тому FEC, которому принадлежит этот пакет, создается тем же LSR, который помещает метку в пакет; т.е. создатель привязки расположен «выше» (ближе к отправителю пакета), чем LSR, к которому пересылается этот пакет. Отметим, что при

привязке «сверху» пакеты, которые переносят определенную метку, передаются в том же направлении, что и информация о привязке этой метки к FEC.

LSR обслуживает также *пул «свободных» меток* (т.е. меток без привязки). При начальной установке LSR пул содержит все метки, которые может использовать LSR для их локальной привязки к FEC. Именно емкость этого пула и определяет, в конечном счете, сколько пар «метка-FEC» может одновременно поддерживать LSR. Когда маршрутизатор создает новую локальную привязку, он берет метку из пула; когда маршрутизатор уничтожает ранее созданную привязку, он возвращает метку, связанную с этой привязкой, обратно в пул.

Вспомним, что LSR может вести либо одну общую таблицу пересылки, либо несколько таблиц — по одной на каждый интерфейс. Когда маршрутизатор ведет общую таблицу пересылки, он имеет один пул меток. Когда LSR ведет несколько таблиц, он имеет отдельный пул меток для каждого интерфейса.

LSR *создает или уничтожает привязку метки к FEC* вследствие определенного события. Такое событие может инициироваться либо пакетами данных, которые должны пересылаться маршрутизатором LSR, либо управляющей (маршрутной) информацией (например, маршрутной информацией протокола OSPF, сообщениями JOIN/PRUNE протокола PIM, сообщениями Path/Resv протокола RSVP), которая должна обрабатываться LSR. Когда создание или уничтожение привязки инициируется пакетами данных, мы называем это привязкой *под воздействием данных (data-driven).* Когда создание или уничтожение привязки инициируется управляющей информацией, мы называем это привязкой *под воздействием управляющей информации (control-driven).* Привязка под воздействием данных предполагает, что LSR поддерживает как функции пересылки при коммутации по меткам, так и функции пересылки при традиционной маршрутизации. Поддержка функций пересылки при традиционной маршрутизации необходима потому, что привязка метки представляет собой эффект, сопутствующий традиционной маршрутизации пакета.

Каждый из этих двух механизмов имеет множество вариантов. Например, механизм привязки под воздействием данных может создавать привязку для потока приложения, как только он обнаружит первый пакет потока, или ждать до тех пор, пока не обнаружит несколько пакетов, предполагая при этом, что поток существует достаточно долго для того, чтобы успеть создать привязку.

Выбор того или иного метода создания привязки будет, безусловно, некоторым образом влиять на производительность и масштабируемость механизма, т.е. на то, насколько хорошо он будет

работать по мере роста сети. Можно также ожидать некоторое влияние метода привязки на живучесть механизма, т.е. насколько хорошо он будет работать в разных условиях эксплуатации.

Важной проблемой качества функционирования, возникающей при использовании схем привязки под воздействием данных (и, в меньшей степени, — схем привязки под воздействием управляющей информации), является *производительность*. Каждый раз, когда LSR решает, что поток должен коммутироваться по меткам, ему необходимо обмениваться информацией о привязке меток со смежными LSR, и ему может также потребоваться внести некоторые изменения в привязке меток к FEC. Все эти процедуры требуют передачи трафика, управляющего раздачей информации о привязке, и, следовательно, потребляют ресурсы средств управления коммутацией по меткам. Более того, эти процедуры потребляют тем больше ресурсов средств управления, чем больше доля потоков, выбранных для коммутации по меткам. Трудно количественно оценить, насколько дорогой является процедура назначения и распределения меток, но не подлежит сомнению, что производительность схем, работающих под воздействием данных, чувствительна к этому фактору. Если LSR не может назначать и распределять метки со скоростью, требуемой алгоритмом обнаружения потоков, то коммутироваться по меткам будет меньший процент потоков, и от этого будет страдать общая производительность.

В меньшей степени это относится к схемам, работающим под воздействием управляющей информации. Пока топология остается стабильной, весь трафик, который поступает в не пограничный маршрутизатор LSR, может коммутироваться по меткам без обработки пакетов управляющим процессором. Схемы, работающие под воздействием управляющей информации, могли получить информацию о привязке маршрутов от соседних LSR, которые не являлись следующими участками этих маршрутов. Когда топология изменится так, что эти «соседи» станут следующими участками маршрутов, коммутация по меткам будет продолжаться без прерывания. Но на производительность схем, работающих под воздействием данных, изменение топологии влияет. Если изменяется маршрут прохождения потока, новые LSR на этом маршруте воспринимают это так, как если бы был создан новый поток. Любой такой новый поток должен вначале пересылаться традиционно. Таким образом, изменение топологии может налагать очень тяжелое бремя на LSR, который только что стал новым следующим маршрутизатором для некоторого другого LSR. Во-первых, он внезапно получает большое число потоков, которые ранее шли по другому маршруту. Кроме того, не прекращается поступление новых потоков от вновь запущенных приложений. Все эти потоки должны пересылаться и анализироваться алгоритмами обнаружения потоков. Это, в свою очередь,

создает дополнительную нагрузку как на средства пересылки при традиционной маршрутизации, так и на средства управления коммутацией по меткам. Во время таких переходных процессов производительность средств пересылки LSR может приблизиться к производительности его средств пересылки при традиционной маршрутизации, т.е. стать примерно на порядок ниже, чем максимальная производительность LSR.

Проблема с производительностью для схем, работающих под воздействием управляющей информации, возникает в случаях, когда имеет место объединение маршрутов, — конфликт между масштабируемостью и производительностью. К этим вопросам мы еще вернемся в заключительных главах книги при обсуждении проблем трафик-инжиниринга TE.

2.7. Режимы операций с метками

Рассмотрим режимы трех базовых операций, которые представляют, по сути, основные принципы механизма коммутации по меткам: *назначение меток, распределение меток и сохранение меток.*

Назначение меток. Когда FEC создается путем анализа адресных префиксов, которые распределяются протоколом внутренней маршрутизации и используются для создания трактов LSP по участкам, возможны два *режима назначения меток*:

- независимое назначение (т.е. независимое создание трактов LSP);
- упорядоченное назначение (т.е. упорядоченное создание трактов LSP).

Независимое назначение. При независимом назначении меток каждый LSR сам, независимо от других событий, принимает решение о привязке метки к обнаруженному FEC и об уведомлении вышестоящего LSR об этой привязке. Такая ситуация аналогична традиционной маршрутизации, выполняемой в обычных IP-сетях, когда обнаруживаются новые маршруты.

Если LSR настроен на режим независимого назначения меток, то сообщение Label Mapping протокола LDP, который рассматривается в следующей главе, передается этим LSR при возникновении любой из следующих ситуаций:

- LSR распознает новый FEC с помощью таблицы пересылки и назначает метку снизу по собственной инициативе;
- LSR получает от вышестоящего маршрутизатора сообщение Label Request для FEC, присутствующего в его таблице пересылки;
- следующий маршрутизатор для FEC заменяется другим одноранговым узлом LDP-сеанса, и при этом активизирован механизм выявления закольцованных маршрутов;

- изменяются атрибуты привязки «метка-FEC»;
- информация о привязке метки получена от нижестоящего маршрутизатора в ситуации, когда не было создано никакой привязки сверху, или активизирован механизм обнаружения закольцованных маршрутов, или изменились атрибуты привязки «метка-FEC».

Упорядоченное назначение. Этот способ назначения меток имеет больше ограничений, чем предыдущий, в том смысле, что привязка метки к определенному FEC происходит только тогда, когда LSR либо выступает в роли выходного узла для этого FEC, либо уже получил информацию о привязке «метка-FEC» от нижестоящего маршрутизатора.

Если LSR использует режим упорядоченного назначения меток, то сообщение Label Mapping передается нижестоящими LSR при возникновении любой из следующих ситуаций:

- LSR распознает новый FEC с помощью таблицы пересылки и является для этого FEC выходным маршрутизатором;
- LSR получает от вышестоящего маршрутизатора сообщение Label Request для FEC, присутствующего в его таблице пересылки, и либо является выходным маршрутизатором для этого FEC, либо имеет привязку к нему метки, назначенную снизу;
- следующий маршрутизатор для FEC заменяется другим одноранговым узлом LDP-сеанса, и при этом активизирован механизм выявления закольцованных маршрутов;
- изменяются атрибуты привязки «метка-FEC»;
- информация о привязке метки получена от нижестоящего маршрутизатора в ситуации, когда не было создано никакой привязки сверху, или активизирован механизм обнаружения закольцованных маршрутов, или изменились атрибуты привязки «метка-FEC».

Распределение меток. В технологии MPLS могут использоваться два режима распределения меток: нижестоящим LSR по запросу вышестоящего или нижестоящим LSR по собственной инициативе. *Режим распределения нижестоящим по запросу вышестоящего* используется для создания трактов LSP по участкам. Он позволяет вышестоящему LSR в явном виде запрашивать привязку метки к определенному FEC у соседнего с ним нижестоящего LSR. *Режим распределения меток нижестоящим по собственной инициативе* используется тогда, когда нижестоящему LSR нужно «раздать» метки вышестоящим LSR, хотя те не запрашивали у него этого в явном виде.

Оба режима распределения меток обсуждались в предыдущем параграфе. Добавим лишь, что архитектура MPLS позволяет применять в одной системе один или оба режима в зависимости

от деталей реализации, таких как характеристики интерфейсов и имеющиеся ресурсы сети MPLS. Однако в каждой совокупности смежных узлов, которые участвуют в распределении меток, вышестоящий LSR и нижестоящий LSR должны согласовать между собой и использовать для создания трактов LSP один и тот же режим распределения.

Это связано с тем, что архитектура MPLS не предполагает применение какого-то единственного протокола распределения меток. В одной и той же сети MPLS могут использоваться: специальный протокол распределения меток Label Distribution Protocol (LDP), рассматриваемый в главе 3, протокол сигнализации RSVP, рассматриваемый в главе 4, а также расширения возможностей протоколов маршрутизации, например, рассматриваемого в главе 7 протокола междоменовой маршрутизации Border Gateway Protocol (BGP).

Распределение меток с помощью протокола RSVP. С начала 1990-х годов до конца XX столетия протокол RSVP (Resource Reservation Protocol) рассматривался только как средство обеспечения QoS в IP-сетях путем резервирования ресурсов для обслуживания того или иного потока данных. При объединении RSVP с MPLS возникло новое качество. Входной LSR тракта LSP может использовать разные способы для того, чтобы определить, какая именно метка назначена для блоков данных (PDU) того или иного протокола. Когда для совокупности PDU назначена метка, эта метка определяет *поток*, проходящий по LSP. Таким образом, внутри LSP создается *LSP-туннель*, называемый так потому, что его трафик «невидим» для промежуточных узлов, через которые он (туннель) проходит. Запросы меток, связанных с потоками RSVP, переносятся в сообщениях Path протокола RSVP, а информация о назначенных метках — в сообщениях Resv этого протокола. Использование RSVP для оптимизации трафика в сети MPLS заметно отличается от того, каким оно виделось разработчикам RSVP в середине 1990-х годов. Протокол RSVP может играть в сети MPLS двойную роль: распределять метки и поддерживать качество обслуживания. Обо всем этом будет сказано не только в главе 4, целиком отданной протоколу RSVP, но и в главах, посвященных трафик-инжинирингу.

Распределение меток с помощью протокола BGP. Когда два смежных LSR, поддерживающих протокол BGP, обмениваются информацией о маршрутах, они должны обмениваться также и информацией о метках, назначенных для этих маршрутов. Обмен обеспечивается путем вложения информации о привязке меток к FEC в то же сообщение Update протокола BGP, которое используется для обмена информацией о маршруте.

Протоколы, управляющие маршрутизацией, различаются функциями, применяемыми алгоритмами и областью использования.

Эти протоколы «понимают» топологию сети, в которой они функционируют, и могут поддерживать либо *внутреннюю (IGP — Interior Gateway Protocol)*, либо *внешнюю (EGP — Exterior Gateway Protocol) маршрутизацию*. Протоколы группы IGP рассылают маршрутную информацию внутри какой-либо автономной системы, а протоколы группы EGP служат для маршрутизации между такими системами (под *автономной системой* понимается совокупность маршрутизаторов, управляемых единой административной системой, т.е. маршрутизаторы одного административного домена).

В книге рассмотрены два протокола внутренней маршрутизации MPLS.

Протоколу OSPF посвящена глава 5 книги. Это — один из самых распространенных протоколов внутренней маршрутизации, используемых сегодня в Интернет. Он был разработан с целью преодолеть ограничения протокола маршрутизации RIP, препятствующие использованию его в крупных сетях, — ограничение по числу пересылок (максимум 15), отсутствие поддержки масок подсети переменной длины при бесклассовой междоменовой маршрутизации CIDR (хотя такая возможность была добавлена во вторую версию протокола — RIP-2), неэффективное, как правило, использование пропускной способности звеньев. Сам протокол RIP в книге подробно не рассматривается. OSPF предусматривает передачу *объявлений о состоянии каналов LSA (Link State Advertisements)* ко всем маршрутизаторам в своей области, поддерживающим этот протокол. LSA содержат информацию об интерфейсах маршрутизатора. Когда маршрутизатор, использующий протокол OSPF, собирает информацию о топологии сети, он выполняет алгоритм Дийкстры, называемый также алгоритмом SPF (Shortest Path First), для расчета маршрутов к каждому другому узлу, сведения о котором этот маршрутизатор имеет в своей базе данных.

Протокол IS-IS представляет собой протокол внутридоменовой маршрутизации с рассылкой объявлений о состоянии каналов. Он специфицирован в ISO/IEC 10589, а его использование в сети Интернет для маршрутизации пакетов IP описано в RFC 1142. Этот протокол схож с протоколом OSPF в плане функциональных возможностей, но разнится с ним в деталях работы, что обсуждается в главе 6.

Режимы сохранения меток. Еще одной характеристикой использования меток является режим их сохранения. Когда вышестоящий LSR получает метку от нижестоящего LSR, который в данный момент не является для него смежным с точки зрения данного FEC, он должен принять относительно этой метки решение: либо использовать ее (т.е. «сохранить» у себя ее привязку к FEC), либо отбросить.

В MPLS используются два основных режима сохранения меток:

- либеральный режим,
- консервативный режим.

При *либеральном режиме* вышестоящий LSR сохраняет у себя любую привязку к FEC меток, которые он получил от не смежных нижестоящих LSR (т.е. метки пришли к нему транзитом). При *консервативном режиме* вышестоящий LSR отказывается от таких меток, т.е. отбрасывает их. Преимуществом либерального режима является то, что если нижестоящий LSR станет с точки зрения данного FEC смежным маршрутизатором, изменять привязку «метка-FEC» не понадобится. Это позволяет намного быстрее реагировать на изменения маршрутов. Конечно, если привязка метки к FEC уже была отброшена, то ее придется назначить повторно до того, как можно будет использовать LSP. При консервативном режиме требуется меньше ресурсов, но реакция на изменения в следующих участках маршрута происходит намного медленнее. Обо всем этом несколько подробнее будет сказано в следующих главах.

2.8. Фиксированные значения метки

В заключение этой главы приведем несколько фиксированных (резервированных) значений меток:

- **0** — «IPv4 Explicit NULL Label». Эта метка должна находиться на дне стека меток. Она указывает, что стек должен быть извлечен из пакета, и дальнейшая маршрутизация этого пакета должна основываться на заголовке IPv4.
- **1** — «Router Alert Label». Эта метка может находиться в любом месте стека меток за исключением его дна. Когда приходит пакет, содержащий такую метку наверху стека, он доставляется местному программному модулю для обработки. Дальнейшая маршрутизация будет производиться либо по нижележащей метке, либо на основе информации, содержащейся в заголовке протокола сетевого уровня или инкапсулированных в него протоколов. При пересылке пакета метка Router Alert Label должна быть снова помещена наверх стека меток. Использование этой метки регламентировано в RFC 2113 и аналогично использованию опции Router Alert в IP-пакетах.
- **2** — «IPv6 Explicit NULL Label». Эта метка должна находиться на дне стека меток. Она указывает, что стек должен быть извлечен из пакета, и дальнейшая маршрутизация пакета должна основываться на заголовке IPv6.

- **3** — «Implicit NULL Label». Эта метку LSR может присваивать и распространять, но пакеты ей никогда не помечаются. Когда LSR, согласно LIB, должен заменить (swap) метку, находящуюся наверху стека, а новая метка является Implicit NULL Label, то вместо замены, LSR удаляет (*pop*) стек меток.
- **4 — 15** — Значения зарезервированы для дальнейшего использования.

Резервирование значений необходимо не только для служебных сообщений. В связи с активным развитием технологии MPLS оно представляется весьма важным и с других точек зрения.

Глава 3
Протокол LDP

3.1. Классы эквивалентности пересылки и LDP

Понятие *класс эквивалентности пересылки FEC* уже обсуждалось в двух предыдущих главах. Там же говорилось о том, что для переноса через сеть MPLS пакетов, принадлежащих разным FEC, в сети создаются *виртуальные тракты LSP*, и было показано, как с помощью метки MPLS устанавливается соответствие «пакет-FEC», определяющее, по какому LSP должен быть направлен пакет с этой меткой. В этой главе речь пойдет о том, каким образом производится распределение меток по всем LSR сети MPLS с использованием протокола LDP (Label Distribution Protocol).

В спецификации LDP к настоящему моменту установлены два типа элементов, с помощью которых может определяться FEC:

- Address Prefix — адресный префикс любой длины от нуля до полного адреса,
- Host Address — полный адрес хоста.

Решения о назначении меток могут основываться на критериях пересылки, таких как одноадресная маршрутизация к получателю, оптимизация распределения трафика в сети, многоадресная рассылка, виртуальная частная сеть VPN, механизмы обеспечения качества обслуживания QoS и др. Спецификация же протокола LDP определяет правила, по которым устанавливается соответствие между входным пакетом и его LSP.

Для распределения меток могут использоваться разные методы:

- метод на основе топологии (topology-based method) использует стандартную обработку протоколов маршрутизации (например, OSPF и BGP, рассматриваемых в главах 5 и 7, соответственно);
- метод на основе запросов (request-based method) использует обработку управляющего протокола на основе запросов (например, протокола RSVP, см. главу 4);
- метод на основе трафика (traffic-based method) запускает процедуру присвоения и распределения меток при получении пакета (также обсуждается в главе 4).

Методы на основе топологии и запросов являются примерами привязки меток к FEC, управляемой от программы, а метод на основе трафика является примером привязки, управляемой данными. Во всех этих случаях архитектурой MPLS предусматривается, что назначение метки, то есть ее привязку к определенному FEC, производит LSR, который является выходным пограничным маршрутизатором для пакетов этого FEC. Перерисуем рис. 1.3 из главы 1, но вместо LSP покажем на нем механизм распределения меток (рис. 3.1). Тогда LSR, о котором идет речь, является LSR5.

По-английски такой LSR называется *downstream LSR*, то есть расположенный «ниже по течению»; мы будем называть его *нижним* или *нижестоящим LSR*, а расположенный «выше по течению» *upstream LSR* будем называть *верхним* или *вышестоящим LSR*. Таким образом, назначение меток всегда производится *снизу,* то есть в сторону, противоположную направлению трафика. Нижний LSR информирует соседние верхние LSR о том, какие метки он привязал к каждому FEC поступающих к нему пакетов. Этот процесс и называется *распределением меток*, а обеспечивает его *протокол распределения меток LDP* (*Label Distribution Protocol*).

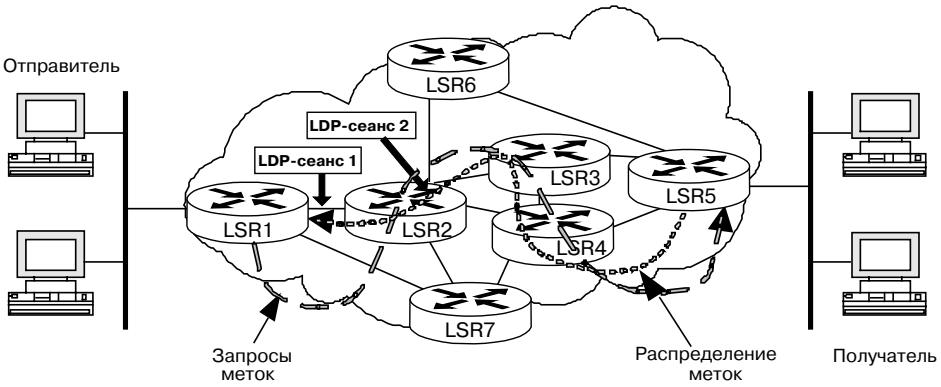


Рис. 3.1. Фрагмент MPLS-сети

Более подробно протокол LDP будет рассмотрен в следующих параграфах этой главы, но прежде еще раз подчеркнем, что архитектура MPLS не требует обязательного применения LDP. Для распределения меток могут применяться модификации существующих протоколов маршрутизации, позволяющие использовать их для передачи информации о метках, например, рассматриваемый в главе 7 протокол BGP. Протокол RSVP, который рассматривается в следующей главе, также имеет расширения, обеспечивающие поддержку обмена метками с уведомлением. Но все же, протокол распределения меток LDP был признан комитетом IETF наиболее удачным и, что еще важнее, хорошо специфицирован им. Кроме того, определено расширение базового протокола LDP для поддержки явной маршрутизации

с учетом обеспечения качества обслуживания QoS и управления трафиком TE — *протокол LDP с учетом ограничивающих условий CR-LDP (Constraint-Based LDP)*. Ко всему прочему, LDP устанавливает надежные транспортные соединения со смежными маршрутизаторами LSR по протоколу TCP, причем в случае, если между двумя LSR надо одновременно установить несколько LDP-сеансов, используется единственное TCP соединение.

Имеются следующие схемы обмена метками:

- LDP преобразует в метки IP-адреса получателя при одноадресной передаче,
- RSVP и CR-LDP используются для оптимизации распределения трафика в сети и для резервирования ресурсов,
- BGP работает с внешними метками VPN.

Все эти схемы рассматриваются в книге далее, а в этой главе внимание сосредоточено на протоколе LDP.

Основным документом, задающим спецификацию протокола LDP, является RFC 3036 «LDP Specification». Этот документ был разработан Л. Андерсоном, П. Дуланом, Н. Фельдманом, А. Фредетте и Б. Томасом в январе 2001 года. В дополнение к нему, документ RFC 3037 «LDP Applicability», написанный Б. Томасом и Э. Греем, определяет применимость протокола LDP именно в контексте MPLS. RFC 3037 описывает процедуры протокола LDP, используемые маршрутизаторами при распределения меток MPLS. В документе рассматриваются также аспекты расширяемости протокола и аспекты безопасности, упоминаемые в параграфе 3.5 этой главы. Документ RFC 3035 «MPLS Using LDP and ATM VC Switching» описывает способы использования ATM-коммутаторов в качестве маршрутизаторов MPLS (эти маршрутизаторы обозначаются ATM-LSR). На момент написания этой книги в рабочей группе по MPLS также обсуждались и дорабатывались дополнительные документы по протоколу LDP: «LDP State Machine» (Конечный автомат обработки сообщений протокола LDP), «Definitions of Managed Objects for the Multiprotocol Label Switching LDP SNMP» (Определения контролируемых объектов LDP для протокола SNMP), «Fault Tolerance for LDP and CR-LDP» (Отказоустойчивость LDP), «LDP Extensions for Optical User Network Interface (O-UNS) Signaling» (Расширения протокола LDP для поддержки сигнализации в оптическом интерфейсе «пользователь-сеть» O-UNI). Все они, в той или иной степени, упоминаются в этой главе.

3.2. Основы протокола LDP

Протокол LDP создан в рабочей группе по MPLS комитета IETF с целью специфицировать процедуры распределения меток MPLS. В связи с тем, что протокол LDP тесно взаимодействует с протоко-

лами внутренней маршрутизации IGP, его часто называют *протоколом пересылки по участкам*. Протокол распределения меток LDP представляет собой *набор процедур и сообщений, при помощи которых LSR организуют тракты коммутации по меткам, обмениваясь информацией о привязке меток к FEC с «соседними» LSR, а также поддерживают и прекращают LSP-сеансы.* Речь идет именно об *обмене* информацией, как это показано на рис. 3.1, поскольку протокол предусматривает двусторонний диалог взаимодействующих LSR, являющихся в данном контексте *одноранговыми узлами LDP*. Этот диалог называется *LDP-сеансом*, в ходе которого каждый из взаимодействующих LSR получает сведения о привязке меток к FEC в другом LSR. В протоколе определен также механизм передачи уведомлений и обнаружения в LSP закольцованных маршрутов, уже обсуждавшийся ранее.

Итак, процедуры протокола LDP позволяют LSR, выполняющему этот протокол, создавать тракты LSP. Конечной точкой такого тракта является либо смежный LSR, имеющий прямую связь с данным LSR, либо выходной LSR, доступный этому LSR через некоторое количество транзитных LSR. Процессы обнаружения конечных точек этих двух типов называются, соответственно, процессом *базового обнаружения* и процессом *расширенного обнаружения*. LDP создает двустороннюю связь двух смежных LSR, которые становятся одноранговыми узлами LDP и взаимодействуют друг с другом посредством LDP-сеанса. На рис. 3.1 LSR1 и LSR2 являются одноранговыми узлами LDP, взаимодействующими друг с другом посредством LDP-сеанса 1, а LSR2 и LSR3 — одноранговыми узлами LDP, взаимодействующими посредством LDP-сеанса 2.

При обмене между LSR информацией, связанной с привязкой «метка-FEC», используются четыре категории сообщений LDP:

- сообщения обнаружения (discovery messages), используемые для того, чтобы объявить и поддерживать присутствие LSR в сети;
- сеансовые сообщения (session messages), предназначенные для создания, поддержки и прекращения LDP-сеансов между LSR;
- сообщения-объявления (advertisement messages), используемые для создания, изменения и отмены привязки метки к FEC;
- уведомляющие сообщения (notification messages), содержащие вспомогательную информацию и информацию об ошибках.

Хотя «раздает» метки всегда нижний LSR, инициатором их распределения не обязательно должен быть он; процесс может инициировать и верхний LSR, направив к нижнему LSR соответствующий запрос; такой режим называется *downstream on-demand*. В той или иной сети может использоваться распределение меток либо только по запросам сверху, либо только по инициативе нижнего LSR *(unsolicited downstream)*, либо и то, и другое вместе.

Распределение меток может быть *независимым* или *упорядоченным*. В первом случае LSR может уведомить вышестоящий LSR о привязке метки к FEC до того, как получит информацию о привязке от нижестоящего LSR. Во втором случае высылать подобное уведомление «наверх» разрешается только после получения таких сведений «снизу».

Заметим, что нижний LSR распределяет метки не только по тем верхним LSR, которые имеют с ним прямые связи. Протокол распределения меток может быть использован и для диалога двух LSR, между которыми существует лишь коммутируемая связь, однако результат распределения в этом случае зависит от того, в каком из двух режимов, либеральном или консервативном, работает верхний LSR.

Консервативный режим распределения меток — в этом режиме сообщения-объявления о привязке «метка-FEC», получаемые от не смежных LSR, не принимаются и отбрасываются. LSR привязывает метку к FEC только в том случае, если он является выходным маршрутизатором или если он получил сообщение о привязке от смежного с ним LSR. Такой режим позволяет LSR обслуживать меньшее число меток.

Либеральный режим распределения меток — в этом режиме привязка метки, выданная тем нижним LSR, с которым нет прямой связи, запоминается и используется. Такой режим удобен тем, что при реконфигурации сети соответствие между меткой и FEC сохраняется, даже если связь с LSR, определившим это соответствие, стала не коммутируемой, а прямой. Недостаток либерального режима состоит в том, что в верхнем LSR приходится хранить и обрабатывать заметно больше информации о соответствии между метками и FEC.

Как уже говорилось, метка всегда локальна, то есть обозначает некоторый FEC только для пары маршрутизаторов, между которыми имеется прямая или коммутируемая связь, и используется при пересылке пакетов этого FEC от того из маршрутизаторов данной пары, который является в ней верхним (*upstream LSR*), к тому, который является нижним (*downstream LSR*). Для пересылки пакетов того же FEC к следующему маршрутизатору используется другая метка, идентифицирующая этот FEC для новой пары маршрутизаторов, в которой маршрутизатор, бывший в предыдущей паре нижним, приобретает статус верхнего, а статус нижнего получает второй маршрутизатор этой новой пары. Отсюда ясно, что каждый маршрутизатор MPLS-сети должен хранить соответствие между входящими и исходящими метками для всех FEC, которыми он оперирует.

Одной из важнейших функций протокола LDP является обнаружение петель. Для этой цели можно использовать два поля в сообщениях *Label Request* и *Label mapping* (рассматриваемых в параграфе 3.4), а именно *Path Vector* и *Hop Count*.

Поле *Path Vector* содержит список *LSR-идентификаторов* (первые 4 октета LDP-идентификатора), принадлежащих тем LSR, через которые прошло содержащее его сообщение. Если LSR получает сообщение и обнаруживает в поле *Path Vector* свой собственный LSR-идентификатор, он «понимает», что возникла петля. В протоколе LDP предусматривается возможность задать максимально допустимое значение поля *Path Length,* по достижении которого тоже принимается решение о возникновении петли.

Второй вариант - поле *Hop Count,* которое содержит счетчик LSR, пройденных сообщением LSR. Каждый пройденный LSR увеличивает его значение на единицу. Маршрутизатор, обнаруживший, что счетчик достиг максимально допустимой величины, обрабатывает сообщение, как сделавшее петлю.

3.3. Формат и параметры сообщений LDP

3.3.1. Блоки данных протокола LDP

Сообщения LDP передаются в *блоках протокольных данных PDU (Protocol Data Unit)* по TCP-соединению во время LDP-сеанса. Каждый PDU может переносить одно или несколько сообщений LDP, причем сообщения в одном PDU не обязательно должны быть связаны друг с другом. Блок данных протокола LDP, который обозначается LDP PDU, показан на рис. 3.2.

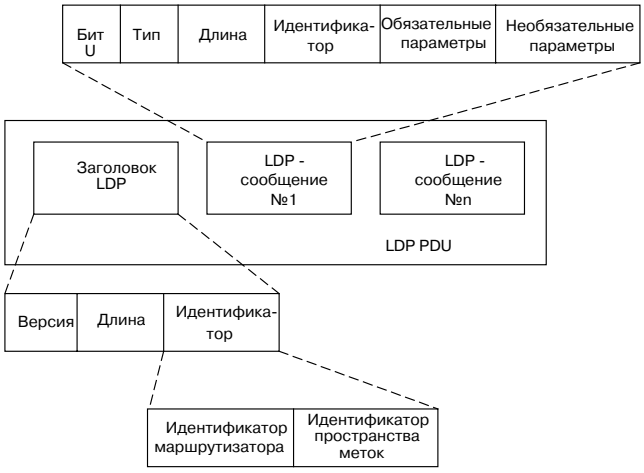


Рис. 3.2. Блок данных протокола LDP

Каждый LDP PDU содержит *LDP-заголовок*, за которым следуют *LDP-сообщения*. Заголовок состоит из трех полей: *версия, длина* и *идентификатор*.

Поле *Version* занимает два октета и содержит целое число — номер версии протокола LDP.

Поле *PDU Length* содержит число, которое обозначает общую длину PDU в байтах (включая LDP-заголовок). Максимально допустимая длина PDU может быть задана при инициировании LDP-сеанса. По умолчанию она равна 4096 байтам.

Поле *LDP Identifier* занимает шесть октетов и однозначно определяет пространство меток маршрутизатора, передающего данный PDU. Первые четыре октета содержат идентификатор LSR, представляющий собой IP-адрес этого LSR. Следующие два октета идентифицируют пространство меток внутри LSR, если используется интерфейсное пространство меток. Если LSR использует платформное пространство меток, это поле содержит два нуля.

Общий размер LDP-заголовка составляет десять октетов.

Сообщение протокола LDP содержит шесть полей: *Бит U, Тип, Длина, Идентификатор, Обязательные параметры* и *Необязательные параметры.*

3.3.2. Схема Type-Length-Value

Большая часть информации, переносимой в LDP-сообщениях, кодируется по схеме *«тип-длина-значение»* (*TLV*), представленной на рис. 3.3. Как станет ясно из дальнейшего материала книги, конструкция TLV является эффективным и расширяемым способом кодирования параметров сообщений не только в LDP.

Суть конструкции *TLV* в том, что каждый параметр представляется в ней тремя элементами: полем типа, полем длины и полем значения. Первый элемент содержит код типа параметра, в поле длины указывается размер поля значения в октетах, а поле значения содержит значение параметра. Конструкция является расширяемой, поскольку в поле значения могут вкладываться другие конструкции «тип-длина-значение», а также имеются резервные поля типа, которые могут использоваться производителем оборудования для своих целей.

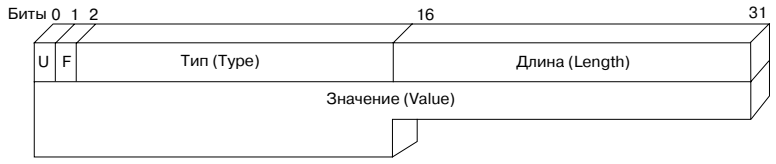


Рис. 3.3. Формат кодирования параметра по схеме TLV

Биты *U (Unknown)* и *F (Forward)* определяют поведение LSR в случае, когда он не распознает поле *Type.* Если бит *U*=0, то сообщение, содержащее нераспознанный параметр, должно игнорироваться, а отправителю должно быть передано уведомление. Если бит *U*=1, то нераспознанная *TLV* игнорируется, и производится обработка оставшейся части сообщения. Бит *F* имеет смысл только в случае, если *U=1.* Если *F=0*, то не распознанная *TLV* исключается из сообщения, если же *F=1*, сообщение пересылается дальше целиком, вместе с не распознанной *TLV*.

Поле *Тип (Type)* длиной 15 битов указывает тип параметра (или сообщения), определяя трактовку поля *Value*.

Поле *Значение* (*Value*) содержит информацию, интерпретируемую LSR согласно значению поля *Type*. Здесь отметим лишь, что основную содержательную часть *Value* составляют *обязательные* и *необязательные параметры TLV.*

Эти параметры могут быть ориентированы только на сообщения какого-то одного типа или входить в сообщения разных типов. Авторам показалось удобнее описать в следующем пункте этого параграфа параметры универсального применения, т.е. встречающиеся более чем в одном сообщении, а параметры индивидуального применения рассмотреть вместе с соответствующими сообщениями в следующем параграфе. Надеемся, что читателю это тоже покажется удобным.

3.3.3. Параметры TLV

Здесь рассматриваются параметры, которые используются в сообщениях LDP более чем одного типа. Начнем рассмотрение с параметра *FEC TLV*, формат которого представлен на рис. 3.4.

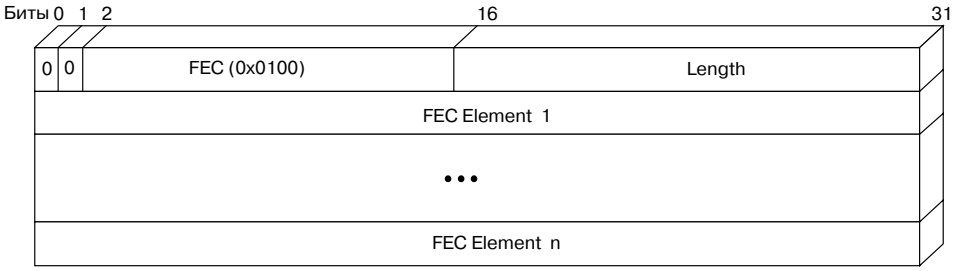


Рис. 3.4. Формат параметра FEC TLV

Класс FEC кодируется при помощи FEC-элементов. Поле *FEC element* содержат один из FEC-элементов, кодирование которого зависит от его типа, определяемого первым байтом поля. За первым байтом следует поле переменной длины, содержащее информацию, интерпретируемую согласно типу элемента:

Тип FEC-элемента	Значение 1-го байта
Wildcard	00000001
Prefix	00000010
Host Address	00000011

Если LSR обнаруживает FEC-элемент, тип которого он не может декодировать, обработка параметра FEC TLV и содержащего его сообщения приостанавливается, о чем передается уведомляющее сообщение *Unknown FEC* отправителю пакета.

Тип *Wildcard* используется только в сообщениях *Label Withdraw (отмена привязки метки)* и *Label Release (освобождение метки)* и означает, что эти отмена или освобождение должны быть выполнены для всех FEC, связанных с меткой, которая содержится в следующем *Label TLV*.

Тип FEC-элемента *Prefix* определяет представленный на рис. 3.5 формат.

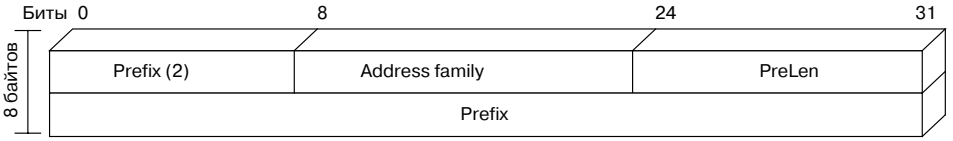


Рис. 3.5. Формат FEC-элемента Prefix

Поле *Address Family* — двухбайтовое поле, содержащее номер семейства адресов определенного *IANA (Internet Assigned Number Authority)*. Если при обработке *FEC TLV* маршрутизатор LSR обнаруживает FEC-элемент со значением *Address Family*, которое он не поддерживает, этот LSR останавливает обработку сообщения и передает отправителю уведомляющее сообщение *Unsupported Address Family*.

Поле *PreLen* — октет, содержащий длину (в битах) адресного префикса, находящегося в следующем поле. Нулевое значение в поле *PreLen* говорит о том, что префиксом является полный адрес. В этом случае значение поля *Prefix* будет нулевым.

И, наконец, сам *Prefix* — адресный префикс, который кодируется в соответствии с полем *Address Field* и длина которого определена в поле *PreLen.*

Формат FEC-элемента типа *Host Address* аналогичен приведенному выше формату FEC-элемента Prefix, за исключением того, что с пятого байта начинается поле *Host Addr*, содержащее адрес хоста (рис. 3.6).

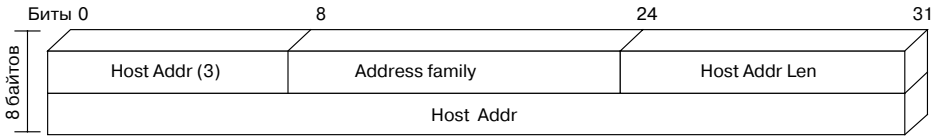


Рис. 3.6. Формат FEC-элемента Host Address

Теперь рассмотрим параметр ***Label TLV.*** Этим параметром кодируются метки, и потому *Label TLV* содержится в сообщениях, служащих для объявления, запроса, освобождения и отмены привязки метки. Существует несколько типов *Label TLV*.

Тип *Generic Label TLV* используется для кодирования меток при соединениях, где значения меток независимы от технологии уровня звена данных, например PPP или Ethernet (рис. 3.7). Здесь к уже рассмотренным выше полям добавляется четырехбайтовое поле *Label,* содержащее 20 битовую метку.

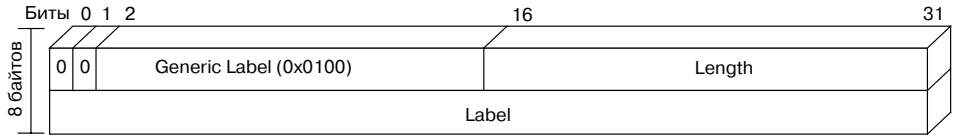


Рис. 3.7. Формат Generic Label TLV

Тип *ATM Label TLV,* как следует из его названия, применяется для соединений ATM (рис. 3.8).

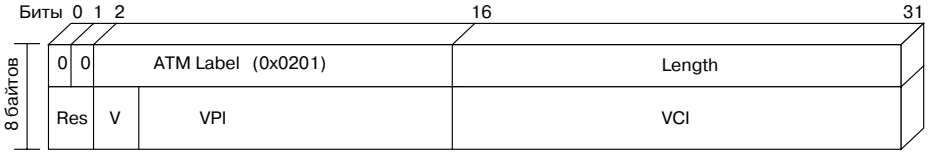


Рис. 3.8. Формат ATM Label TLV

Поле *Res* резервировано, получает значение ноль на передаче и игнорируется на приеме.

Поле *V* — двухбитовый индикатор, отмечающий значащие поля (незначащие поля должны игнорироваться получателем). Предусмотрены следующие значения *V:*

- 00 — оба поля VPI и VCI являются значащими,
- 01 — значащим является только VPI,
- 10 — значащим является только VCI.

VPI (Virtual Path Identifier) — 12-разрядный идентификатор виртуального тракта. Если значение VPI содержит меньше 12 битов, остальные биты имеют значение ноль.

VCI (Virtual Channel Identifier) — 2-байтовый идентификатор виртуального канала. Точно так же, если VCI содержит меньше 16 битов, остальные биты имеют значение ноль.

Тип *Frame Relay Label TLV* используется для соединений FR (рис. 3.9).

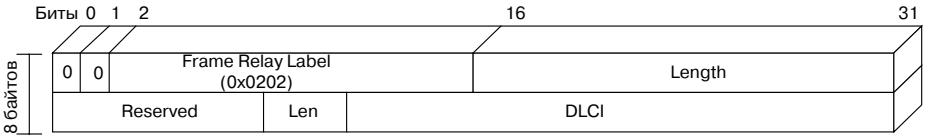


Рис. 3.9. Формат FR Label TLV

Поле *Res* резервировано, оно получает значение ноль на передаче и игнорируется на приеме.

Поле *Len* определяет количество битов *DLCI:*

- 00 — 10-битовый DLCI,
- 01 — резервировано,
- 10 — 23-битовый DLCI,
- 11 — резервировано.

DLCI (Data Link Connection Identifier) — идентификатор соединения в звене данных.

Параметр *Address List TLV* представлен на рис. 3.10. Его значения присутствуют в сообщениях *Address* и *Address Withdraw.*

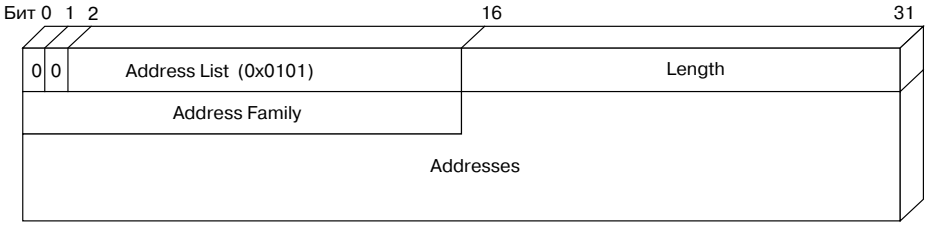


Рис. 3.10. Формат параметра Address List TLV

Address Family — двухбайтовое поле, содержащее номер семейства адресов, определенного IANA.

Addresses — список адресов, кодируемых в соответствии с ранее указанным семейством адресов. В базовой версии LDP для семейств адресов предусмотрен следующий формат:

- IP версия 4 — полный 4-байтовый адрес,
- IP версия 6 — полный 16-байтовый адрес.

Параметр ***Hop Count TLV*** представлен на рис. 3.11. Этот TLV появляется в качестве необязательного параметра в сообщениях, организующих LSP. В нем подсчитывается число LSR вдоль LSP в процессе его создания.

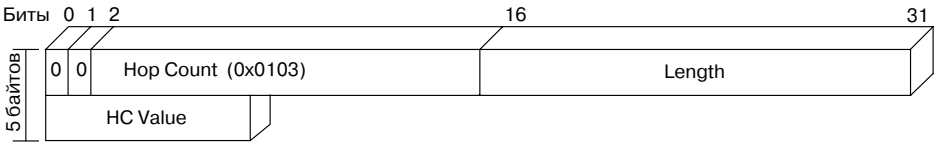


Рис. 3.11. Формат параметра Hop Count TLV

Поле *HC Value* — однобайтовое поле, содержащее результат работы счетчика LSR вдоль LSP.

Параметр ***Path Vector TLV*** представлен на рис. 3.12. Он используется вместе с *Hop Count TLV* в сообщениях *Label Request* и *Label Mapping* для предотвращения петель, о чем говорилось выше. Как раз *LSR Id x* и содержит список идентификаторов тех LSR, через которые прошел пакет.

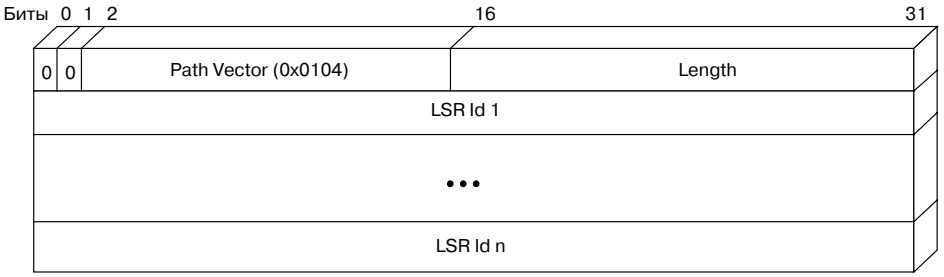


Рис. 3.12. Формат параметра Path Vector TLV

Параметр ***Status TLV*** представлен на рис. 3.13. Этот TLV несут уведомляющие сообщения, чтобы специфицировать событие, о котором сигнализирует уведомление.

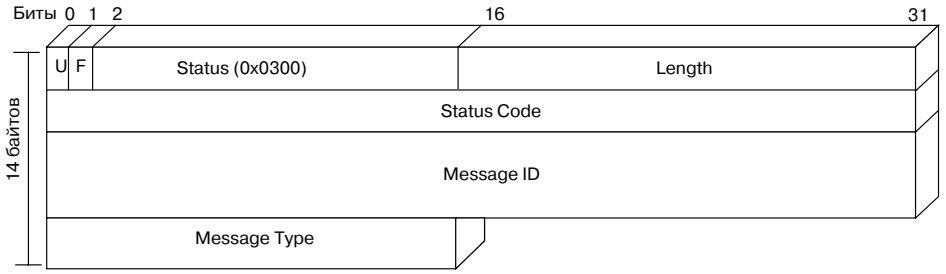


Рис. 3.13. Формат параметра Status TLV

Бит *U* имеет значение 0, если Status TLV передается в уведомляющем сообщении, и 1 во всех других сообщениях.

Бит *F* должен иметь то же значение, что и *F*-бит в поле *Status Code*.

Status Code — поле, кодирующее произошедшее событие. Его структура показана на рис. 3.14.

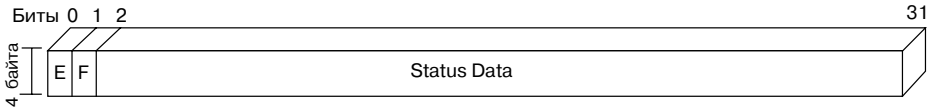


Рис. 3.14. Структура поля Status Code

Бит *E* — бит критической ошибки (*Fatal Error*). Имеет значение 1 в случае уведомления о критической ошибке.

Бит *F* — бит пересылки (*Forward*). Когда он имеет значение 1, следующему или предыдущему маршрутизатору LSR должно быть передано уведомление, если произошло событие, связанное с указанным в *Status Code*.

Поле *Status Data* — 30-битовое информационное поле, нулевое значение которого соответствует успеху.

Поле *Message ID* — 32 битовое поле, идентифицирующее сообщение, к которому относится *Status TLV*. Если это поле имеет нулевое значение, *Status TLV* не относится к какому-либо определенному сообщению.

Message Type — поле, обозначающее тип сообщения, к которому относится *Status TLV.* Если это поле имеет нулевое значение, *Status TLV* не относится к сообщению какого-либо определенного типа.

3.3.4. Формат сообщений LDP

Теперь перерисуем верхнюю часть рис. 3.2 несколько более подробно. В этом виде формат сообщения LDP изображен на рис. 3.15.

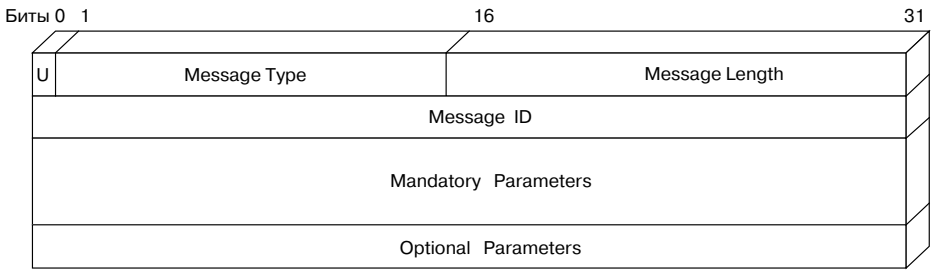


Рис. 3.15. Формат сообщения LDP

Бит *U* — бит неизвестного сообщения. Если он имеет значение 0, то отправителю передается уведомление о том, что тип сообщения не опознан, если же бит *U* имеет значение 1, то сообщение просто игнорируется.

Поле *Message Type* — идентифицирует тип сообщения. В таб. 3.1 представлены идентификаторы основных сообщений протокола LDP и их назначение. Краткому описанию приведенных в табл. 3.1 сообщений посвящен следующий параграф.

Таблица 3.1. Сводный перечень сообщений протокола LDP

Сообщения протокола LDP	Message Type	ОПИСАНИЕ
Notification (Уведомление)	0001	Информирует участвующие в сеансе маршрутизаторы о сбое в работе или отказе при обработке сообщений протокола LDP
Hello (Приветствие)	0100	Используется как часть механизма обнаружения узлов и периодического обмена приветствиями во время LDP-сеанса
Initialization (Инициализация)	0200	Инициирует организацию LDP-сеанса и обеспечивает согласование общих параметров этого сеанса
KeepAlive (Поддержание)	0201	Используется для поддержания LDP-сеанса в активном (рабочем) состоянии
Address (Адрес)	0300	Сообщает транспортные адреса интерфейсов маршрутизатора
Address Withdraw (Отмена адреса)	0301	Информирует маршрутизаторы о том, что содержащиеся в сообщении транспортные адреса более не доступны
Label Mapping (Назначение метки)	0400	В ответ на запрос передает значение метки запрашивающему ее LSR
Label Request (Запрос метки)	0401	Запрашивает метку у нижестоящего LSR
Label Withdraw (Отмена привязки метки)	0402	Используется для отмены ранее объявленной привязки
Label Release (Освобождение метки)	0403	Информирует LSR о том, что привязка «FEC-метка» для данной метки больше не нужна
Label Abort Request (Отмена запроса метки)	0404	Используется для прерывания (отмены) текущего сообщения запроса метки
Vendor-private (резерв производителя)	3E00-3EFF	Сообщения, специфицируемые производителем оборудования
Experimental (экспериментальные)	3F00-3FFF	Экспериментальные сообщения

Поле *Message Length* определяет общую длину сообщения в байтах, начиная с поля *Message ID* и включая обязательные и необязательные параметры.

Поле *Message ID* имеет длину 32 бита и используется для того, чтобы однозначно идентифицировать уведомляющие сообщения протокола LDP.

Поле *Mandatory Parameters* представляет собой список обязательных параметров сообщения. Некоторые сообщения протокола не требуют никаких параметров. Если в сообщении предусмотрены обязательные параметры, они должны быть указаны в том порядке, в каком они следуют в этом сообщении.

Поле *Optional Parameters* содержит необязательные параметры. Они, в отличие от обязательных параметров, могут появляться в сообщениях в произвольном порядке или не появляться вообще.

3.4. Сообщения LDP

В рассматриваемых в этом параграфе сообщениях протокола LDP и его расширения CR-LDP будут встречаться элементы TLV из предыдущего параграфа. Чтобы вспомнить их назначение и формат, читателю достаточно вернуться на несколько страниц назад к соответствующему месту п. 3.3.3.

3.4.1. Уведомляющее сообщение Notification Message

Уведомляющие сообщения используются для передачи информации, на которую следует обратить внимание, и сообщений об ошибках. Выделяют два класса уведомлений:

- уведомления о фатальных ошибках, которые вынуждают маршрутизатор, получивший такое сообщение, прекратить сеанс и отбросить привязку соответствующих меток;
- уведомления справочного характера, используемые для передачи информации о предыдущих сообщениях или о сеансе в целом.

С учетом этого возможны следующие случаи передачи уведомляющих сообщений:

а) Искаженный PDU или сообщение, которые вынужденно отбрасываются. Блок данных протокола LDP (LDP PDU), полученный по TCP-соединению для LDP-сеанса, считается искаженным, если идентификатор LDP в заголовке PDU неизвестен получателю, или известен, но не является идентификатором LDP, который получатель связывает с одноранговым узлом для данной LDP-сессии; если версия протокола LDP не поддерживается получателем, или поддерживается, но не является той версией, которую согласовали между собой маршрутизаторы на этапе организации сеанса; если значение поля «Длина» заголовка PDU слишком мало (< 14) или слишком велико (> максимального размера PDU). Сообщение протокола LDP считается искаженным, если неизвестен его тип, если значение поля «Длина сообщения» слишком велико, (т.е. сообщение выходит за пределы PDU) или если в сообщении потерян хотя бы один необязательный параметр.

б) Неизвестные или искаженные TLV, содержащиеся в сообщении протокола LDP, требуют удаления этого сообщения. Параметр TLV, содержащийся в сообщении протокола LDP, которое получено по TCP-соединению сеанса LDP, считается искаженным, если значение поля «Длина» TLV слишком велико, т.е. TLV выходит за пределы сообщения, или неизвестен тип TLV, или значение TLV искажено, т.е. когда получатель не может декодировать значение TLV.

в) Срабатывание таймера KeepALive.

г) Одностороннее завершение сеанса.

д) События, связанные с сообщением Initialization, т.е. согласование инициированного сеанса оказалось неуспешным.

е) События, связанные с другими сообщениями, также могут привести к событиям, о которых необходимо информировать одноранговые узлы LDP-сеанса посредством сообщений Notification.

ж) Внутренние ошибки.

Структура сообщения *Notification Message* представлена на рис. 3.16.

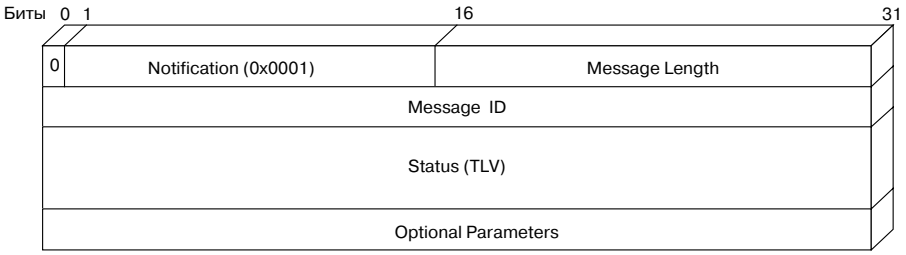


Рис. 3.16. Формат сообщения Notification Message

Необязательные параметры (в скобках после названия указывается код типа параметра):

Extended Status (0×0301) — 4-байтовое поле с информацией, которая дополняет сведения, содержащиеся в *Notification Status Code*.

Returned PDU (0×0302) — поле переменной длины, используемое LSR, чтобы вернуть часть PDU отправителю. Оно содержит заголовок PDU, за которым следует возвращаемая часть PDU.

Returned Message (0×0303) — поле переменной длины, используемое LSR, чтобы вернуть часть сообщения отправителю. Оно содержит поля типа и длины сообщения, за которыми следует возвращаемая часть сообщения.

3.4.2. Приветственное сообщение Hello

Структура сообщения *Hello* представлена на рис. 3.17. Оно используется маршрутизатором MPLS, чтобы информировать другие LSR в домене о том, что данный LSR готов работать в качестве узла тракта LSP с поддержкой LDP. Сообщение *Hello* периодически рассылается (с помощью транспортного протокола UDP) по адресу «всем маршрутизаторам данной подсети». Когда LSR получает такое сообщение, он начинает процедуру инициирования LDP, используя набор сеансовых сообщений протокола LDP. После успешного инициирования LDP маршрутизаторы становятся одноранговыми узлами и начинают обмен сообщениями (объявления метки).

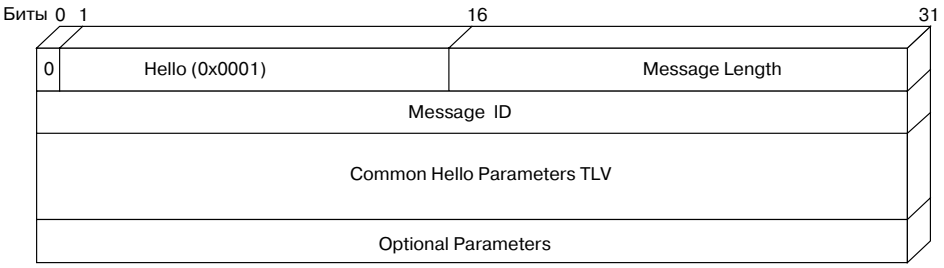


Рис. 3.17. Формат сообщения Hello

Блок *Common Hello Parameters* содержит общие для всех приветственных сообщений параметры TLV (рис. 3.18).

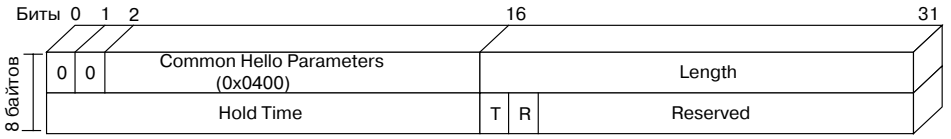


Рис. 3.18. Формат блока Common Hello Parameters

Поле *Hold Time* — содержит выраженное в секундах время, в течение которого актуальна и удерживается информация о соседних маршрутизаторах. До истечения этого времени LSR должен получить следующее сообщение *Hello*. Нулевое значение поля соответствует времени удержания по умолчанию, то есть 15 секундам для *Link Hello* и 45 секундам для *Targeted Hello.* Поле, целиком заполненное единицами, устанавливает таймер в бесконечность. При конфигурации системы рекомендуется, чтобы интервал между сообщениями Hello составлял не больше трети времени удержания.

Если бит *T* принимает значение 1, это указывает, что приветственное сообщение является целевым (*Targeted Hello*). Значение 0 указывает, что сообщение является общим.

Значение бита R, равное 1, запрашивает получателя сообщения периодически передавать целевые сообщения Hello к отправителю полученного сообщения Hello. Значение бита *R,* равное 0, означает отсутствие такого запроса.

Значение поля *Reserved* должно быть равно нулю при передаче и игнорироваться при приеме.

Поле *Optional Parameters* имеет переменную длину и может быть пустым или содержать несколько TLV. В текущей версии протокола определены следующие необязательные параметры:

Параметр	Тип (T)	Длина (L)	Значение (V)
Транспортный адрес IPv4	0×0401	4	*
Порядковый номер конфигурации	0×0402	4	**
Транспортный адрес IPv6	0×0403	16	***

(*) Специфицирует адрес IPv4, который будет использоваться передающим LSR при создании TCP-соединения для LDP-сеанса. Если этот необязательный TLV отсутствует, то должен использоваться адрес IPv4 отправителя пакета UDP, несущего сообщение Hello.

(**) Специфицирует 4-байтовый порядковый номер конфигурации передающего LSR. Позволяет принимающему LSR выявлять изменения конфигурации передающего LSR.

(***) Специфицирует адрес IPv6, который будет использоваться передающим LSR при создании TCP-соединения для LDP-сеанса. Если этот необязательный TLV отсутствует, то должен использоваться адрес IPv6 отправителя пакета UDP, несущего сообщение Hello.

3.4.3. Инициирующее сообщение Initialization

Сообщение Initialization относится к сеансовым сообщениям протокола LDP, которые инициируют TCP-сеанс с вновь обнаруженным одноранговым узлом LDP. К сеансовым сообщениям протокола LDP относятся рассматриваемое здесь сообщение Initialization, а также сообщения KeepAlive (Поддержание соединения), Address (Адрес) и Address Withdraw (Отмена адреса). Сообщение *Initialization* является первым сообщением, передаваемым в процедуре организации LDP-сеанса. Формат этого сообщения показан на рис. 3.19.

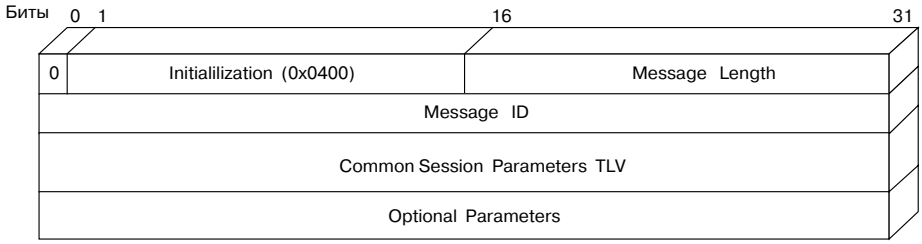


Рис. 3.19. Формат сообщения Initialization

Common Session Parameters — общие параметры сеанса. Этот TLV специфицирует предложенные передающим LSR значения параметров, которые должны согласовываться для каждого LDP-сеанса. Формат Common Session Parameters TLV показан на рис. 3.20.

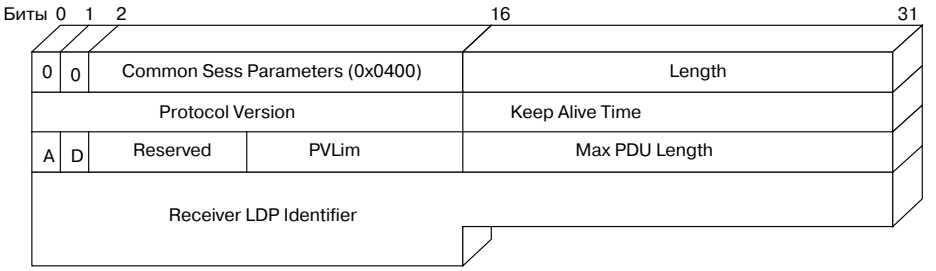


Рис. 3.20. Формат Common Session Parameters TLV

Protocol Version — номер версии протокола.

KeepAlive Time — число секунд, которое передающий LSR предлагает для значения времени KeepAlive.

A — указывает используемый режим объявления меток. Значение 0 означает *Downstream Unsolicited*. Значение 1 означает *Downstream On Demand*.

D — обнаружение петель. Указывает, используется ли механизм обнаружения закольцованных маршрутов на основе векторов пути. Значение 0 означает, что этот механизм не используется, значение 1 означает, что используется.

PVLim — лимит вектора пути. Представляет собой заданную максимальную длину вектора пути. Если механизм обнаружения петель не используется, этому полю присваивается значение 0. Когда механизм обнаружения закольцованных маршрутов действует только в части сети, рекомендуется, чтобы для всех LSR этой части сети был задан одинаковый лимит вектора пути.

Reserved — резервировано для будущего использования, имеет значение ноль при передаче и игнорируется при приеме.

Max PDU Length — максимальная длина PDU для данного сеанса. По умолчанию составляет 4096 октетов.

Receiver LDP Identifier — LDP-идентификатор получателя. Идентифицирует пространство меток получателя данных.

Optional Parameters — необязательные параметры. Поле переменной длины, содержит 0 или более параметров. Определены два вида необязательных параметров переменной длины: параметры сеанса, относящиеся к ATM (0×0501), и параметры сеанса, относящиеся к Frame Relay (0×0502).

3.4.4. Сообщение KeepAlive

Сообщение *Keep Alive* передается для контроля целости транспортного соединения при LDP-сеансе. Таймер KeepAlive перезапускается всякий раз, когда во время TCP-сеанса принимается блок данных LDP PDU. Каждый участвующий в сеансе маршрутизатор должен гарантировать, что сообщение KeepAlive передается для того, чтобы гарантировать целость соединения, даже если никаких других сообщений не передается. Формат сообщения показан на рис. 3.21. Какие-либо необязательные параметры для этого сообщения не определены.

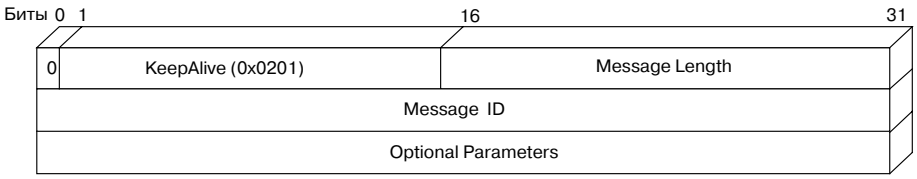


Рис. 3.21. Формат сообщения KeepAlive

3.4.5. Адресное сообщение Address

Сообщение *Address* передается одним одноранговым LSR другому, чтобы объявить адреса своих интерфейсов. Его формат представлен на рис. 3.22. Для сообщения Address также не определены какие-либо необязательные параметры. LSR, который получает сообщение Address, использует указанные в нем адреса для ведения базы данных, позволяющей устанавливать соответствие между LDP-идентификаторами одноранговых узлов и адресами следующих маршрутизаторов на пути пересылки пакета.

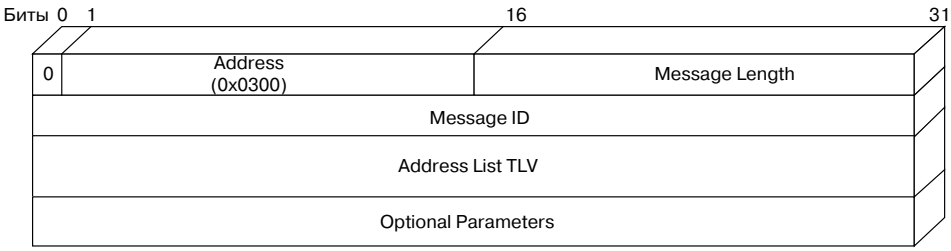


Рис. 3.22. Формат сообщения Address

3.4.6. Сообщение отмены адреса Address Withdraw

Сообщение передается одноранговому маршрутизатору, чтобы отменить ранее объявленный адрес интерфейса (рис. 3.23). Необязательные параметры этого сообщения не определены.

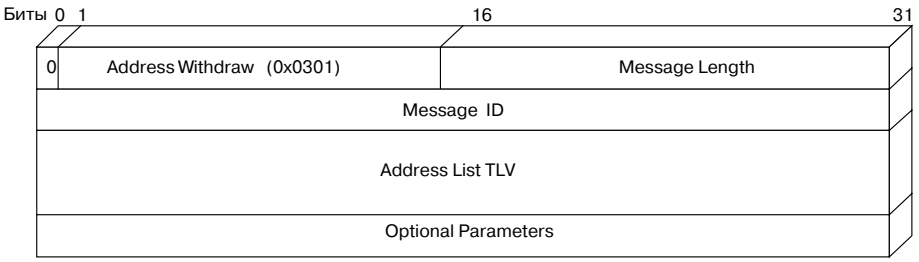


Рис. 3.23. Формат сообщения Address Withdraw

3.4.7. Сообщение привязки метки Label Mapping

После того как между двумя маршрутизаторами установлен LDP-сеанс, используется набор сообщений для обмена специальной информацией управления метками. Определены пять сообщений управления: Label Mapping (Назначение метки), Label Request (Запрос метки), Label Abort Request (Отмена запроса метки), Label Withdraw (Отмена привязки метки) и Label Release (Освобождение метки).

LSR передает сообщение *Label Mapping* к одноранговому узлу LDP-сеанса (верхнему маршрутизатору), чтобы уведомить его о привязке «метка-FEC». LSR отвечает за то, чтобы выданные им метки не были противоречивыми и чтобы его одноранговые узлы получили эти метки. Формат сообщения представлен на рис. 3.24.

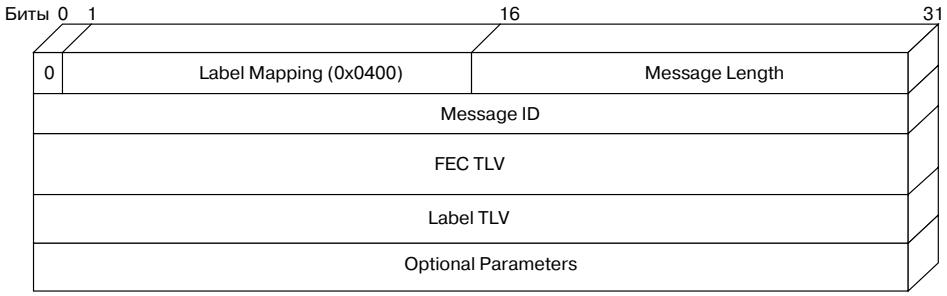


Рис. 3.24. Формат сообщения Label Mapping

Для этого сообщения определены следующие необязательные параметры:

Параметр	Длина (L)	Значение (V)
TLV «Идентификатор сообщения Запрос метки»	4	*
TLV «Число пересылок (Hop Count)»	1	см. 3.3.3
LV «Вектор пути (Path Vector)»	T - переменная	см. 3.3.3

(*) идентификатор сообщения запроса метки *(Label Request Message ID)* необходим, если сообщение Label Mapping передается в ответ на сообщение Label Request. Значением этого необязатель-

ного параметра является идентификатор сообщения Label Request, в ответ на которое передается Label Mapping.

Два других необязательных параметра встречаются не только в этом сообщении, и они уже были рассмотрены в предыдущем параграфе.

3.4.8. Сообщение запроса метки Label Request

LSR передает сообщение *Label Request* к одноранговому узлу LDP-сеанса, чтобы запросить метку для ее привязки к определенному FEC (рис. 3.25). Сообщение передается вышестоящим LSR к нижестоящему LSR, тот назначает метку для FEC, указанного в запросе, и сообщает ее вышестоящему LSR. LSR может передавать сообщение Label Request при возникновении любой из следующих ситуаций:

- LSR распознает новый FEC с помощью таблицы пересылки, следующим маршрутизатором является одноранговый узел LDP-сеанса, данный LSR еще не получил метку для нового FEC от следующего маршрутизатора;
- изменяется следующий маршрутизатор для известного FEC, а LSR еще не получил метку для этого FEC от нового следующего маршрутизатора;
- LSR получает запрос метки для FEC от вышестоящего маршрутизатора, следующим маршрутизатором для этого FEC является одноранговый узел LDP-сеанса, а LSR еще не получил привязку метки к FEC от следующего маршрутизатора.

LSR, получивший запрос, должен ответить на сообщение Label Request сообщением Label Mapping, содержащим значение метки, или сообщением Notification с указанием причины, по которой удовлетворить запрос невозможно. В базовой версии протокола LDP для сообщения *Notification* определены следующие коды ситуаций (типы ответов), в которых невозможно удовлетворить запрос метки:

- No Route (нет маршрута). FEC, для которого запрошена метка, содержит элемент, не обеспеченный ресурсами и не имеющий маршрута.
- No Label Resources (нет свободных меток). LSR не может предоставить метку из-за отсутствия ресурсов пула меток. Когда такие ресурсы появятся, LSR должен уведомить об этом маршрутизатор, который запрашивал метку, передав ему сообщение Notification с кодом ситуации Resources Available (Имеются ресурсы меток). LSR, который в ответ на сообщение Label Request получил сообщение Notification с кодом «Отсутствуют ресурсы меток», не должен генерировать новые сообщения Label Request до тех пор, пока не получит сообщение Notification с кодом «Имеются ресурсы меток».

- Loop Detected (выявлен закольцованный маршрут). LSR выявил передачу сообщения Label Request по кольцу.

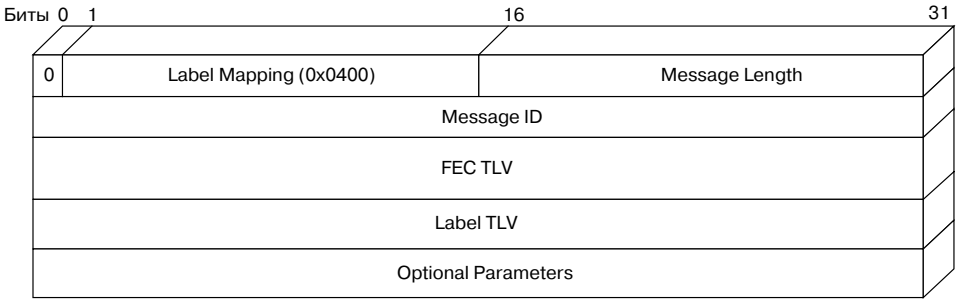


Рис. 3.25. Формат сообщения Label Request

Определены следующие необязательные параметры, уже рассмотренные в предыдущем параграфе:

Параметр	Длина (L)	Значение (V)
TLV «Число пересылок»	1	см. 3.3.3
TLV «Вектор пути»	Переменная	см. 3.3.3

3.4.9. Сообщение отмены запроса метки Label Abort Request

Сообщение *Label Abort Request* может использоваться для отмены ранее переданного и еще не подтвержденного сообщения Label Request. Верхний LSR (LSRu) может передать сообщение Label Abort Request для отмены ждущего обработки сообщения Label Request, переданного им ранее к нижестоящему LSR (LSRd), при следующих обстоятельствах:

- следующий маршрутизатор LSRd для вышестоящего LSRu в отношении данного FEC изменился на некий LSRx, или верхний LSR не является входным LSR, не поддерживает слияние FEC и получил сообщение отмены запроса метки для данного FEC от вышестоящего LSRy;
- верхний LSR не является входным LSR, поддерживает слияние FEC и получил сообщение отмены запроса метки для данного FEC от вышестоящего LSRy, а этот LSRy является самым верхним LSR, запрашивающим метку для данного FEC.

Могут возникать и другие ситуации, в которых LSR захочет отменить свой запрос метки для освобождения ресурса, связанного с еще не установленным LSP.

Когда LSR получает сообщение Label Abort Request и еще не ответил сообщением Label Mapping или сообщением Notification какого-либо типа на сообщение Label Request, которое теперь нужно отменить, он должен подтвердить отмену запроса, пере-

дав сообщение Notification типа Label Request Aborted (запрос метки отменен). В это сообщение должен входить TLV «Идентификатор сообщения Label Request», идентифицирующий отменяемое сообщение.

Если LSR получает сообщение Label Abort Request после того, как он ответил сообщением Label Mapping или Notification на сообщение Label Request, которое теперь нужно отменить, он игнорирует сообщение Отмена запроса метки. Структура сообщения Label Abort Request представлена на рис. 3.26.

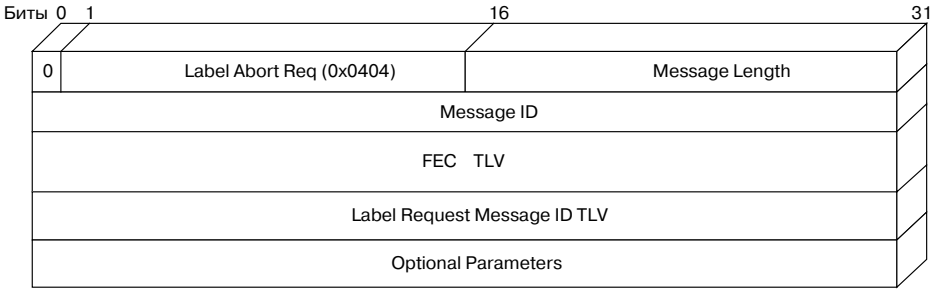


Рис. 3.26. Формат сообщения Label Abort Request

Label Request Message ID TLV — идентификатор сообщения с запросом метки. Идентифицирует сообщение Label Request, которое должно быть отменено.

Для сообщения Label Abort Request не определено никаких необязательных параметров.

3.4.10. Сообщение отмены привязки метки Label Withdraw

Сообщением *Label Withdraw* LSR информирует одноранговый узел о том, что тот не должен больше использовать ранее определенную привязку меток к FEC. LSR передает сообщение Label Withdraw, когда он больше не опознает FEC, для которого он ранее назначил метку, или в одностороннем порядке решил больше не коммутировать по меткам пакеты этого FEC. FEC TLV специфицирует FEC, метки для которого должны быть отменены. Если за FEC TLV не следует Label TLV, то отменяются все метки, связанные с этим FEC; если Label TLV имеется, то удаляется только указанная в нем метка. LSR, который получил сообщение Label Withdraw, должен в ответ передать сообщение Label Release. Структура сообщения Label Withdraw представлена на рис. 3.27.

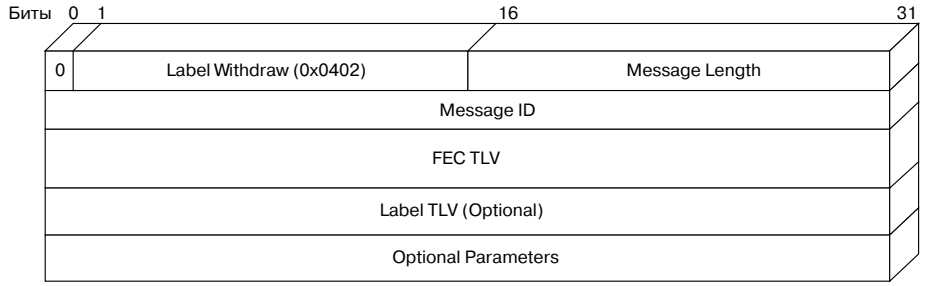


Рис. 3.27. Формат сообщения Label Withdraw

Определен только один необязательный параметр:

Параметр	Длина (L)	Значение (V)
Label TLV	Переменная	см. 3.3.3

3.4.11. Сообщение освобождения метки Label Release

LSR передает сообщение *Label Release* одноранговому узлу LDP-сеанса в любой из следующих ситуаций: LSR, который передал сообщение Label Mapping, больше не является смежным маршрутизатором для данного FEC, и используется консервативный режим распределения меток, или LSR получил сообщение Label Mapping от несмежного маршрутизатора, и используется консервативный режим распределения меток, или LSR получил сообщение Label Withdraw. Формат сообщения Label Release показан на рис. 3.28.

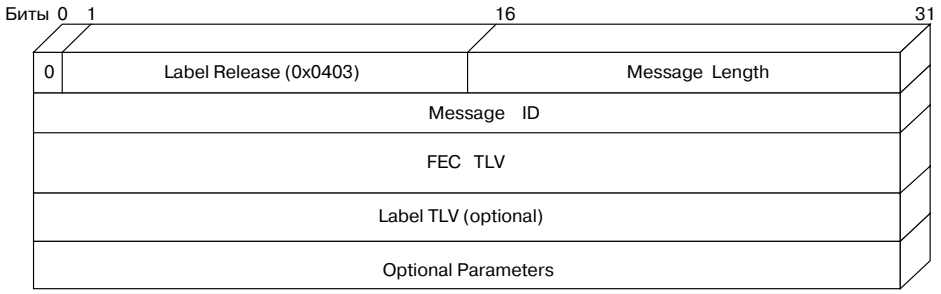


Рис. 3.28. Формат сообщения Label Release

3.4.12. Дополнительные сообщения и TLV

Напомним, что согласно табл. 3.1 в базовом протоколе LDP предусмотрены сообщения еще двух типов: *LDP Vendor-private Messages* и *Experimental Messages,* т.е. те или иные производители могут специфицировать сообщения для своих реализаций LDP, а также могут вводиться экспериментальные сообщения. Точно так же, в сообщения могут входить определяемые производителем дополнительные TLV, для которых предусмотрены отдельные диапазоны кодов.

3.5. Протокол CR-LDP

Протокол LDP может следовать только таблицам маршрутизации IP. Чтобы преодолеть это ограничение, было предложено расширение LDP, названное CR-LDP.

Протокол CR-LDP является вариантом LDP, в котором определены механизмы создания и поддержания трактов LSP с явно заданным маршрутом. Для создания тракта CR-LSP используется больше информации, чем можно получить от традиционных протоколов внутренней маршрутизации. CR-LDP используется для таких приложений MPLS, как TE и QoS, где требуется дополнительная информация о маршрутах. В этом протоколе запрос метки может не следовать слепо вдоль дерева маршрутизации для данного адресата, т.к. предусмотрена возможность точно сообщить, как он должен следовать, включив в сообщение явно заданный маршрут. При этом программное обеспечение CR-LDP не использует для маршрутизации таблицы пересылки, а маршрутизирует запрос в соответствии с содержащимися в сообщении инструкциями.

Протокол CR-LDP не поддерживает динамического вычисления явно задаваемых маршрутов, поэтому сведения о динамическом резервировании пропускной способности должны включаться в вещательную информацию протоколов OSPF или IS-IS, или в извещения о состоянии каналов LSA. Используя эти механизмы, CR-LDP может занимать и резервировать пропускную способность. Доступная пропускная способность изменяется в соответствии с запросом, и ее новое значение рассылается другим узлам с помощью расширений протоколов OSPF и IS-IS, которые будут рассмотрены ниже. Новые маршруты можно вычислить затем с учетом принятых ограничений, используя модифицированный алгоритм Дийкстры. В результате протокол CR-LDP получает в свое распоряжение явный маршрут для организации LSP. Тракт создается посредством сообщения запроса метки, содержащего динамически вычисляемый явный маршрут. К этому вопросу мы еще вернемся далее.

Протокол CR-LDP имеет также другие, новые по сравнению с базовой версией LDP функциональные возможности:

- явная маршрутизация с точно определенными и свободными маршрутами, при которой маршрут задается в виде последовательности групп узлов. В том случае, если в группе указано более одного маршрутизатора, при создании явного маршрута возможна определенная гибкость;
- спецификация параметров трафика (например, пиковая скорость передачи, гарантированная скорость передачи и допустимая вариация задержки);

- закрепление маршрута (route pinning), которое может использоваться в тех случаях, когда изменять трассу LSP нежелательно, например, в сегментах со свободной маршрутизацией, когда в этом сегменте становится доступным лучший маршрут;
- механизм приоритетного вытеснения LSP с помощью системы приоритетов создания и удержания. Ранжируются существующие тракты LSP (приоритет удержания) и новые тракты LSP (приоритет создания) с тем, чтобы определить, может ли новый LSP вытеснять существующий LSP. Для приоритетов предложен диапазон значений от 0 (высший приоритет) до 7 (низший приоритет);
- введены новые коды состояний LSR, указанные в табл. 3.2;
- введен LSPID — уникальный идентификатор тракта CR-LSP в сети;
- введены классы (цвета) сетевых ресурсов, назначаемые администратором сети.

Таблица 3.2. Новые коды состояния LSR в протоколе CR-LDP

Статус (Status Code)	Кодировка (Type)
Bad Explicit Routing TLV Error	0×04000001
Bad Strict Node Error	0×04000002
Bad Loose Node Error	0×04000003
Bad Initial ER-Hop Error	0×04000004
Resource Unavailable	0×04000005
Traffic Parameters Unavailable	0×04000006
LSP Preempted	0×04000007
Modify Request Not Supported	0×04000008

В протоколе CR-LDP не появилось новых сообщений, но модифицированы некоторые сообщения базовой версии (например, *Label Request, Label Mapping*), а некоторые остались без изменений (*Label Release , Label Withdraw, Label Abort Request*).

Добавлен ряд новых TLV, передаваемых в модифицированных сообщениях или в качестве необязательных параметров в других сообщениях:

- TLV явного маршрута (Explicit route TLV) — специфицирует маршрут создаваемого тракта LSP, содержит одно или более полей Explicit route hop TLV.
- TLV сетевых узлов явного маршрута (Explicit route hop TLV) — серия полей переменной длины, в каждом из которых указывается адрес маршрутизатора(ов).
- TLV параметров трафика (Traffic parameters TLV) — описывает параметры трафика.
- TLV вытеснения (Preemption TLV) — содержит данные о приоритетах создания и удержания.

- LSPID TLV — содержит уникальный идентификатор LSP, состоящий из идентификатора входного LSR (или его IP-адреса) и локально уникального идентификатора LSP для этого LSR.
- Поле класса (цвета) ресурса (Resource class (color) TLV) — определяет, какие типы каналов передачи приемлемы для LSP. Выражается 32-битовым кодом.
- TLV закрепления маршрута (Route pinning TLV) — указывает на наличие или отсутствие запроса закрепления маршрута.

Хотя CR-LDP обладает довольно широкими возможностями инжиниринга трафика (*TE — Traffic Engineering*) в сетях MPLS и не требует реализации в оборудовании дополнительного протокола, а лишь расширения уже существующего, но в самое последнее время по ряду косвенных данных прослеживается сворачивание работ над CR-LDP в IETF в пользу протокола RSVP-TE. По этой причине CR-LDP рассмотрен в книге довольно кратко, а протоколу RSVP целиком посвящена глава 4.

3.6. Аспекты безопасности LDP

Сэкономленное за счет краткости описания CR-LDP место мы уделим вопросам информационной безопасности, связанным с применением протокола LDP.

3.6.1. Несанкционированные действия

Имеется два типа взаимодействий по протоколу LDP, которые могут стать объектом атаки со стороны нарушителей.

Сообщения обнаружения, переносимые по транспортному протоколу UDP. LSR, связанные друг с другом на уровне звена данных, обмениваются базовыми сообщениями Hello. Угроза несанкционированных сообщений Hello может быть снижена путем приема базовых сообщений Hello только через интерфейсы, прямо связанные с теми LSR, которым можно доверять, и/или за счет того, что будут игнорироваться базовые сообщения Hello, не адресованные «Ко всем маршрутизаторам» в группе многоадресной рассылки данной подсети. LSR, которые не связаны прямым звеном данных, могут извещать о своей готовности установить LDP-сеанс расширенными сообщениями Hello (Extended Hello). LSR может снизить риск появления несанкционированных расширенных сообщений Hello путем их фильтрации и приема только тех из них, которые исходят от источников, включенных в список разрешенных пользователей. Однако процедура фильтрации отнимает много ресурсов LSR.

Информация о сеансе, переносимая по транспортному протоколу TCP. Протокол LDP может использовать в качестве опции сигнатуру TCP Message Digest 5 (MD5). Это предотвращает введе-

ние имитированных сегментов TCP в LDP-сеансы. Использование сигнатуры TCP MD5 в протоколе LDP аналогично ее использованию в протоколе BGP (глава 7), которое специфицировано в RFC 2385 «Protection of BGP Sessions via the TCP MD5 Signature Option». Сам алгоритм MD5 представлен в RFC 1321 «The MD5 Message-Digest Algorithm». Однако LDP может использовать и любую другую методику, обеспечивающую защиту TCP-сообщений. Следовательно, когда будет специфицирована методика, более сильная, чем MD5, протокол LDP будет относительно просто модернизировать.

3.6.2. Конфиденциальность

Протокол LDP не предусматривает никакого специального механизма, обеспечивающего конфиденциальность распределения меток, но сам механизм, гарантирующий аутентичность и сохранность сообщений протокола LDP, дает уровень защиты не хуже того, который дают протоколы маршрутизации.

Известны высказывания, что для снижения угрозы несанкционированных действий распределение меток требует конфиденциальности. Однако такая конфиденциальность не защитит от имитации меток, поскольку пакеты данных переносят метки в явном, незашифрованном виде. Более того, такого рода атаки возможны и без знания злоумышленником конкретной привязки метки к FEC. Для предотвращения имитации меток необходима гарантия того, что пакеты получают метки от надежных и достоверных LSR, и что эти метки надлежащим образом контролируются маршрутизаторами, которые их назначают.

3.6.3. Отказ в обслуживании

LDP имеет две потенциальные мишени для атак, приводящих к отказам в обслуживании.

Известный (*well-known*) UDP-порт для процедур обнаружения. Администратор LSR может принимать меры для устранения угроз передачи несанкционированных базовых сообщений Hello, приводящих к отказам в обслуживании, гарантируя, что LSR непосредственно подключен только к надежным узлам, от которых нельзя ожидать инициирования подобных атак. Если существует надежная и достоверная область MPLS, то маршрутизаторы, расположенные на границе этой области, могут использоваться для защиты внутренних LSR от фатальных атак.

Известный TCP-порт для организации LDP-сеанса. Подобно другим управляющим протоколам, которые в качестве транспортного протокола используют TCP, протокол LDP может стать мишенью атак, приводящих к отказу в обслуживании, например SYN-атак.

В этом отношении LDP уязвим в той же степени, что и другие управляющие протоколы, использующие протокол TCP. Применение на границе области MPLS списков доступа (списков разрешенных пользователей) способом, аналогичным предложенному выше для расширенных сообщений Hello, позволяет защитить внутреннюю область MPLS от атак из-за пределов области.

3.7. Сигнализация LDP

Следуя традиции, сложившейся в серии книг «Телекоммуникационные протоколы», в завершение главы рассмотрим некоторые сценарии распределения меток с помощью протокола LDP.

Как уже отмечалось, маршрутизаторы LSR могут использовать либо независимый, либо упорядоченный способ распределения меток. При независимой раздаче меток LSR может объявлять метки одноранговым объектам LDP в любой момент, когда пожелает, т.е. может передать сообщение Label Mapping в вышестоящий LSR до получения сообщения Label Mapping от нижестоящего LSR. При упорядоченной раздаче LSR может объявлять метку вышестоящим LSR только тогда, когда он имеет привязку «FEC-метка» для следующего участка, или когда сам является выходным маршрутизатором MPLS.

Если мы комбинируем режим «нижестоящим-по-требованию» с упорядоченным способом раздачи меток, то их распределение будет происходить следующим образом. Входной LSR передает сообщение Label Request в нижестоящий LSR, и это сообщение будет продвигаться по сети до выходного LSR. Выходной LSR отвечает соседнему вышестоящему LSR сообщением Label Mapping. Этот вышестоящий LSR запоминает привязку «FEC-метка» для своего выходного интерфейса, создает новую привязку «FEC-метка» для своего входного интерфейса и информирует об этой привязке следующий вышестоящий LSR с помощью собственного сообщения Label Mapping. Процесс продолжается до тех пор, пока входной LSR не получит сообщение Label Mapping в ответ на свой первоначальный запрос. Сценарий сигнализации для этого случая показан на рис. 3.29.

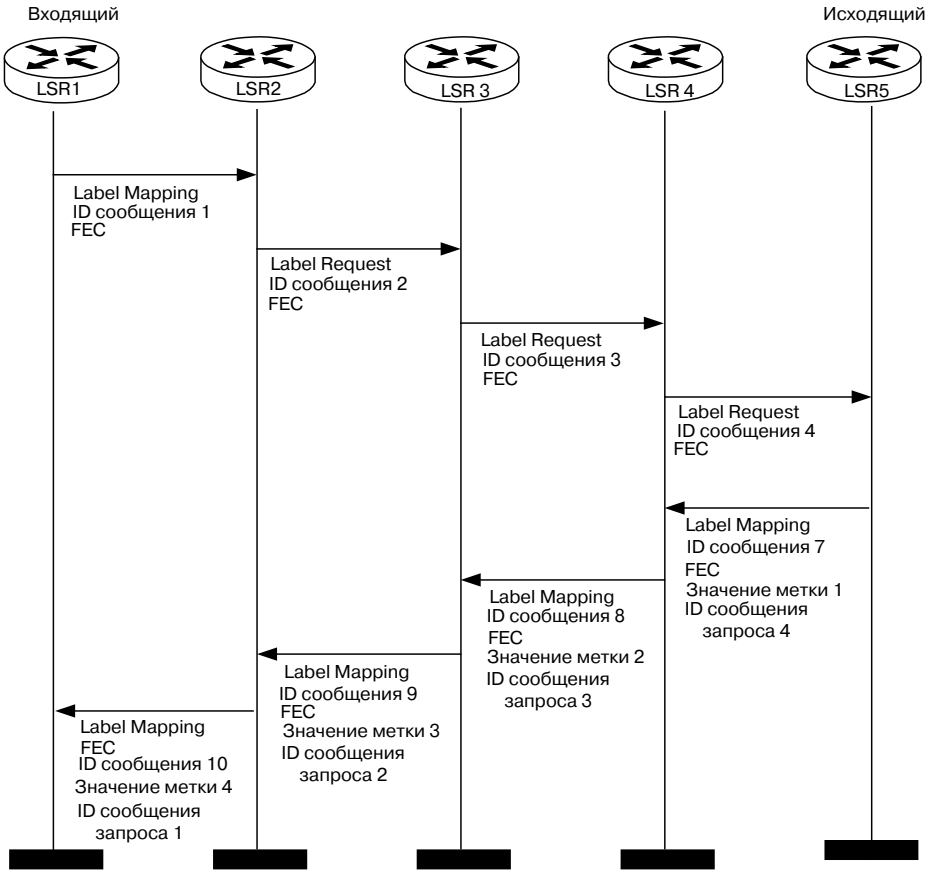


Рис. 3.29. Организация LSP в режиме «нижестоящим по требованию» с упорядоченным распределением меток

Для правильного восприятия приведенного на рис. 3.29 сценария напомним, что LDP является протоколом прикладного уровня, использующим в качестве транспортных протоколов для связи с другими LSR, которые могут стать его одноранговыми узлами, как протокол UDP, так и протокол TCP (рис. 3.30). В частности, транспортный протокол UDP используется LSR для передачи сообщений обнаружения, информирующих другие LSR о том, что данный LSR является потенциальным кандидатом на роль однорангового узла LDP, а транспортный протокол TCP используется для передачи всех других сообщений, в частности, сообщений *Session, Advertisement* и *Notification*, которым требуется механизм надежной и своевременной их доставки, что и обеспечивает TCP.

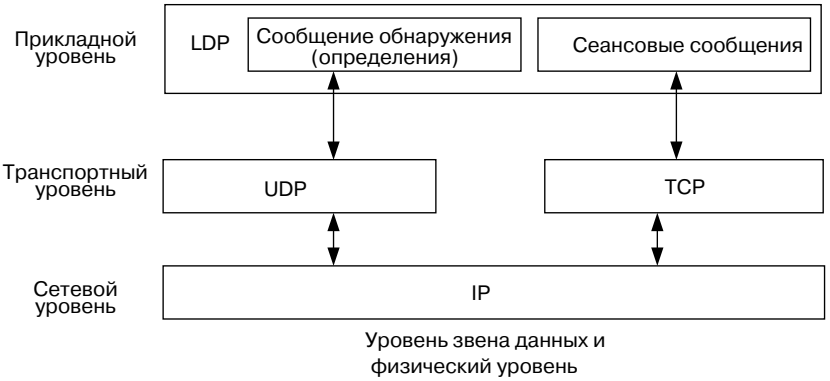


Рис. 3.30. LDP как протокол прикладного уровня

Использование транспортного уровня протоколом LDP [14] показано на рис. 3.30. UDP-портом по умолчанию для сообщений *Hello* протокола LDP является порт 646. TCP-портом по умолчанию для всех других сообщений протокола LDP также является порт 646.

Сценарий на рис. 3.31 построен в строгом соответствии с этой иерархией и с другим изложенным в данной главе материалом. Прежде всего, протокол LDP должен обнаружить маршрутизаторы LSR, с которыми возможна организация LDP-сеансов. Для этой цели в LDP предусмотрено два механизма: базовый и расширенный. Базовый механизм обеспечивает обнаружение LSR путем периодической передачи сообщений *Hello* к UDP-порту 646 по IP-адресу многоадресной рассылки «Все маршрутизаторы подсети» (224.0.0.2). Расширенный механизм используется для обнаружения LSR, не имеющих прямой связи (на уровне звена данных) с данным маршрутизатором. При этом пакеты *Hello* передаются по IP-адресу запрашиваемого LSR.

В сообщениях *Hello* передается LDP-идентификатор *пространства меток*, которое LSR, передающий эти сообщения, намерен использовать на следующем этапе процесса распределения меток — при создании соединения между маршрутизаторами по протоколу TCP, — а также вспомогательная информация. Сведения о смежных LSR, полученные посредством сообщений *Hello*, действительны в течение времени, заданного в этих сообщениях. По истечении заданного времени они могут быть аннулированы, о чем инициатор должен известить встречный LSR уведомляющим сообщением *Notification Message*, если только таймер не установлен на бесконечность, или если до срабатывания таймера не получено другое сообщение *Hello*.

После обмена сообщениями *Hello*, одноранговые узлы устанавливают TCP-соединение через порты 646. Сеанс TCP-связи инициируется тем из двух LSR, чей *транспортный адрес* (идентификатор пространства меток) короче. LSR, инициировавший TCP-соединение, является *ведущим (master)* маршрутизатором сеанса.

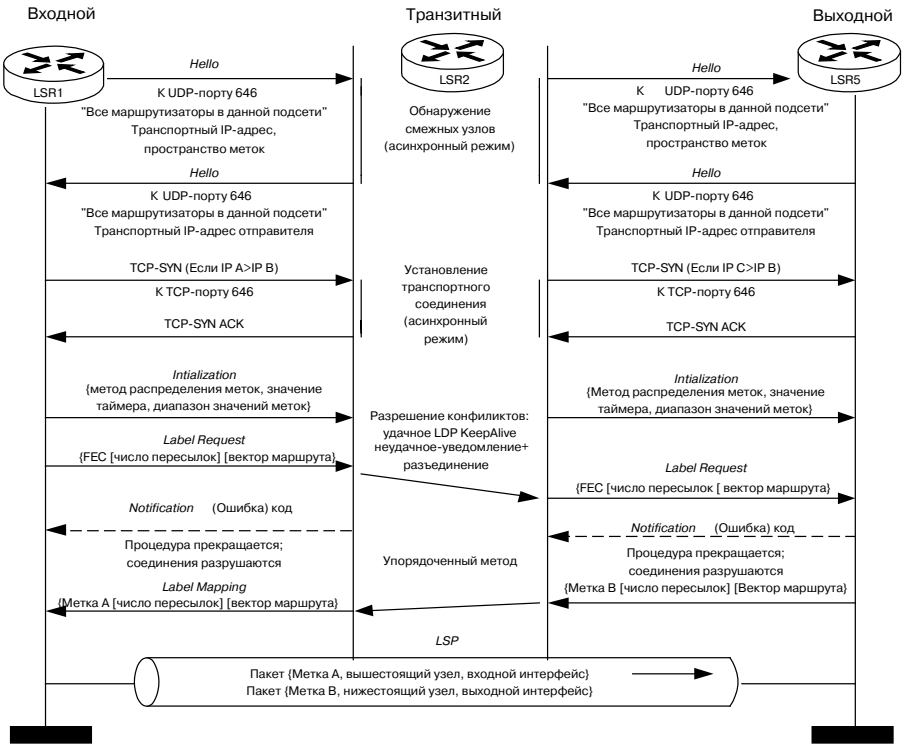


Рис. 3.31. Сигнализация LDP для распределения меток

После того как TCP-соединение создано, оба LSR, по инициативе ведущего маршрутизатора, обмениваются сообщениями *Initialization*. Этими сообщениями они информируют друг друга о версии протокола LDP, о желательной дисциплине распределения меток, о диапазонах значений меток для ATM (VPI/VCI) и Frame Relay (DLCI), о значении таймера KeepAlive, а также о других параметрах *TLV*.

Этап инициирования сеанса связи завершается обменом между LSR сообщениями *KeepAlive*. Таким образом, поверх TCP-соединения организуется LDP-сеанс. При наличии неразрешенных конфликтов и проблем с совместимостью, LSR передает уведомление об ошибке, и LDP-сеанс разрушается (на представ-

ленной диаграмме такие уведомления обозначены пунктирными линиями). В этом случае ведущий LSR может предпринять, но не ранее чем через 15 секунд, повторную попытку установить связь, причем при нескольких отказах подряд выдержка времени каждый раз увеличивается, достигая максимально 2 минут.

Установив LDP-сеанс, каждый LSR ведет мониторинг прихода сообщений LDP в рамках этого сеанса. Если в течение времени, установленного таймером KeepAlive, от смежного узла не получено никакой информации, LSR закрывает LDP-сеанс, разрушая транспортное соединение и оповещая об этом встречный маршрутизатор сообщением *Shutdown*. Каждый раз, когда от корреспондента принимается сообщение, таймер KeepAlive сбрасывается; таким образом, оба участника соединения должны обеспечивать передачу любого протокольного сообщения внутри интервала времени, заданного таймером. Если маршрутизатор не имеет полезной информации, он должен передавать сообщение *KeepAlive*.

Итак, LDP-сеанс организован, и теперь LSR может запросить метку, передав сообщение *Label Request*. Это сообщение предусматривает ограничение максимального числа пересылок для предотвращения бесконечной циркуляции запроса по сети при возникновении закольцованных маршрутов. Обязательным параметром сообщения *Label Request* является параметр FEC TLV, указывающий, для какого FEC запрашивается метка. В состав сообщения *Label Request* может также входить рассмотренный выше вектор маршрута (Path Vector), представляющий собой перечень всех маршрутизаторов, через которые прошло это сообщение.

Если в это время какой-либо узел столкнется с проблемами, связанными с совместимостью, с отсутствием ресурсов и т.д., вышестоящему маршрутизатору, запрашивающему метку, будет передано соответствующее уведомление, и оно будет пересылаться дальше по цепочке, пока не дойдет до входного пограничного маршрутизатора. Все соединения между маршрутизаторами на этом маршруте будут разрушены. Если сообщение *Label Request*, в конце концов, благополучно дойдет до выходного маршрутизатора (при упорядоченном режиме), в выходном маршрутизаторе будет генерироваться сообщение *Label Mapping*, содержащее метку, которая имеет локальное значение на участке между выходным и соседним с ним вышестоящим маршрутизатором. Если на всех следующих далее вышестоящих маршрутизаторах успешно пройдет привязка меток к FEC, то, после обработки во входном пограничном маршрутизаторе сообщения *Label Mapping*, полученного от соседнего с ним нижестоящего маршрутизатора, маршрут для LSP будет создан.

Глава 4
Протокол RSVP для MPLS

4.1. Стили резервирования ресурсов для MPLS

Протокол резервирования ресурсов RSVP (Resource reSerVation Protocol) был разработан в исследовательском центре Xerox в Пало-Альто, как, впрочем, и целый ряд других рассматриваемых в книге протоколов. В разработке RSVP участвовал также Институт научной информации ISI (Information Science Institute) при Южнокалифорнийском университете, а первая открытая версия протокола в составе архитектуры интегрированного обслуживания IntServ была специфицирована IETF в документе RFC 2205 в 1997 году. В скором времени протокол RSVP стал восприниматься как самостоятельный механизм, обеспечивающий гарантированное качество обслуживания (QoS) и призванный решить проблему передачи по IP-сетям трафика реального времени. С точки зрения обеспечения QoS главным конкурентом протокола RSVP была появившаяся примерно на год позже модель DiffServ. Обсуждение достоинств и недостатков этих двух подходов продолжалось вплоть до появления технологии MPLS, после чего многим стало казаться, что и DiffServ, и RSVP исчезнут, так и не проявив себя на практике.

Вопреки этим ожиданиям RSVP, как и DiffServ, не найдя широкого самостоятельного применения, успешно влился в технологию MPLS, способствуя, наряду с CR-LDP, улучшению ее характеристик. К использованию модели DiffServ мы вернемся в следующих главах, а здесь рассмотрим применение протокола RSVP и его расширения RSVP-TE.

Первой из двух функций, возложенных на RSVP технологией MPLS, является распределение меток (вместо рассмотренного в предыдущей главе протокола LDP). Вторая, традиционная для RSVP роль заключается в поддержании QoS в сети MPLS. Действительно, вне зависимости от используемого протокола распределе-

ния меток, маршрутизаторы LSR должны согласовать между собой параметры QoS для каждого FEC. Метки позволяют определить огромное число классов QoS, но реально в типичных мультисервисных сетях, даже при очень большом количестве классов FEC, будут существовать, как правило, всего несколько классов QoS.

Использовать метки MPLS намного проще, чем распределить их по маршрутизаторам, составляющим тракт LSP. Основными средствами распределения меток являются обсуждавшиеся в предыдущей главе протокол LDP и его расширения, рассматриваемые здесь расширения протокола резервирования ресурсов RSVP, а также протокол пограничных шлюзов BGP, которому посвящена глава 7. На рис. 4.1 представлена процедура назначения и распределения меток с использованием расширенного протокола RSVP.

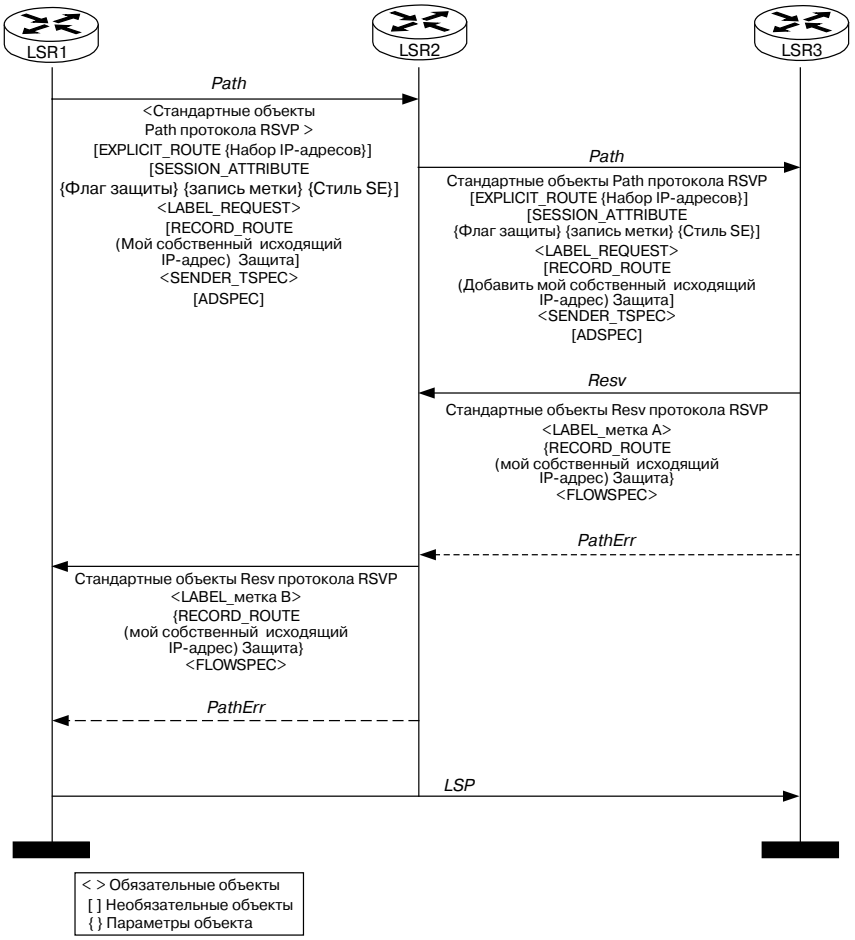


Рис. 4.1. Процедура распределения меток с помощью протокола RSVP

Как это уже упоминалось в контексте LDP, организация LSP может быть *независимой* или *упорядоченной*. В первом случае LSR в одностороннем порядке принимает решение о том, каким образом привязать метку к определенному FEC: он выбирает метку и уведомляет о ее привязке вышестоящий LSR, запросивший метку. Во втором случае привязку меток к FEC начинает выходной LSR, и процедура привязки последовательно продвигается вверх в сторону входного LSR. При этом единственный узел, принимающий «независимое» решение о привязке — это выходной LSR. Все другие (вышестоящие) LSR ждут, пока каждый из них не получит значение метки от смежного нижестоящего маршрутизатора, произведет привязку метки к FEC и уведомит о своей привязке ближайший вышестоящий маршрутизатор.

Эти два метода функционально несовместимы. Упорядоченный метод лучше подходит для предотвращения образования закольцованных маршрутов, и он может быть реализован с использованием либо протокола RSVP, либо протокола LDP.

Запрос резервирования RSVP включает в себя две опции, одна из которых описывает *способ резервирования (Reservations),* а другая — *выбор отправителей (Sender Selection)*, определяющих направление *потоков (flows).* В одних случаях каждому отправителю ставится в соответствие определенная спецификация фильтра, в других случаях таких спецификаций не требуется вовсе. Протокол RSVP определяет три стиля резервирования:

- раздельное, с фиксированным фильтром — стиль Fixed Filter (FF),
- совместное явное — стиль Shared Explicit (SE),
- совместное, с произвольной фильтрацией — стиль Wildcard Filter (WF).

Эти стили резервирования и способы выбора отправителя, которым они соответствуют, приведены в табл. 4.1.

Таблица 4.1. Стили резервирования

Выбор отправителя	Резервирование	
	Раздельное	Совместное
Явный (explicit)	Стиль FF	Стиль SE
Произвольный (wildcard)	—	Стиль WF

Согласно табл. 4.1 стиль *WF (Wildcard-Filter)* соответствует опциям *совместного (shared) резервирования* и *произвольного (wildcard) выбора отправителя*. Стиль WF предусматривает групповой резервный ресурс, который делится между потоками всех отправителей. Этот резерв может рассматриваться как «труба», диаметр которой соответствует ресурсу, наибольшему из запрошенных получателями, и не зависит от числа отправителей. Таким образом, создается совместный LSP «группа точек — точка». Символически запрос резервирования стиля WF можно представить как

WF(*{Q}),

где звездочка представляет произвольную подстановку при выборе отправителя, а Q — спецификацию потока flowspec.

Стиль *FF (Fixed-Filter)* соответствует опциям *раздельного (distinct) резервирования* и *явного (explicit) выбора отправителя.* Таким образом, запрос резервирования стиля FF создает резервный ресурс для информационных пакетов, идущих от определенного отправителя, без совместного с другими отправителями использования этого ресурса в пределах одного и того же сеанса. При этом каждый отправитель использует уникальную метку, а сам идентифицируется на основе IP-адреса и LSP-ID. Для каждой пары отправитель/получатель создается LSP «точка-точка». Символически запрос резервирования FF можно представить как:

FF(S{Q}),

где S — выбранный отправитель. RSVP позволяет применять стиль резервирования FF одновременно для нескольких потоков. При этом формируется список дескрипторов потоков:

FF(S1{Q1}, S2{Q2}, ...).

Стиль *SE (Shared Explicit)* соответствует опциям *совместного (shared) резервирования и явного (explicit) выбора отправителя*. Таким образом, стиль SE формирует резервный ресурс, который используется совместно несколькими отправителями. В отличие от стиля WF, стиль SE позволяет получателю непосредственно специфицировать набор отправителей, и так как пользователи перечислены в сообщении *Resv*, разным пользователям могут быть присвоены различные метки, и для них могут быть созданы разные LSP. Если в сообщениях *Path* отсутствовали объекты ERO или если эти объекты были одинаковыми для всех пользователей, то может быть создан LSP «группа точек-точка». Запрос резервирования типа SE можно представить символически как:

SE((S1,S2,...){Q}).

Раздельное резервирование WF и SE пригодно для вещательных приложений, где несколько отправителей данных редко ведут передачу одновременно. Примером раздельного резервирования может служить случай с пакетной передачей речи. Каждый получатель может направить запрос резервирования WF или SE для удвоения полосы пропускания, отводимой одному отправителю, что позволяет говорить обоим партнерам одновременно. С другой стороны, стиль FF, который предусматривает резервирование для потоков отдельных отправителей, подходит для передачи видеосигналов.

Из этих трех стилей для обслуживания неоднородного мультисервисного трафика, проходящего через узлы сети MPLS, интерес представляют стили резервирования SE и FF, которые и рассматриваются в этой главе.

При резервировании FF полоса пропускания резервируется для каждого отправителя данных индивидуально. Резервные ресурсы и метки не используются совместно потоками разных отправителей и, следовательно, резервированная полоса пропускания через узел представляет собой сумму индивидуальных резервных полос пропускания. Из двух названных выше стилей резервирования FF более прост, но он не позволяет нескольким потокам использовать резервированную полосу пропускания совместно.

При использовании стиля резервирования SE объединение отправителей в группу, использующую общий ресурс резервирования, производится по усмотрению получателя. Однако для разных отправителей данных могут назначаться разные метки, и поэтому для каждого отправителя может существовать отличный от других тракт LSP.

Применение резервирования этих двух стилей будет рассмотрено ниже, а прежде целесообразно изложить основные принципы протокола RSVP.

4.2. Основы протокола RSVP

Упрощенно говоря, протокол RSVP является механизмом, который позволяет приложениям информировать сеть о своих требованиях. На основе этой информации сеть может либо резервировать внутри себя ресурсы, чтобы гарантировать выполнение требований, либо отказать приложению, заставляя его пересмотреть свои требования к качеству обслуживания, либо прекратить процедуру резервирования, отложив сеанс связи. Процедура резервирования ресурсов позволяет приложениям заранее определить, есть ли в сети возможность доставить многоадресный трафик всем адресатам в полном объеме, и, возможно, принять решение о доставке отдельным получателям усеченных версий потоков. Иначе говоря, RSVP — это протокол сигнализации, который обеспечивает резервирование ресурсов, управляет ими с целью предоставления интегрированных услуг и эмулирует тем самым выделенные каналы в IP-сетях. Протокол RSVP позволяет запрашивать, например, гарантированную пропускную способность, предсказуемую задержку, максимальный уровень потерь. Но само резервирование, разумеется, выполняется только в том случае, если в наличии имеются требующиеся для этого ресурсы.

Рассмотрим процедуру резервирования ресурсов по протоколу RSVP (рис. 4.2). Отправитель данных передает по уникальному или по групповому адресу получателя (или получателей) специальное сообщение *Path,* в котором он указывает параметры, рекомендуемые для обслуживания трафика с нужным качеством: верхнюю и ниж-

нюю границу полосы пропускания, задержку и вариацию задержки. Сообщение *Path* передается маршрутизаторами сети по направлению к получателю либо с использованием таблиц маршрутизации в узлах сети, либо по явно заданному маршруту (эта опция доступна только при реализации RSVP-TE).

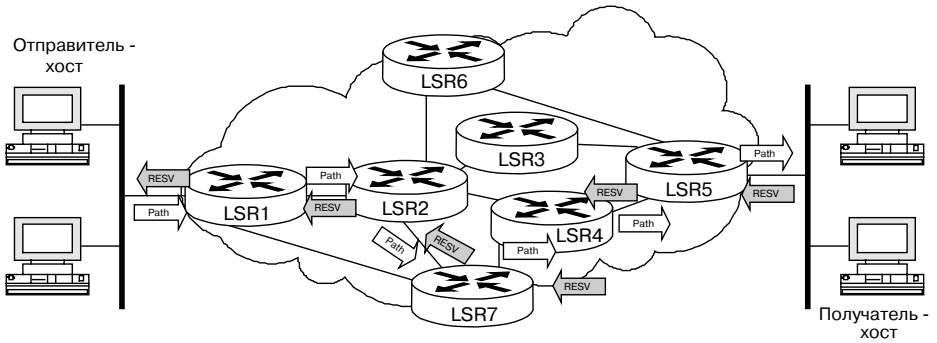


Рис. 4.2. Резервирование ресурсов при помощи протокола RSVP

Каждый маршрутизатор, поддерживающий RSVP, получив сообщение Path, фиксирует «состояние процесса создания маршрута», которое включает в себя адрес предыдущего маршрутизатора. В сети образуется фиксированный маршрут передачи сообщений в рамках сеанса RSVP.

Сообщение Path должно нести в себе шаблон отправителя (Sender Template), который описывает тип передаваемых отправителем данных. Этот шаблон представляет собой спецификацию фильтра, который может использоваться для того, чтобы отделять пакеты определенного отправителя от других пакетов в пределах сеанса. Кроме того, сообщения Path должны содержать спецификацию потока данных отправителя Tspec, определяющую характеристики формируемого отправителем трафика. Спецификация Tspec используется для предотвращения избыточного резервирования.

Шаблоны отправителя имеют тот же формат, что и спецификации фильтра, которые используются в сообщениях Resv. Следовательно, шаблон отправителя может специфицировать только его IP-адрес и (как опция) UDP/TCP порт с учетом идентификатора протокола, заданного для сеанса.

Получив сообщение Path, получатель передает маршрутизатору, от которого это сообщение получено, запрос резервирования ресурсов Resv (Reservation Request). В дополнение к информации Tspec, сообщение Resv содержит спецификацию запроса Rspec, в которой указываются нужные получателю параметры качества обслуживания, и спецификацию фильтра filterspec, определяющую, к каким пакетам сеанса относится это сообщение.

Вместе Rspec и filterspec представляют собой дескриптор потока, который маршрутизатор использует для идентификации каждой новой процедуры резервирования ресурсов.

Получив сообщение *Resv,* каждый поддерживающий RSVP маршрутизатор, находящийся на маршруте сообщений резервирования, с целью определить приемлемость указанных в запросе параметров резервирования выполняет две процедуры. С помощью *механизма управления доступом (Admission Control)* проверяется, имеются ли у маршрутизатора ресурсы, необходимые для поддержки запрашиваемого качества обслуживания, а с помощью *процедуры контроля прав (Policy Control)* — имеет ли пользователь право на резервирование ресурсов. Если запрос не может быть удовлетворен, маршрутизатор передает его отправителю сообщение об ошибке.

Если же запрос принимается, маршрутизатор передает сообщение *Resv* следующему маршрутизатору, а данные о требуемом качестве обслуживания передаются механизмам маршрутизатора, ответственным за управление трафиком. Сообщение *Resv*, переадресованное следующему узлу, несет в себе спецификацию *flowspec*, которая определяет желательное QoS и включает в себя два набора параметров:

- набор Rspec, определяющий желательные характеристики QoS,
- набор Tspec, описывающий информационный поток.

Прием маршрутизатором запроса резервирования означает также передачу им характеристик QoS на обработку в соответствующие собственные функциональные блоки, так как форматы и содержимое *Tspec* и *Rspec* установлены общими моделями обслуживания, определенными в RFC 2210 «The Use of RSVP with IETF Integrated Services». Способ обработки маршрутизатором параметров QoS в спецификациях протокола RSVP не описывается.

Когда последний маршрутизатор получает сообщение Resv и принимает запрос, он передает подтверждающие сообщение обратно к узлу-получателю (последний маршрутизатор располагается ближе всего к инициатору процедуры резервирования, а в случае групповых потоков — в точке слияния резервируемых трактов). После выполнения резервирования инициатор вышеописанной процедуры начинает передавать свой трафик, который на всем пути к получателю обслуживаются с заданным качеством.

Отменить резервирование можно двумя путями: прямо или косвенно. В первом случае отмена производится по инициативе источника или приемника с помощью специальных сообщений RSVP. Во втором случае — по таймеру, т.к. резервирование имеет определенный срок действия.

В рамках этой книги авторы решили не останавливаться более подробно на базовой версии протокола RSVP, так как информация об ее процедурах и форматах имеется, в том числе, и на русском языке, и каждый интересующийся этим протоколом может, при необходимости, получить нужные ему сведения. В то же время, протокол RSVP-TE отнюдь не столь подробно освещен в литературе, и поэтому в книге ему уделяется несколько больше внимания. Но, прежде всего, здесь рассматривается роль обоих вариантов протокола именно в сетях MPLS.

4.3. Роль RSVP и RSVP-TE в MPLS

Первая цель введения в сеть MPLS функций поддержки протокола RSVP состоит в том, чтобы LSR, которые классифицируют пакеты, анализируя их метки, а не IP-заголовки, могли распознавать пакеты, принадлежащие тем потокам, для которых было сделано резервирование ресурсов. Другими словами, нам нужно создавать привязку меток к FEC для потоков, которые обеспечены резервированными ресурсами с помощью протокола RSVP. Мы можем рассматривать совокупность пакетов, для которых было выполнено резервирование по протоколу RSVP, как совокупность пакетов, принадлежащих некоторому новому классу FEC.

В протоколе RSVP довольно просто произвести привязку меток к потокам с резервированием, по крайней мере, в случае использования уникратных адресов. В расширенной версии протокола, описанной в RFC 3209 «Extensions to RSVP for LSP Tunnels» и получившей название RSVP-TE, определен новый *объект LABEL*, который переносится в сообщении *Resv*. Таким образом, RSVP становится инструментом для распределения меток MPLS. Когда маршрутизатору LSR нужно передать сообщение *Resv* для нового потока, он выбирает из своего пула свободную метку, создает запись в своей таблице LFIB, определяя выбранную метку как входящую, и затем передает сообщение *Resv*, содержащее эту метку в объекте LABEL. Следует отметить, что поскольку сообщения *Resv* идут от получателя к отправителю, это — разновидность распределения меток снизу.

При получении сообщения *Resv* с объектом LABEL, содержащим эту входящую метку, вышестоящий LSR вносит ее в свою таблицу как исходящую для пересылки к нижестоящему LSR, передавшему сообщение *Resv*. Затем он назначает новую метку, которая будет использоваться им как входящая метка, и вставляет эту новую метку в сообщение *Resv*, передаваемое следующему вышестоящему LSR. Понятно, что по мере прохождения сообщений *Resv* от нижестоящих маршрутизаторов к вышестоящим создается тракт LSP

вдоль всего пути следования этих сообщений. Следует также отметить, что поскольку в сообщениях *Resv* указываются метки, каждый LSR может легко связать с трактом LSP соответствующие ресурсы, необходимые для поддержки QoS.

Протокол RSVP, расширенный объектом LABEL, поддерживает пересылку пакетов по сети MPLS (тракт LSP) только вдоль маршрута, вычисленного схемой традиционной маршрутизации пакетов IP. Это обусловлено тем, что при использовании обычного протокола RSVP маршрут, по которому идет сообщение *Path*, управляется парадигмой пересылки на основе пункта назначения, и задает маршрут для LSP. Чтобы определить, к какому ближайшему маршрутизатору нужно переслать сообщение *Path*, маршрутизатор, который должен его передать, использует имеющуюся у него таблицу маршрутизации, формируемую с помощью рассматриваемых в трех следующих главах протоколов IS-IS, OSPF, RIP или BGP, и адрес получателя, содержащийся в заголовке IP-пакета. При этом отсутствует возможность «управлять» сообщением *Path*, направляя его по определенному, явно заданному маршруту.

Для того чтобы это стало возможно, протокол RSVP расширяется новым объектом — *Explicit Route Object (ERO)*. Объект переносится в сообщении *Path* и содержит явно заданный маршрут, по которому должно идти сообщение. Пересылка такого сообщения маршрутизатором определяется не адресом получателя, содержащимся в заголовке IP пакета, а содержанием объекта ERO. Эта функция позволяет автоматически (или в результате действий администратора) ремаршрутизировать LSP в обход аварийных участков, перегруженных областей и узких мест сети. Использование явно заданных маршрутов помогает решать задачи управления трафиком, к которым мы еще вернемся в главе 9, целиком посвященной *Traffic Engineering*.

Объект ERO содержит описание упорядоченной последовательности сетевых сегментов («пересылок»), которая задает явный маршрут. Каждый такой сетевой сегмент представляется *абстрактным узлом* (*abstract node*), идентифицирующим группу из одного или нескольким маршрутизаторов, внутренняя топология которой не прозрачна для маршрутизатора, вычисляющего явный маршрут. Одним из примеров абстрактного узла является совокупность маршрутизаторов, которые принадлежат определенной автономной системе. Другой пример абстрактного узла — совокупность маршрутизаторов, адреса которых имеют определенный адресный префикс. С каждым абстрактным узлом связан индикатор «*strict/loose*», который позволяет специфицировать явно заданный маршрут как смесь *строго (strict)* и *не строго (loose) определенных участков*. Не строго определенные участки могут использоваться, например, когда маршрутизатор, который вычисляет явный маршрут, не имеет

достаточной информации о топологии для расчета маршрута в виде только строго определенных участков. Такая ситуация может иметь место, когда, например, маршрут должен пройти через несколько областей OSPF или IS-IS.

С точки зрения кодирования, объект ERO представляет собой последовательность обсуждавшихся в параграфе 3.3.2 *конструкций TLV (тип, длина, значение)*, в которой каждый триплет описывает определенный абстрактный узел. Для того чтобы показать, как используется протокол RSVP с целью обеспечить пересылку пакетов по явно заданному маршруту в сети MPLS, рассмотрим более пристально пример, представленный на рис. 4.2.

Предположим, что узлу LSR1 нужно создать условия для пересылки пакетов по явно заданному маршруту в сети MPLS (LSR7, LSR4, LSR5). Для того чтобы это сделать, LSR1 формирует объект ERO, описывающий последовательность трех абстрактных узлов — LSR7, LSR4 и LSR5. Каждый абстрактный узел представлен адресным префиксом IP длиной 32 бита (это не что иное, как обычный IP-адрес). Отметим, что такой адрес может описывать либо весь LSR, либо один определенный интерфейс LSR. Затем LSR1 формирует сообщение *Path* и включает в это сообщение объект ERO, а также новые объекты — LSP_TUNNEL, о котором речь пойдет ниже, и LABEL_REQUEST, который не только инициирует процесс присвоения метки, но и несет информацию о протоколе сетевого уровня, использующем этот тракт. Когда сообщение сформировано, LSR1 анализирует данные о первом абстрактном узле, указанном в ERO (т.е. об LSR7), находит звено, соединяющее LSR1 и LSR7, и передает сообщение *Path* по этому звену. Когда LSR7 получает сообщение *Path*, он обнаруживает, что является первым из абстрактных узлов, указанных в объекте ERO. Затем LSR7 анализирует данные о следующем абстрактном узле LSR4 и находит звено, соединяющее его с этим узлом. После этого LSR7 модифицирует объект ERO, удаляя из него описание абстрактного узла, связанного с LSR7, и передает к LSR4 сообщение *Path*, содержащее модифицированный ERO и объект LABEL_REQUEST. Обработка сообщения *Path* маршрутизатором LSR4 аналогична обработке сообщения маршрутизатором LSR7. Теперь ERO в сообщении *Path*, которое LSR4 передает к LSR5, содержит описание только узла LSR5. Когда сообщение *Path* поступает к LSR5, этот маршрутизатор выясняет, что он является последним из указанных в ERO абстрактных узлов. Маршрутизатор LSR5 определяет по описанным параметрам трафика, какую пропускную способность он должен резервировать, и выделяет необходимые ресурсы. Затем он выбирает метку для нового LSP и, поместив ее в объект LABEL, создает сообщение *Resv,* которое передает к LSR4. Когда LSR4 получает это сообщение, он вносит значение метки, полученной им в объекте LABEL, в свою таблицу пересылки. Вслед

за этим LSR4 передает сообщение *Resv* к LSR7. Это сообщение содержит модифицированный объект LABEL. Обработка сообщения маршрутизатором LSR7 аналогична обработке сообщения маршрутизатором LSR4. Наконец, LSR7 передает сообщение *Resv* к LSR1. Когда сообщение поступило к LSR1, мы говорим, что создан LSP для явно заданного маршрута. Следует обратить внимание, что инициатором процедуры назначения метки был входной LSR, то есть верхний маршрутизатор, который передал запрос LABEL_REQUEST. Следовательно, налицо режим распределения меток *downstream on demand.*

4.4. Расширение RSVP-TE

Поскольку трафик, проходящий по LSP, определяется меткой, присвоенной во входном маршрутизаторе, то с точки зрения протокола IP LSP можно считать своеобразным туннелем, находящимся под уровнем IP-маршрутизации.

Для поддержки функций такого туннеля (т.е. уже привычного как LSP) в RSVP-TE специфицированы новые *C-типы* для *объектов SESSION, SENDER_TEMPLATE* и *FILTER_SPEC.* Эти новые типы называются *LSP_TUNNEL_IPv4* и *LSP_TUNNEL_IPv6*. Так как объекты обоих типов функционально идентичны и различаются только форматом адреса, с которым они работают, в дальнейшем эти типы именуются просто *LSP_TUNNEL_IP.*

В некоторых приложениях, например, при операциях ремаршрутизации, вызванных обнаружением лучшего маршрута или неисправностями в сети, а также при распределении трафика по нескольким маршрутам, может оказаться полезным создавать ассоциированный набор LSP. В приложениях управления трафиком такие наборы называются *Traffic Engineered Tunnels* или *TE-туннели*. Авторы вынуждены использовать этот термин в английском написании, так как русский перевод выглядит довольно громоздко — туннель, образованный с помощью средств управления трафиком. Для идентификации TE-туннелей используются 2 идентификатора.

Идентификатор туннеля (tunnel ID) является частью объекта SESSION и однозначно определяет TE-туннель. Объекты SENDER_TEMPLATE и FILTER_SPEC несут *идентификатор* (*LSP_ID*) и, в паре с объектом SESSION, любой из них может однозначно идентифицировать LSP. Прежде чем перейти к детальному рассмотрению объектов и процедур, целесообразно назвать еще несколько нововведений в протоколе RSVP-TE.

Чтобы отправитель мог получать информацию о маршруте, по которому проходит LSP, в сообщение *Path* может включаться *объект RECORD_ROUTE*. Отправитель может также использовать этот объект для того, чтобы запросить получение уведомлений

об изменениях маршрута LSP. Объект RECORD_ROUTE практически аналогичен вектору пути (Path Vector) и поэтому может использоваться для обнаружения закольцованных маршрутов.

В сообщение *Path* может быть также включен *объект SESSION_ATTRIBUTE*, который содержит информацию, относящуюся к приоритету данного сеанса создания LSP по отношению к другим текущим сеансам, к защите в системе коммутации, обеспечивающей создание нового маршрута в обход отказавших звеньев, а также к категории срочности данного сеанса. Этот объект может также указывать, должны ли записываться значения меток при записи информации о маршрутах.

4.5. Ремаршрутизация TE-туннелей

Одним из основных требований к управлению трафиком (Traffic Engineering) является возможность изменять маршрут, т.е. *ремаршрутизировать* TE-туннели согласно некоторым условиям, основанным на административной политике. В общем случае весьма нежелательно перераспределять трафик или оказывать неблагоприятное воздействие на сеть в процессе ремаршрутизации TE-туннелей. Это соображение привело к применению *механизма make-before-break (создать новый тракт до уничтожения старого)*, т.е. механизма, при котором старый тракт продолжает использоваться в то время, пока создается новый тракт, а после того как он создан, LSR, производящий ремаршрутизацию, выполняет переключение на новый тракт и лишь затем разрушает старый. Эта базовая методика может применяться во избежание двойного резервирования ресурсов как в протоколе CR-LDP (путем использования значения *modify* флага в сообщении Label Request, как это упоминалось в главе 3), так и в протоколе RSVP (путем использования фильтров при стиле резервирования Shared-Explicit, рассмотренном в первом параграфе этой главы).

Кроме того, может возникнуть соперничество старого и нового LSP по поводу ресурсов сети, в результате чего возможен отказ в создании нового LSP. Схожая ситуация может возникнуть при попытке увеличить пропускную способность TE-туннеля. Новое резервирование будет выполняться для всей требуемой пропускной способности и может быть отвергнуто, когда на самом деле необходимо дополнительно резервировать лишь разницу между требующимися и уже существующими резервными ресурсами.

Расширение RSVP-TE помогает обойти эти проблемы. Комбинация объекта LSP_TUNNEL SESSION и рассмотренного в начале главы стиля резервирования SE позволяет плавно увеличивать пропускную способность и производить динамическую маршрутизацию.

Идея состоит в том, чтобы старый и новый LSP разделяли между собой ресурсы, являющиеся для них общими. Объект LSP_TUNNEL SESSION используется для того, чтобы сузить область RSVP-сеанса до конкретного TE-туннеля, идентифицируемого комбинацией IP-адреса назначения, идентификатора туннеля *Tunnel ID* и IP-адреса узла, являющегося для туннеля входным.

При ремаршрутизации или при увеличении пропускной способности входной узел туннеля выступает в RSVP-сеансе в роли двух разных отправителей, что достигается добавлением еще одного идентификатора LSP *(LSP ID),* переносимого в объектах SENDER_TEMPLATE и FILTER_SPEC.

Чтобы произвести ремаршрутизацию, узел берет новый идентификатор LSP ID и формирует новый объект SENDER_TEMPLATE. Затем он создает объект ERO, чтобы задать новый маршрут, и передает сообщение *Path*, содержащее исходный объект SESSION, в котором поле *Extended_Tunnel_ID* заполнено его IP-адресом, и новые объекты SENDER_TEMPLATE и ERO. Узел продолжает использовать старый LSP и обновлять старые сообщения Path. Для звеньев, которые в старом и новом LSP совпадают, применяется стиль резервирование SE, позволяющий им совместно использовать ресурс, а на новом участке сообщение *Path* обрабатывается как при создании нового LSP. Как только входной узел получит сообщении *Resv* по новому LSP, он сможет перенести на него весь трафик и разрушить старый LSP, передав сообщение *PathTear*.

При попытке увеличить пропускную способность используется новое сообщение *Path* с новым LSP ID, причем одновременно с обновлением текущего LSP ID, чтобы резервирование не было утрачено в случае, если попытка увеличить резервный ресурс окажется безуспешной.

Стандартный RSVP и IntServ, согласно RFC 2210, «The Use of RSVP with IETF Integrated Services» задают отправителю RSVP наименьшее значение *MTU (Maximum Transfer Unit)* из всех возможных MTU на пути от этого отправителя к получателю. Таким же образом определяется MTU для LSP, организуемого протоколом RSVP-TE. Информация *Path MTU* переносится в объектах либо *Integrated Services*, либо *Null Service*. Когда используются объекты Integrated Services, *Path MTU* определяется на базе процедур, описанных в RFC 2210. Определение Path MTU при использовании объектов Null Service описано в RFC 1191 «Path MTU Discovery». То, что LSR-отправитель знает допустимый размер MTU, позволяет ему выявлять IP-пакеты с длиной, превышающей это значение, которые подлежат фрагментации (если она разрешена) или стиранию (если запрещена), с последующим извещением об этом отправителя по протоколу ICMP.

4.6. Форматы RSVP-TE

4.6.1. Сообщения создания LSP

В спецификации RSVP-TE определено пять новых объектов: LABEL_REQUEST, LABEL, EXPLICIT_ROUTE, RECORD ROUTE, SESSION ATTRIBUTE. Добавлены также новые C-типы для объектов SESSION, SENDER_TEMPLATE и FILTER_SPEC. Все новые объекты не обязательны для RSVP, но объекты LABEL_REQUEST и LABEL обязательны для RSVP-TE.

Форматы сообщений Path и Resv с этими новыми объектами имеют следующий вид.

Сообщение Path

```
<Path Message> ::=   <Common Header> [ <INTEGRITY> ]
                              <SESSION> <RSVP_HOP>
                              <TIME_VALUES>
                              [ <EXPLICIT_ROUTE> ]
                              <LABEL_REQUEST>
                              [ <SESSION_ATTRIBUTE> ]
                              [ <POLICY_DATA> ... ]
                              <sender descriptor>
```

Формат sender descriptor (описателя отправителя):

```
<sender descriptor> ::=  <SENDER_TEMPLATE> <SENDER_TSPEC>
                              [ <ADSPEC> ]
                              [ <RECORD_ROUTE> ]
```

Сообщение Resv

```
<Resv Message> ::=       <Common Header> [ <INTEGRITY> ]
                         <SESSION>  <RSVP_HOP>
                         <TIME_VALUES>
                         [ <RESV_CONFIRM> ]  [ <SCOPE> ]
                         [ <POLICY_DATA> ... ]
                         <STYLE> <flow descriptor list>
```

Формат flow descriptor list:

```
<flow descriptor list> ::= <FF flow descriptor list>
                                    | <SE flow descriptor>
```

Формат FF flow descriptor list:

```
<FF flow descriptor list> ::= <FLOWSPEC> <FILTER_SPEC>
                                      <LABEL> [ <RECORD_ROUTE> ]
                                    | <FF flow descriptor list>
                                      <FF flow descriptor>
```

Формат FF flow descriptor:

```
<FF flow descriptor> ::= [ <FLOWSPEC> ] <FILTER_SPEC> <LABEL>
                                     [ <RECORD_ROUTE> ]
```

Формат SE flow descriptor:

```
<SE flow descriptor> ::= <FLOWSPEC> <SE filter spec list>
```

Формат SE filter spec list:

```
<SE filter spec list> ::= <SE filter spec>
                                  | <SE filter spec list> <SE filter spec>
```

Формат SE filter spec:

```
<SE filter spec> ::=     <FILTER_SPEC> <LABEL> [ <RECORD_ROUTE> ]
```

4.6.2. Объект LABEL

Объект LABEL передается в сообщении *Resv* сразу после объекта FILTER_SPEC для этого отправителя. LABEL класс = 16, C-тип = 1. Метку передает нижний LSR, причем если в ее запросе указан диапазон значений меток, то она должна быть выбрана в этом диапазоне. Если у узла нет подходящей метки, то в ответ на запрос он передает сообщение *PathErr*, в котором указывается, что назначить метку невозможно.

Если узел получает сообщение *Resv,* в котором нескольким отправителям присвоена одна метка, он также может присвоить им одну метку, но если ему необходимо управлять определенными пользователями, он должен присвоить им индивидуальные метки. В случае с сетью ATM следует также учитывать, что некоторые ATM-коммутаторы не могут объединять потоки, о чем они сообщают в объекте LABEL_REQUEST C-тип 2, присваивая M-биту значение 0.

Рис. 4.3. Формат объекта LABEL

Метка может оказаться неприемлемой для верхнего LSR при следующих условиях:

- Узел является ATM-коммутатором, не поддерживающим объединение потоков, а нижний LSR назначил двум отправителям одинаковые метки.
- Была присвоена неявная нулевая метка (Implicit NULL label), но узел не в состоянии определить ассоциированный с ней идентификатор сетевого уровня.
- Присвоенная метка находится вне диапазона значений, указанного в запросе.

В любом из этих случаев верхний LSR передает сообщение *PathErr* с указанием, что значение полученной метки неприемлемо.

4.6.3. Объект LABEL_REQUEST

Класс объекта LABEL_REQUEST = 19. При этом возможны три C-типа:

- тип 1 — запрос метки без указания диапазона ее значений,
- тип 2 — запрос метки с заданным диапазоном ее значений в случае с ATM,
- тип 3 — запрос метки с заданным диапазоном ее значений в случае с Frame Relay.

На рис. 4.4 представлен формат C-тип = 1. Резервным битам присваивается значение 0 на передаче, а при приеме они игнорируются.

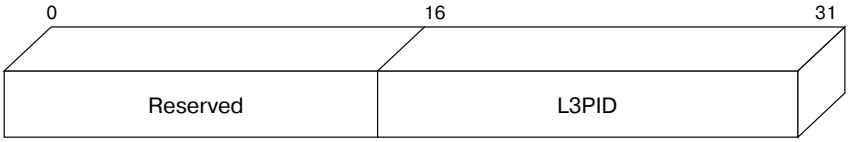


Рис. 4.4. Формат объекта LABEL REQUEST без указания диапазона значений метки

L3PID — идентификатор протокола третьего уровня, использующего данный путь. Используются стандартные значения EtherType.

На рис. 4.5 представлен формат объекта, C-тип = 2.

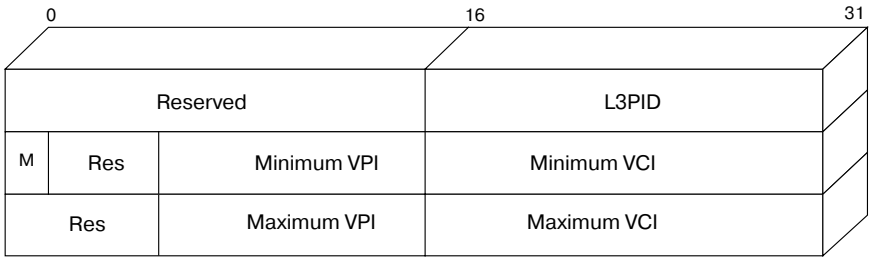


Рис. 4.5. Формат объекта LABEL REQUEST с диапазоном меток, ATM

M — значение этого бита, равное единице, свидетельствует о том, что узел способен объединять потоки.

Minimum VPI — 12-битовое поле, которое специфицирует нижнюю границу блока VPI, поддерживаемого коммутатором-отправителем.

Minimum VCI — 16-битовое поле, которое специфицирует нижнюю границу блока VCI, поддерживаемого коммутатором-отправителем.

Maximum VPI — 12-битовое поле, которое специфицирует верхнюю границу блока VPI, поддерживаемого коммутатором-отправителем.

Maximum VCI — 16-битовое поле, которое специфицирует верхнюю границу блока VCI, поддерживаемого коммутатором-отправителем.

В случае, если значение в поле для VPI меньше 12 битов или значение в поле для VCI меньше 16 битов, то они должны быть выровнены по правому краю, а оставшиеся разряды заполнены нулями.

На рис. 4.6 представлен формат объекта, C-тип = 3.

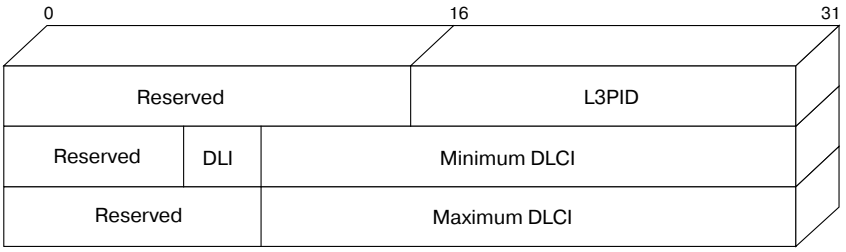


Рис. 4.6. Формат объекта LABEL REQUEST с диапазоном меток FR

DLI — DLCI Length Indicator. Определяет число битов в DLCI. Поддерживаются следующие значения: для поля DLI, соответствующего 00, число битов DLCI равно 10, а для поля DLI, соответствующего 10, число битов DLCI равно 23.

Minimum DLCI — 23-битовое поле, которое определяет нижнюю границу блока DLCI, поддерживаемого отправителем.

Maximum DLCI — 23-битовое поле, которое определяет верхнюю границу блока DLCI, поддерживаемого отправителем.

Объект LABEL_REQUEST передается в сообщении *Path* и является индикатором запроса привязки метки. Он переносит также информацию о протоколе сетевого уровня, чтобы сообщения протоколов не-IP могли передаваться по LSP. Эта информация может также быть полезной при присвоении меток, потому что некоторые резервированные метки являются протокольно-зависимыми. Если конечный получатель не поддерживает протокол, указанный в L3PID, он передает сообщение *PathErr* с кодом ошибки «не поддерживаемый L3PID».

Если маршрутизатор не поддерживает объект LABEL_REQUEST или соответствующий ему C-тип, то он отвечает сообщением *PathErr* с информацией «неизвестный класс объекта» или «неизвестный C-тип», соответственно. Если маршрутизатору известно, что следующий узел не поддерживает RSVP, он передает отправителю запроса сообщение *PathErr* с уведомлением о том, что при работе MPLS обнаружен маршрутизатор, не поддерживающий RSVP.

4.6.4. Объект EXPLICIT_ROUTE (ERO)

Этот объект имеет Класс=20 и, на данный момент, — только один C-тип = 1 (рис. 4.7). EXPLICIT_ROUTE содержит описание последовательности узлов, через которые должен будет пройти создаваемый LSP. Существует также возможность задавать вместо любого узла группу узлов, что предоставляет системе маршрутизации некоторую гибкость и позволяет иметь при формировании ERO неполную информацию о маршруте. Таким образом, ERO — это последовательность *абстрактных узлов*, определение которых было дано в начале главы.

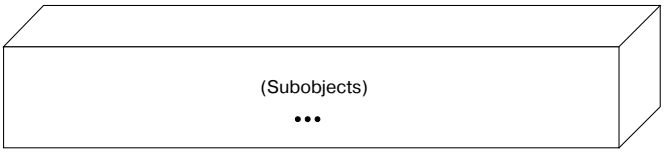


Рис. 4.7. Формат объекта ERO

Subobjects — информационные элементы переменной длины, составляющие содержание ERO (рис. 4.8). В сообщении *Path*, содержащем несколько *Subobjects*, только первый является значимым, а остальные могут игнорироваться и не передаваться дальше.

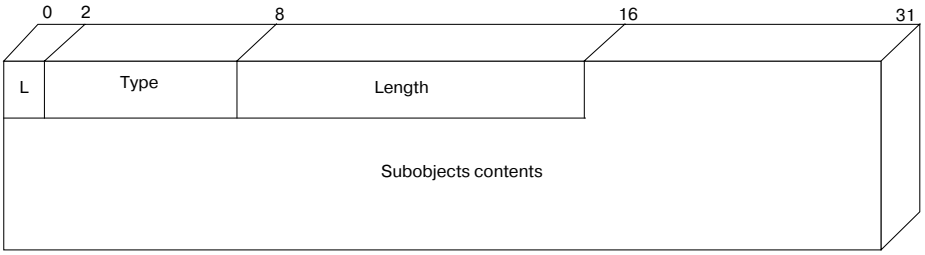


Рис. 4.8. Общий формат Subobject

Если бит *L* имеет значение 1, то в Subobject содержится информация о не строгом (loose) маршруте, если же бит имеет нулевое значение, то следующая пересылка в Subobject определяется строго (strict). Разница между строгим и не строгим описанием маршрута состоит в том, что если пересылка задана строго, то сообщение от отправителя к заданному узлу не должно проходить через промежуточные маршрутизаторы, а в случае, если пересылка задана не строго, оно может идти через некоторые промежуточные узлы сети.

Length — это поле содержит общую длину Subobject в байтах, включая заголовок. Минимальное значение длины — 4, другие возможные значения должны быть кратны четырем.

Type — это поле определяет тип содержимого *Subobject*. На данный момент определены три следующих значения:

Поле Type	Содержимое Subobject
0000001	IPv4 префикс (1)
0000010	IPv6 префикс (2)
0100000	Номер автономной системы (32)

Формат Subobject IPv4 Prefix представлен на рис. 4.9. Абстрактный узел, определяемый этим Subobject, — это сетевые узлы, IP-адрес которых лежит в пределах указанного префикса.

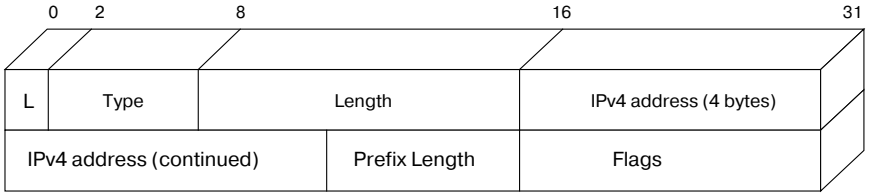


Рис. 4.9. Формат Subobject IPv4 Prefix

IPv4 address — содержит IPv4-адрес, воспринимаемый как префикс, в соответствии с полем *Prefix Length*. Биты, следующие за префиксом, игнорируются на приеме, и на передаче им следует присваивать значение 0.

Prefix Length — Длина IPv4 префикса в битах.

Padding — заполнение нулями. Игнорируется при приеме.

Для Subobject IPv6 Prefix используется следующий формат (рис. 4.10). И сам формат IPv6 Prefix, и назначение полей аналогичны IPv4 Prefix, различие состоит только в размере полей, что обусловленно длиной IPv6 адреса.

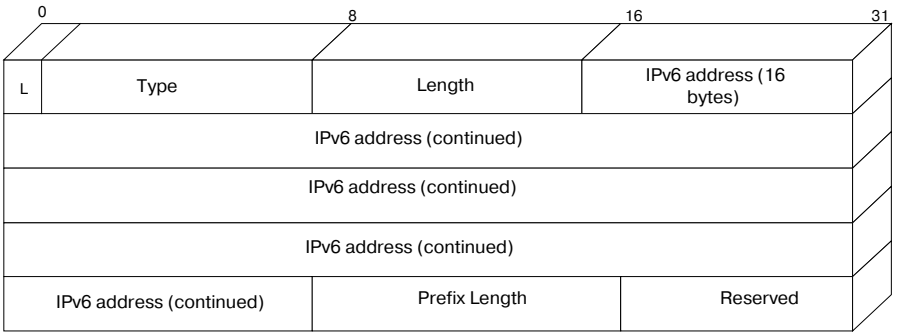


Рис. 4.10. Формат Subobject IPv6 Prefix

И, наконец, Subobject Autonomous System Number. Содержимое объекта — 2-байтовый номер автономной системы AS. Абстрактный узел для этого Subobject представляет собой все узлы, принадлежащие указанной AS.

4.6.5. Объект RECORD_ROUTE (RRO)

Класс этого объекта = 21, и на данный момент определен только один C-тип = 1. Объект RRO может присутствовать в сообщениях *Path* и *Resv*. Если в сообщении *Path* содержится несколько RRO, то значимым является только первый, а остальные игнорируются и не передаются дальше.

Так же, как ERO на рис. 4.7, RRO состоит из Subobjects. Каждый из них имеет длину, кратную 4 байтам, минимально возмож-

ная длина = 4. Subobjects организованы по принципу стека LIFO (last-in-first-out). На текущий момент определено 3 типа Subobjects.

Поле Type	Содержание Subobject
00000001	IPv4 адрес
00000010	IPv6 адрес
00000011	Метка

Формат Subobject IPv4 address представлен на рис. 4.11.

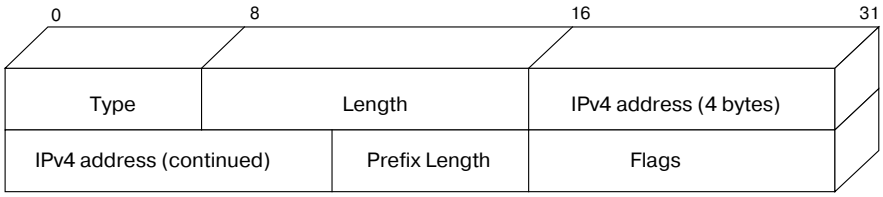


Рис. 4.11. Формат Subobject IPv4 address

Length — значение поля «длина» всегда равно 8.

IPv4 address — содержит 32-битовый индивидуальный IP-адрес.

Prefix Length —значение поля всегда равно 32.

Flags — флаги, принимающие одно из двух значений:

Поле Flags	Значение флага
00000001	*Доступна местная защита* — означает, что нижнее относительно данного узла звено защищено локальными средствами восстановления (этот флаг может быть установлен только в том случае, если в объекте SESSION_ATTRIBUTE соответствующего сообщения *Path* был установлен флаг местной защиты)
00000010	*Местная защита активизирована* — означает, что локальные средства восстановления для поддержки данного тракта (туннеля) работоспособны

Формат Subobject IPv6 address представлен на рис.4.12. Он аналогичен формату Subobject IPv4 address и отличается от него только размером поля адреса и тем, что *Length* = 20, *Prefix Length* = 128.

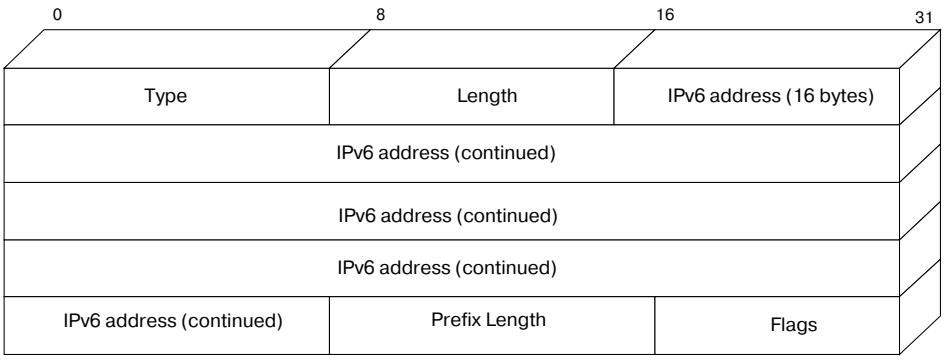


Рис. 4.12. Формат Subobject IPv6 address

Формат Subobject Label представлен на рис. 4.13.

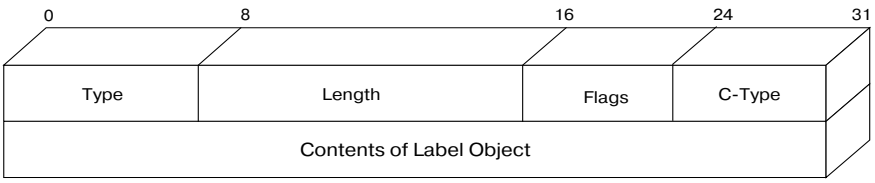


Рис. 4.13. Формат Subobject Label

Length — указывает общую длину Subobject в байтах.

Flags — 00000001 = Global Label; это означает, что метка будет понятна при ее приеме через любой интерфейс.

C-Type копируется из объекта LABEL.

Contents of Label Object копируется из объекта LABEL.

Возможны три варианта применения объекта RECORD_ROUTE. Во-первых, RRO может использоваться для обнаружения на сетевом уровне зацикливания маршрутов. Во-вторых, он накапливает детальную информацию о маршруте RSVP-сеанса и доставляет затем эти сведения отправителю или получателю, оповещая их обо всех изменениях маршрута. Наконец, синтаксически RRO спроектирован так, что его можно, с минимальными изменениями, поместить в объект EXPLICIT_ROUTE. Это может быть полезно, если отправитель получает RRO от получателя в сообщении *Resv* и включает его в ERO, передаваемый в следующем сообщении *Path*, чтобы точно установить маршрут сеанса.

Если маршрутизатор принимает сообщение, содержащее RRO, то при дальнейшей пересылке этого сообщения он должен включить в RRO новый Subobject, поместив в него свой IP-адрес. Если в объекте SESSION_ATTRIBUTE установлен флаг *Label_Recording* (запись меток), то маршрутизатору следует добавить еще и Subobject Label. Если добавленный Subobject сделает RRO слишком большим для переноса его в сообщении *Path* (или *Resv*), то RRO следует исключить из сообщения и продолжить его обработку как обычно. Отправителя (получателя) же следует известить об этом сообщением *PathErr* (*ResvErr*) с тем, чтобы он исключил RRO из своих дальнейших сообщений.

Для обнаружения закольцованных маршрутов промежуточный маршрутизатор просматривает все Subobjects в RRO, и, увидев в списке себя, делает вывод о существовании петли.

Обнаруженные петли делятся на два общих класса. К первому относятся временные петли, образовавшиеся в результате работы систем маршрутизации на сетевом уровне. Ко второму относят-

ся постоянные закольцованные маршруты, возникающие обычно из-за неправильной конфигурации сети.

Действия маршрутизатора при обнаружении петель зависят от того, в каком сообщении был получен RRO, извещающий о маршрутной петле. Если это было сообщение *Path*, узел формирует сообщение *PathErr*, поместив в него извещение о наличии петли, и прекращает дальнейшую пересылку *Path*. Обнаружив петлю при получении сообщения *Resv*, маршрутизатор просто отбрасывает его, так как *Resv* не может зациклиться, если этого не произошло с соответствующим ему сообщением *Path.*

В дальнейшем для RRO могут быть определены новые Subobjects. Когда маршрутизатор встречает Subobject неизвестного ему типа, он пропускает его и продолжает обработку объекта.

4.6.6. Объект SESSION

Рассматриваемые здесь объекты SESSION, SENDER_TEMPLATE и FILTER_SPEC не являются новыми для протокола RSVP; в протоколе RSVP-TE в них только добавлены новые C-типы.

Объект LSP_TUNNEL_IPv4 SESSION имеет C-тип = 7 и представлен на рис. 4.14.

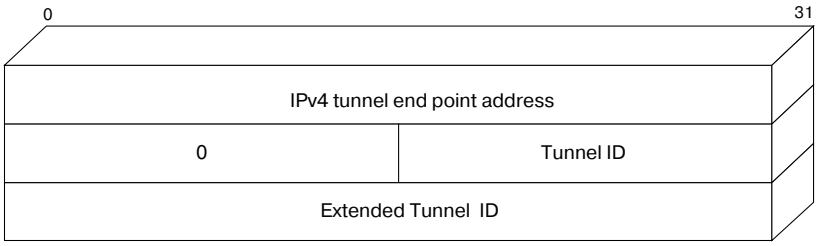


Рис. 4.14. Формат объекта LSP_TUNNEL_IPv4 SESSION

IPv4 tunnel end point address — IPv4 адрес входного узла туннеля.

Tunnel ID — 16-битовый идентификатор, используемый в объекте SESSION и остающийся постоянным в течение всего времени существования туннеля.

Extended Tunnel ID — 32-битовый идентификатор, используемый в объекте SESSION и остающийся постоянным в течение всего времени существования туннеля. Обычно это поле заполняется нулями. Входной узел, которому нужно сузить рамки SESSION до пары «входной-выходной», может поместить сюда свой IPv4 адрес, как уникальный идентификатор.

Объект LSP_TUNNEL_IPv6 SESSION имеет C-тип = 8 и представлен на рис. 4.15. Формат его аналогичен формату объекта LSP_TUNNEL_IPv4 SESSION, отличие состоит только в размере одноименных полей и обусловлено величиной IPv6-адреса.

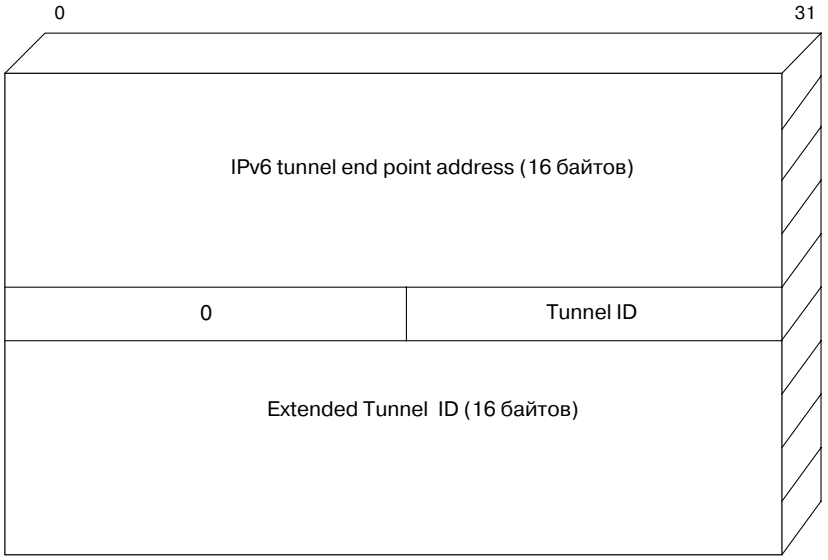


Рис. 4.15. Формат объекта LSP_TUNNEL_IPv6 SESSION

4.6.7. Объект SENDER_TEMPLATE

Объект LSP_TUNNEL_IPv4 SENDER_TEMPLATE имеет C-тип = 7 и представлен на рис. 4.16.

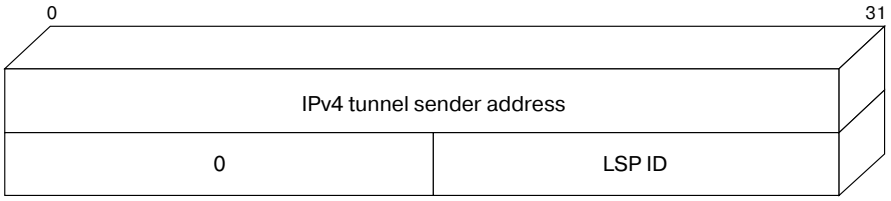


Рис. 4.16. Формат объекта LSP_TUNNEL_IPv4 SENDER_TEMPLATE

IPv4 tunnel sender address — IPv4-адрес отправителя.

LSP ID — 16-битовый идентификатор, используемый в объекте SENDER_TEMPLATE и FILTER_SPEC, который может быть изменен, чтобы позволить отправителю разделять ресурс с самим собой.

Объект LSP_TUNNEL_IPv6 SENDER_TEMPLATE имеет C-тип = 8. Формат и назначение этого объекта такие же, как объекта LSP_TUNNEL_IPv4 SENDER_TEMPLATE, различие состоит только в величине полей, что обусловлено размером IPv6-адреса и показано на рис. 4.17.

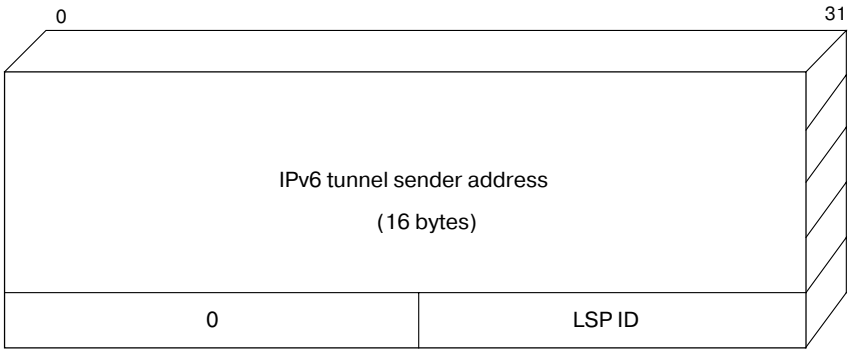


Рис. 4.17. Формат объекта LSP_TUNNEL_IPv6 SENDER_TEMPLATE

4.6.8. Объект FILTER_SPEC

Формат объектов LSP_TUNNEL_IPv4 FILTER_SPEC и LSP_TUNNEL_IPv6 FILTER_SPEC аналогичен формату объектов LSP_TUNNEL_IPv4 SENDER_TEMPLATE и LSP_TUNNEL_IPv6 SENDER_TEMPLATE, соответственно.

4.6.9. Объект SESSION_ATTRIBUTE

Объект SESSION_ATTRIBUTE Класс = 207, LSP_TUNNEL C-тип = 7, формат представлен на рис. 4.18.

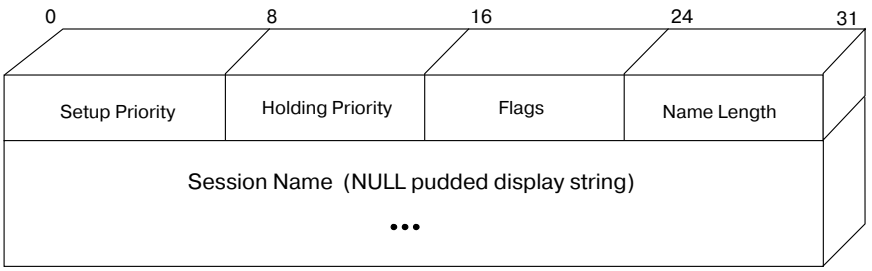


Рис. 4.18. Формат объекта SESSION_ATTRIBUTE LSP_TUNNEL

Setup Priority — приоритет в занятии ресурсов. Выбирается в диапазоне от 0 до 7, ноль — высший приоритет. Поле используется для того, чтобы принять решение, может ли данный сеанс вытеснить другой.

Holding Priority — приоритет в удержании ресурсов. Тоже выбирается в диапазоне от 0 до 7 и используется для того, чтобы принять решение, может ли данный сеанс быть вытеснен другим.

Flags — флаги.

Поле Flags	Значение флага
00000001	Желательна местная защита. Флаг разрешает транзитным маршрутизаторам использовать локальные средства восстановления, что может привести к отклонению от маршрута, заданного в ERO. Но обнаружив неисправность соседнего звена, транзитный узел может ремаршрутизировать трафик для быстрого восстановления обслуживания.
00000010	Желательна запись меток. Флаг указывает, что информация о присваиваемых метках должна вноситься в RRO.
00000100	Желателен стиль резервирование SE. Используется при ремаршрутизации или при увеличении пропускной способности туннеля. Флаг указывает, что при передаче сообщения *Resv* выходной узел должен использовать стиль резервирования SE.

Name Length — длина имени сеанса до начала заполнения нулями, в байтах.

Session Name — строка символов, кратная 4 байтам и выравниваемая с этой целью нулями. Минимальная длина = 8 байтов.

Объект SESSION_ATTRIBUTE: класс = 207, LSP_TUNNEL_RA (resource affinities) C-тип = 1, формат представлен на рис. 4.19.

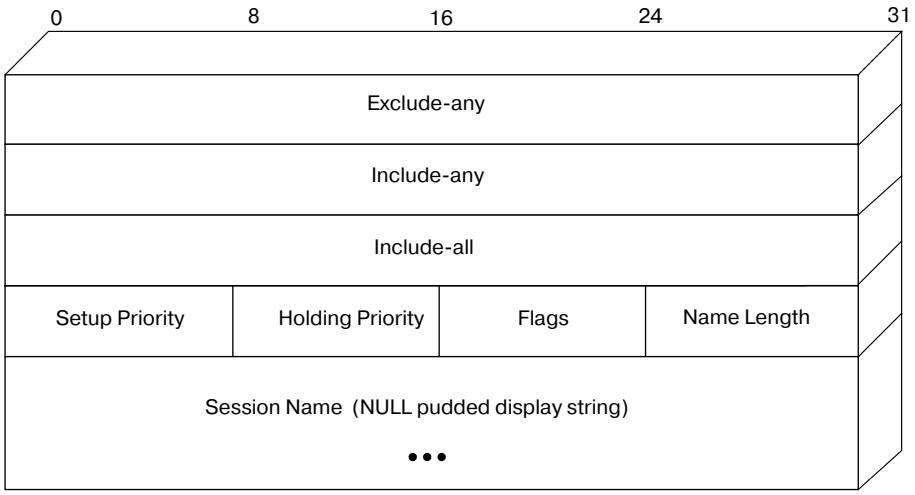


Рис. 4.19. Формат объекта SESSION_ATTRIBUTE LSP_TUNNEL_RA

Exclude-any — 32-битовый вектор, представляющий собой набор ассоциированных с туннелем атрибутов; наличие любого из них делает туннель непригодным для использования.

Include-any — 32-битовый вектор, представляющий собой набор ассоциированных с туннелем атрибутов; наличие любого из них делает туннель пригодным для использования.

Include-all — 32-битовый вектор, представляющий набор ассоциированных с туннелем атрибутов; все они должны быть в наличии для того, чтобы туннель считался пригодным для использования.

Поддержка приоритетов занятия и удержания (setup/holding) ресурсов не является обязательной. Узел может распознавать эту информацию, но быть не в состоянии выполнить требуемые операции; в таком случае ему следует переслать эту информацию дальше без изменений. Для одного и того же сеанса приоритет занятия ресурсов не должен быть выше, чем приоритет их удержания. Приоритеты занятия и удержания аналогичны приоритетам вытеснения (preemption) и защиты (defending), описанным в RFC 2751 «Signaled Preemption Priority Policy Element».

Чтобы туннель мог использоваться, он должен пройти три теста, и если хотя бы один из них окажется неуспешным, будет передано сообщение *PathErr* с кодом *policy control failure*. Когда принимается решение о возможности резервирования в строго определенном ERO узле, этот узел может проверить соответствие требуемых ресурсов *классам ресурсов* звена. При выполнении той же операции с не строго определенным узлом такая проверка обязательна.

Речь идет о следующих тестах: тест Exclude-any исключает туннель из рассмотрения, если он имеет хоть один из перечисленных в наборе атрибутов, тест Include-any допускает использование туннеля, если он имеет хоть один из перечисленных в наборе атрибутов, и тест Include-all допускает использование туннеля, если он имеет все перечисленные в наборе атрибуты. Если сообщение *Path* содержит несколько объектов SESSION_ATTRIBUTE, значащим является только первый, а остальные игнорируются и могут дальше не передаваться.

4.7. Расширение сообщения Hello

Расширения сообщений *Hello* позволяют узлам RSVP определять, когда соседний узел недоступен, т.е. неисправен. Если такая неисправность обнаружена, она обрабатывается как неисправность на уровне 2. Этот механизм используется, когда уведомления от уровня 2 недоступны, и не используются ненумерованные каналы, или когда механизмы уровня 2 не могут своевременно распознать неисправность узла. Механизм обнаружения неисправности соседнего узла отличается от механизма обнаружения неисправности канала, в частности, из-за наличия нескольких параллельных ненумерованных каналов.

Расширением сообщения *Hello* являются объекты HELLO REQUEST и HELLO ACK. Обмен сообщениями *Hello* поддерживает независимый выбор интервала обнаружения. Любой из соседних узлов может автоматически передавать запросы HELLO REQUEST, каждый из которых требует подтверждения.

Обнаружение неисправностей в соседнем узле основано на получении и хранении некоторого «эталонного» для этого узла значения. Если обнаруживается изменение эталонного значения, или если сосед неверно отвечает на локально объявленное эталонное значение, это свидетельствует о неисправности соседнего узла.

Сообщениями *Hello* всегда обмениваются соседние узлы RSVP. Поле IP TTL всех исходящих сообщений *Hello* имеет значение 1. Номер типа сообщения (Msg Type) равен 20. Его формат выглядит следующим образом:

```
<Hello Message> ::= <Common Header> [ <INTEGRITY> ]
                                          <HELLO>
```

Объект HELLO имеет класс = 22. Для него определены два C-типа: C-тип = 1, объект HELLO REQUEST, и C-тип = 2, объект HELLO ACK. Они имеют одинаковый формат, представленный на рис. 4.20.

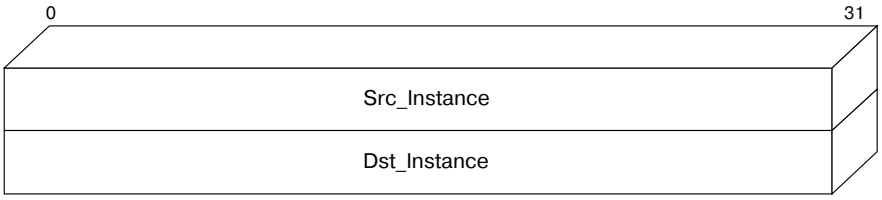


Рис. 4.20. Формат объекта HELLO

Src_Instance — 32-битовое число, представляющее эквивалент отправителя. Это значение должно изменяться, если узел перезагружается или если потеряна связь с соседним узлом. Поле не должно иметь нулевое значение.

Dst_Instance — 32 битовое число — последнее полученное от соседнего узла значение *Src_Instance*. Это поле должно иметь значение 0, если от соседнего узла еще не было получено ни одного *Src_Instance*.

Сообщения *Hello* являются необязательными и могут игнорироваться всеми узлами, не желающими ими обмениваться. Узел периодически генерирует сообщения с объектом HELLO REQUEST. Периодичность определяется договоренностью между соседними узлами, по умолчанию устанавливается периодичность 5 мс. Создавая объект HELLO REQUEST, узел заполняет поля *Src_Instance* и *Dst_Instance* соответствующими значениями. Создание сообщения должно быть приостановлено, если узел получил HELLO REQUEST до истечения обусловленного интервала. Узел, получивший объект-запрос, должен сформировать в ответ объект HELLO ACK. Если обнаружено, что связь потеряна, узел может снова инициировать обмен сообщениями *Hello*, при этом он должен выбрать значение *Src_Instance*, отличное от предыдущего. Следует отметить, что такой механизм может использоваться и для обнаружения

неисправности нумерованных каналов путем организации аналогичного обмена сообщениями между двумя портами, связанными этими каналами.

4.8. Управление трафиком в MPLS

Протокол MPLS стратегически достаточен для управления трафиком, и существует возможность автоматизировать функции такого управления. Для этого сигнальные протоколы MPLS должны переносить информацию, которая необходима для работы механизмов управления трафиком, находящихся на прикладном уровне, а также создавать LSP с явно заданными маршрутами. При этом есть возможность получить дополнительную гибкость, если маршрут может быть задан как строго, так и не строго, т.е. если группа узлов может быть задана как «абстрактный узел», в рамках которого существует известная свобода выбора маршрута.

Перед IETF, точнее, перед ее рабочей группой MPLS, возникла задача выбора такого протокола. Лучшими оказались два варианта: RSVP-TE, рассматриваемый в этой главе, и CR-LDP, рассмотренный в главе 3. В первом варианте протокол RSVP должен делать в сети MPLS то же, что он делает в сетях IP, а именно — обрабатывать информацию, связанную с QoS, и резервировать ресурсы. Необходимо лишь добавить к этому возможности распределения меток. Во втором варианте, как это показано в параграфе 3.5, добавить к уже используемому для распределения меток в MPLS протоколу LDP несколько новых объектов, обеспечивающих перенос информации о QoS, также не представлялось сложным. На сегодняшний день оба протокола развиты до статуса *proposed standard*.

Механизмам TE (Traffic Engineering) в MPLS будет посвящена глава 9, в которой мы рассмотрим варианты и примеры использования разных механизмов управления трафиком в MPLS, в том числе, и возможности модели DiffServ, а также сравним протоколы CR-LDP и RSVP-TE. Но прежде целесообразно обсудить протоколы маршрутизации, что и делается в трех следующих главах.

Глава 5
Протокол OSPF

5.1. Протоколы OSPF и RIP

При рассмотрении в предыдущих главах принципов MPLS обсуждалась необходимость заполнения в маршрутизаторах LSR таблиц меток, используемых при маршрутизации пакетов по сети MPLS. Для этого в каждом из узлов сети с использованием протокола маршрутизации создается база топологической информации о сетевых маршрутах. Наряду с рассматриваемыми в двух следующих главах протоколами BGP4 и IS-IS, для этой цели может быть применен *протокол маршрутизации по состояниям каналов OSPF (Open Shortest Path First — первым выбирается кратчайший путь).* Поскольку OSPF используется наиболее часто, рассмотрение протоколов маршрутизации для MPLS в этой книге начинается именно с него.

Слово *Open* в названии протокола означает, что спецификация протокола маршрутизации свободно распространяется (в отличие, например, от спецификации протокола EIGRP).

На рис. 5.1 приведены номера документов RFC, отражающие эволюцию протокола OSPF. Силами IETF были разработаны две версии OSPF, описанные, соответственно, в RFC 1131 и RFC 2328. Несмотря на некоторые весьма любопытные идеи, первая версия уже несколько устарела и в книге не рассматривается. Вторая версия OSPF поддерживает только маршрутизацию IP-пакетов, чего, впрочем, вполне достаточно для того, что требуется MPLS от протокола маршрутизации. В процессе разработки второй версии были созданы документы RFC 1254 «Protocol Analysis» и RFC 1246 «Experience with the OSPF Protocol», где дан анализ работы протокола в Интернет и сформулированы основы для спецификации второй версии, которые вошли в RFC 1247 «OSPF Version 2», вышедший в 1991 г. Последовательно сменявшие друг друга RFC посвящены практическому использованию MIB (Management Information Base). Это RFC 1248, вытесненный в том же 1991 году сначала документом RFC 1252, а затем — RFC 1253; сейчас же действителен документ RFC 1850.

Несколько позже был выпущен RFC 1364 «BGP OSPF Interaction», содержащий спецификации по созданию маршрутизаторов ASBR, которые поддерживают протокол BGP, а еще через год он был заменен RFC 1403. К ним мы еще вернемся в главе 7, посвященной BGP. В документе RFC 1370 «Applicability Statement for OSPF» обсуждаются области применения OSPF, а в качестве спецификаций в настоящее время действуют следующие документы: RFC 1584, определяющий поддержку многоадресной рассылки, RFC 1586, посвященный работе OSPF в сетях Frame Relay, документ RFC 3101 — новая версия документа RFC 1587, в которой представлен вариант спецификации протокола с упоминающимся далее понятием тупиковой области NSSA, документ RFC 1745 о маршрутизаторе ASBR, поддерживающем BGP4/IDRP, документ RFC 1765, посвященный проблеме перегрузки баз данных OSPF, документ RFC 1793 — расширение OSPF для соединений по требованию, документ RFC 2370, содержащий описание трех дополнительных Opaque LSA, документ RFC 2676, описывающий поддержку QoS, документ RFC 2740, посвященный OSPF для IPv6, и документ RFC 2844, специфицирующий работу OSPF в сетях ATM с поддержкой стандарта Proxy-RAR. Основным же действующим документом по OSPF является RFC 2328.

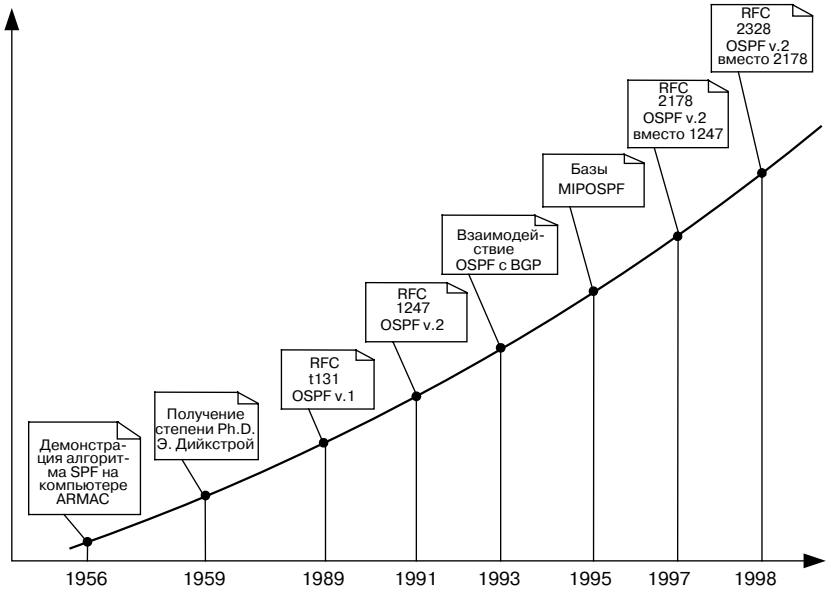


Рис. 5.1. Эволюция протокола OSPF

Протокол OSPF относится к *протоколам внутреннего шлюза IGP (Interior Gateway Protocol)*. К этой категории относятся протоколы маршрутизации, обеспечивающие обмен информацией в пределах *автономной системы AS (Autonomous System).* Наиболее попу-

лярными из них являются протоколы IGRP, RIP, упоминаемый ниже в этом параграфе, OSPF, составляющий предмет данной главы, и IS-IS, которому посвящена следующая глава. Вторая категория протоколов *EGP (Exterior Gateway Protocol)* существенно отличается от этих протоколов, дополняя функции маршрутизации в сети, что будет рассмотрено в главе 7, посвященной протоколу BGP.

Предшественник OSPF *протокол RIP (Routing Information Protocol)* был одним из первых протоколов внутренней маршрутизации, применявшихся в Интернет. Своим происхождением и названием RIP обязан *архитектуре XNS (Xerox Network Systems)*. Широкое распространение протокола RIP связано с тем, что он был включен в версию 1982 года операционной системы *Berkeley UNIX*, поддерживающей стек протоколов TCP/IP, под названием *роут ди (route d)*. Это название соответствует принятому в системе UNIX правилу именования фоновых программных процессов — демонов. К имени каждого такого процесса добавляется буква *d* от слова Daemon и всегда читается именно как *роут ди.*

Версия 1 протокола RIP, специфицированная в RFC 1058, в качестве метрики стоимости маршрута использует количество содержащихся в нем участков. Максимальная стоимость ограничена значением 15, таким образом, выходной маршрутизатор, количество участков до которого более 15, считается недостижимым. В следующих версиях протокола RIP сохранен тот же принцип *дистанционно-векторной маршрутизации,* основывающейся на *векторе расстояния (Distance Vector).* В качестве метрики маршрутизации RIP использует число пересылок между узлами — *(hops)*. Если между отправителем и получателем расположено *N* маршрутизаторов, считается, что между ними *N+1*пересылок. Поиск эффективного маршрута в этом случае производится с помощью алгоритма Беллмана-Форда, описание которого мы отложим до главы 7.

Эта метрика не учитывает различий в пропускной способности, величины загрузки или надежности отдельных сегментов сети. Для этого существуют объективные основания. Они же являются причиной, по которой протокол RIP не используется в MPLS и далее в этой книге не рассматривается.

Дело в том, что протокол RIP создавался для первых маршрутизаторов Интернет, построенных на мини-ЭВМ типа Honeywell 516, выполнялся на 8-разрядных микропроцессорах типа Intel 8080 или Zilog Z80, обеспечивая простоту и экономное использование более чем скромного быстродействия и оперативной памяти тех компьютеров. Современные же вычислительные возможности, в полном соответствии с законом Мура, позволяют применять куда более сложные и ресурсоемкие средства маршрутизации, к которым относится и протокол OSPF.

5.2. Метрики OSPF

В OSPF используется принцип *контроля состояния канала (link-state protocol),* а метрика представляет собой оценку эффективности связи в этом канале: чем меньше метрика, тем эффективнее организация связи. В простейшем случае метрика маршрута может равняться его длине в пересылках (hops), как это имеет место в протоколе RIP. Но в общем случае значения метрики могут определяться в гораздо более широком диапазоне.

Метрика, оценивающая пропускную способность канала, определяется, например, компанией CISCO, как количество секунд, нужное для передачи 100 Мбит. Имеется следующая формула для вычисления метрики доставки информации через каналы сети OSPF:

метрика = 10^8/скорость передачи в битах в секунду.

По этой формуле вычислены, например, следующие метрики:

- Канал со скоростью ≥ 100 Мбит/с соответствует метрике 1.
- Сеть Ethernet/802.3 соответствует метрике 10.
- Тракт E1 2.048 Мбит/с соответствует метрике 48.
- Тракт T1 1.544 Мбит/с соответствует метрике 65.
- Канал 64 Кбит/с соответствует метрике 1562.
- Канал 56 Кбит/с соответствует метрике 1785.
- Канал 19.2 Кбит/с соответствует метрике 5208.
- Канал 9.6 Кбит/с соответствует метрике 10416.

Кроме того, протокол OSPF позволяет определить для любой сети значения метрики в зависимости от *типа услуги ToS (Type of Service).* Для каждой из метрик протокол OSPF строит отдельную таблицу маршрутизации. Чаще всего OSPF выбирает маршрут на основании полосы пропускания канала. Загрузка канала представляет собой величину, которая изменяется в зависимости от использования канала, причем интенсивная эксплуатация канала повышает его загрузку, и поэтому при маршрутизации бывает целесообразно выбирать менее нагруженные каналы. Еще одна возможная метрика — задержка — определяет время в микросекундах, которое требуется маршрутизатору для обработки, установки в очередь и передачи пакетов.

В случае, когда имеется несколько маршрутов с одинаковым значением метрики, маршрутизаторы могут использовать для передачи пакетов все эти маршруты, обеспечивая *балансировку нагрузки*. Маршрутизатор OSPF помещает в таблицу маршрутизации все маршруты с одинаковыми значениями метрики, и балансировка нагрузки между маршрутами происходит автоматически. Стандартизованный порядок расчета метрик, оценивающих надежность, задержку и стоимость, пока не определен. Эти вопросы решаются администратором сети.

Для сравнения отметим, что рассматриваемый в следующей главе протокол IS-IS также использует неструктурированные метрики, значения которых находятся в диапазоне от 0 до 1023. По умолчанию всем транзитным пересылкам в IS-IS присваивается метрика 10. В OSPF же значения метрик находятся в диапазоне от 0 до 65385.

Итак, OSPF представляет собой протокол, основанный на контроле состояния каналов, распространяющий эту информацию и определяющий на ее основе маршруты наименьшей стоимости в заданной метрике. Именно с его помощью LSR отображает видимый ему граф домена сети MPLS, где для каждой пары смежных вершин графа (маршрутизаторов) указано ребро (канал), их соединяющее, и метрика этого ребра. Граф считается ориентированным, т.е. ребро, соединяющее LSR1 с LSR2, и ребро, соединяющее LSR2 с LSR1, могут быть разными, или это может быть одно и то же ребро, но с разными метриками.

Маршрутизатор, работающий по протоколу OSPF, выполняет последовательно три операции: определяет отношения соседства и смежности с другими маршрутизаторами, обменивается с ними OSPF-пакетами извещений LSA, формируя таким образом полную топологическую карту сети, а затем вычисляет дерево маршрутов, используя алгоритм *«первым выбирается кратчайший путь» SPF (Shortest Path First)*, известный также по имени его создателя как *алгоритм Дийкстры*.

5.3. Алгоритм Дийкстры

Знаменитый алгоритм определения кратчайшего пути SPF был придуман выдающимся теоретиком программирования Эдсгером Дийкстрой в 1956 году для демонстрации возможностей компьютера ARMAC. Сам автор, как это часто случается, не придавал большого значения именно этой работе, опубликовав ее со значительным опозданием. Сегодня этот алгоритм применяется при организации авиаперевозок, в дорожном строительстве, в производстве печатных плат, а также в протоколах маршрутизации OSPF и IS-IS, рассматриваемых в этой книге.

Для сети MPLS с помощью этого алгоритма протокол OSPF, основываясь на базе данных об условиях использования возможных связей, вычисляет кратчайшие пути между заданным LSR — вершиной графа — и всеми остальными вершинами. Результатом работы алгоритма является таблица, где для каждой вершины графа сети MPLS указан список ребер, соединяющих ее со всеми другими вершинами этого графа по кратчайшему пути.

Суть алгоритма иллюстрирует следующая процедура. Представим изображенную на рис. 5.2 сеть MPLS, содержащую 7 LSR, как набор из 7 фишек, лежащих на поверхности стола и соединенных между собой нитями разной длины. Пусть, например, алгоритм Дийкстры выполняется в маршрутизаторе LSR4. Постепенно поднимаем со стола фишку, соответствующую LSR4. Нити, связывающие эту фишку с другими, начинают натягиваться, и следующей со стола будет поднята фишка LSR2, связанная с LSR4 самой короткой нитью. При дальнейшем подъеме фишки LSR4 мы поднимем с поверхности стола и фишку LSR5. Кратчайший (в рассмотренном выше смысле) путь между LSR4 и LSR5 представит либо нить, связывающая соответствующие этим двум LSR фишки непосредственно, либо составной путь из нитей между фишками LSR4 и LSR2, LSR2 и LSR3, LSR3 и LSR5. Продолжая процедуру подъема фишки LSR4, мы шаг за шагом поднимем все фишки, находя каждый раз кратчайший путь между LSR4 и тем LSR, которому соответствует очередная поднимаемая нитью фишка.

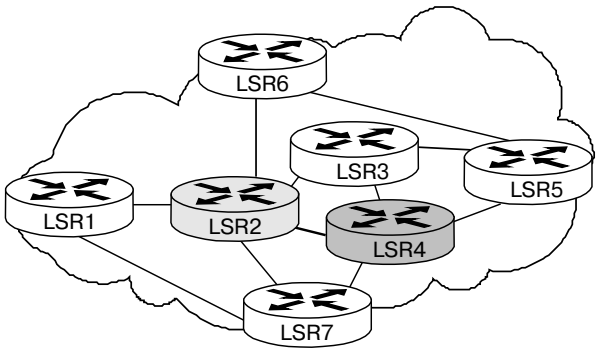


Рис. 5.2. Иллюстрация алгоритма Дийкстры

Теперь запишем этот алгоритм более строго. Пусть M — множество обработанных вершин, т.е. вершин, кратчайший путь к которым от заданной вершины LSR4 уже найден; N — множество оставшихся вершин графа (т.е. множество вершин графа за вычетом множества M); L — упорядоченный список путей.

Шаг 1. Определить M={LSR4} и N={все вершины графа, кроме LSR4}. Поместить в список L все односегментные (длиной в одно ребро) пути, начинающиеся из LSR4, в порядке возрастания их метрик.

Шаг 2. Если L пуст или первый путь в L имеет бесконечную метрику, то отметить все вершины множества N как недостижимые и закончить работу алгоритма.

Шаг 3. Рассмотрим l — кратчайший путь в списке L. Удалить l из L. Пусть LSR* — последний узел в l. Если LSR*$\in$ M, перейти к шагу 2; иначе l является кратчайшим путем из LSR4 в LSR*; перенести LSR* из N в M.

Шаг 4. Построить набор новых путей, подлежащих рассмотрению, путем добавления к пути l всех односегментных путей из LSR* в N. Метрика каждого нового пути равна сумме метрики l и метрики соответствующего односегментного отрезка, начинающегося в LSR*. Добавить новые пути в упорядоченный список L, поместив их на нужные места в соответствии со значениями метрик. Возврат к шагу 2.

Теперь вернемся к рассмотренному примеру и обратим внимание на наличие двух разных маршрутов между LSR4 и LSR5: прямого и составленного из участков между LSR4 и LSR2, LSR2 и LSR3, LSR3 и LSR5. В том случае, если между двумя узлами сети существует несколько маршрутов с близкими по значению метриками, протокол OSPF позволяет распределять трафик по этим маршрутам в пропорции, соответствующей значениям метрик. Например, если прямой маршрут между LSR4 и LSR5 имеет метрику 4, а составной маршрут из участков LSR4 и LSR2, LSR2 и LSR3, LSR3 и LSR5 имеет метрику 8, то две трети трафика будет направлено по первому из них, а оставшаяся треть — по второму.

Суммарный эффект такого решения заключается в уменьшении средней задержки прохождения дейтаграмм от отправителя к получателю, а также в сглаживании колебаний задержки.

Еще одно преимущество поддержки альтернативных маршрутов связано с соображениями надежности. Когда при использовании только одного из возможных маршрутов он внезапно выходит из строя, весь трафик должен быть срочно переведен на альтернативный маршрут. При массовом переключении больших объемов трафика с одного маршрута на другой весьма велики потери и даже вероятно образование затора на новом маршруте. Если же до аварии использовалось разделение трафика по нескольким маршрутам, отказ одного из них вызовет ремаршрутизацию лишь части трафика, что может сгладить отрицательные последствия аварии.

5.4. Области OSPF

Здесь уместно вернуться к двум основным понятиям, связанным со вспомогательной ролью OSPF. В терминологии этого протокола *домен* сети MPLS, рассмотренный в главе 1 с использованием неоднократно модифицировавшегося рис. 1.3, в этой главе может соответствовать понятиям *автономная система* и *область* сети OSPF. Границы и в том, и в другом случае несколько условны. Более того, могут иметь место разные организационные и/или технологические причины, на основании которых строится, например, такая сетевая иерархия для OSPF, когда домен MPLS содержит

порядка 100 узлов, а требуемая скорость сходимости OSPF делает целесообразным разбиение этого домена на две-три зоны по 30 — 50 узлов в каждой. Или, наоборот, небольшую по количеству маршрутизаторов зону OSPF сетевой администратор может определить как два разных домена MPLS, например, в силу их принадлежности разным корпоративным пользователям (разным компаниям в одном здании бизнес-центра). Принимая во внимание эти соображения, поясним используемые здесь и далее термины *автономная система* и *область*.

Автономная система AS (Autonomous System) представляет собой сеть, находящуюся под единым административным управлением.

Область OSPF (OSPF area) представляет собой логическую подсистему автономной системы, в которой OSPF функционирует в качестве ее протокола внутренней маршрутизации. Области «укрупняют» маршрутную информацию протокола OSPF и помогают скрывать детали топологии сети. Так, топология одной области не известна ни в какой другой области. Внутренние маршрутизаторы области вообще не владеют информацией о топологии использующих OSPF сетей, которые находятся за пределами этой области, что дает выигрыш в затратах на поддержание маршрутной информации. Эта, а также некоторые другие возможности делают OSPF хорошо масштабируемым протоколом маршрутизации для крупных сетей. А сам протокол OSPF основан на концепции областей как совокупностей смежных сетей и относящихся к ним маршрутизаторов с интерфейсами, связывающими их с этими сетями и с узлами в них.

Автономная система, базирующаяся на протоколе OSPF, может представлять собой одну область или состоять из нескольких областей. В каждой области работает собственная копия *алгоритма маршрутизации по состоянию каналов*, что позволяет каждой области формировать свою базу данных сетевой топологии. Именно область ограничивает охват лавинной рассылки уведомлений, т.к. уведомления не выходят за пределы области, в которой они были сформированы.

Каждая область идентифицируется 32-битовым *идентификатором area ID*, внешне (но только внешне) схожим с форматом IP-адреса. Одна из областей выполняет специальные функции; она называется *опорной областью (backbone)* и помечается как 0.0.0.0. Остальные области взаимодействуют друг с другом через опорную область. Общепринятый идентификатор этой области именно 0.0.0.0, а возможные идентификаторы других областей строго не определены. Например, для определения области 1 нет строго назначенного идентификатора 1.1.1.1 или 0.0.0.1.

Кроме того, каждому маршрутизатору области присваивается свой идентификатор, независимо от идентификатора области, в которой все они находятся. Это 32-битовое слово однозначно определяет маршрутизатор в автономной системе. Обычно идентификатор маршрутизатора выбирается по IP-адресу одного из его интерфейсов.

С учетом этого протокол OSPF разграничивает функции маршрутизаторов в зависимости от того, какое место они занимают в автономной системе OSPF, объединяющей все маршрутизаторы, которые ведут обмен информацией под управлением общего протокола. Разумеется, маршрутизаторы разделяются на классы с точки зрения протокола OSPF, а не технологии MPLS, но так как все маршрутизаторы MPLS поддерживают протокол OSPF (если он выбран в качестве протокола, с помощью которого составляется топологическая карта сети), подобная классификация полностью к ним применима. Таким образом, термины *маршрутизатор LSR* и *OSPF-маршрутизатор* используются в этой главе как синонимы.

Договорившись об этом, отметим, что, с точки зрения протокола OSPF, имеются маршрутизаторы четырех типов:

- внутренний маршрутизатор IR (Internal Router), все интерфейсы которого находятся внутри одной области OSPF;
- маршрутизатор опорной области BR (Backbone Router), все интерфейсы которого находятся внутри опорной области 0;
- пограничный маршрутизатор области ABR (Area Border Router), располагающийся на границе двух областей OSPF;
- пограничный маршрутизатор автономной системы ASBR (Autonomous System Boundary Router), который располагается на границе двух автономных систем, поддерживающих OSPF.

Кроме этого назначаются два маршрутизатора, так и называемые *назначенный маршрутизатор DR (Designated Router)* и *резервный назначенный маршрутизатор BDR (Backup Designated Router)*, которые являются центральными узлами сбора всех сообщений о корректировках. Все остальные маршрутизаторы являются по отношению к маршрутизаторам DR и BDR подчиненными, к чему мы вернемся далее в этой главе.

Здесь лишь отметим, что один и тот же маршрутизатор может выполнять в системе несколько функций одновременно.

5.5. Структура OSPF-пакета

Пакеты упомянутого в начале главы протокола RIP передаются при помощи протокола UDP, работающего поверх протокола IP, В отличие от этого, пакеты OSPF передаются непосредственно протоколом IP. Для обозначения маршрутизаторов OSPF в сетях «точка-точка» или в локальных вещательных сетях для IP-пакетов используется стандартный групповой IP-адрес 224.0.0.5. В сетях, не поддерживающих вещательную рассылку, используются определенные IP-адреса получателей, настраиваемые в маршрутизаторе заранее.

Все OSPF-пакеты имеют один и тот же 24-байтовый заголовок (рис. 5.3) со следующими полями:

Номер версии (Version Number). Указывает номер версии протокола OSPF, применяемой в маршрутизаторе, который сформировал этот пакет. Текущая версия протокола OSPF имеет номер 2.

Тип пакета (Packet Type). Один из пяти типов пакетов, обсуждаемых в следующем параграфе.

Длина пакета (Packet Length). Длина OSPF-пакета в байтах, включая заголовок и содержимое.

Идентификатор маршрутизатора (Router ID). Идентифицирует маршрутизатор — источник пакета. Каждому маршрутизатору внутри области присваивается уникальный 32-разрядный идентификатор, а протокол OSPF использует эти идентификаторы для выбора маршрутизаторов DR и BDR. Администратор может установить эти значения вручную, или они устанавливается автоматически.

Идентификатор области (Area ID). Идентифицирует область, откуда поступил OSPF-пакет. Area 0 всегда имеет идентификационный номер 0.0.0.0.

Контрольная сумма (Checksum). Используется для контроля целости пакета OSPF. Стандартная контрольная сумма протокола IP для всего содержимого OSPF-пакета, кроме 64-битового поля аутентификации, вычисляется как сумма 16-разрядных слов пакета в дополнительном коде.

Тип аутентификации (AuType). Идентифицирует процедуру аутентификации, которая должна использоваться для этого пакета. Среди возможных вариантов: отсутствие аутентификации, простой пароль, шифрование. Если включена функция простого пароля, маршрутизатор будет формировать смежные взаимоотношения только с теми маршрутизаторами, которые имеют одинаковый с ним пароль. Поле типа аутентификации имеет длину 16 битов.

Данные аутентификации. 64-разрядное поле, используемое процедурой аутентификации.

Рис. 5.3. Заголовок OSPF-пакета

5.6. Типы пакетов OSPF

Следом за заголовком располагается тело OSPF-пакета, содержимое которого зависит от типа пакета.

Протокол OSPF использует пакеты 5 типов:

- Тип 1. Приветствие (Hello).
- Тип 2. Описание базы данных DD (Database Description).
- Тип 3. Запрос сведений о состоянии каналов (Link State Request).
- Тип 4. Корректировка сведений о состоянии каналов (Link State Update).
- Тип 5. Подтверждение получения сведений о состоянии каналов (Link State Acknowledgement).

Каждый заголовок пакета OSPF включает в себя дополнительный заголовок со специфической маршрутной информацией, относящейся к одному из пяти типов пакетов. Дополнительный заголовок следует после общих полей заголовка OSPF.

5.6.1. Пакеты-приветствия

Пакеты-приветствия типа 1 (Hello) используются для обнаружения соседей. С их помощью устанавливаются и поддерживаются взаимоотношения между смежными маршрутизаторами OSPF. Каждый маршрутизатор периодически передает пакеты этого типа через все свои интерфейсы. Пакет содержит идентификаторы соседних маршрутизаторов, чьи пакеты-приветствия уже получены. На рис. 5.4 приведен основной формат такого пакета.

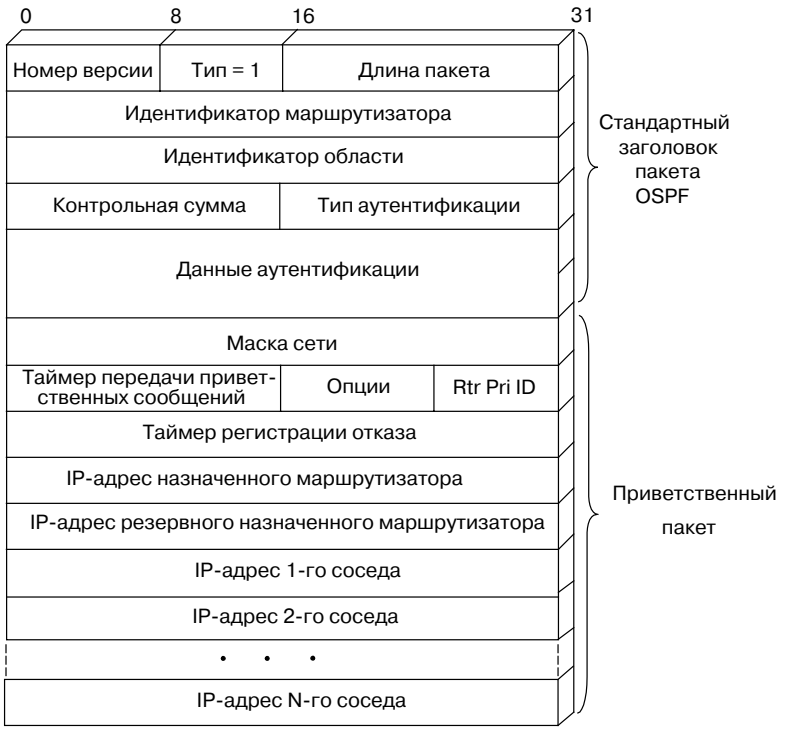


Рис. 5.4. Структура приветственного пакета

Маска сети (Network Mask). В этом поле указывается маска подсети, которая относится к данному интерфейсу. Если не используется ни один из методов организации подсетей, то в этом поле указывается количество битов в части адреса, содержащей обозначение сети.

Таймер передачи сообщений Hello (Hello Interval) определяет, как часто передаются пакеты-приветствия. Отмеряет число секунд, которое должно пройти с момента передачи одного пакета-приветствия до момента передачи другого. Значение таймера изменяется в зависимости от топологии уровня 2 сети, использующей OSPF. В вещательных сетях маршрутизаторы передают сообщения Hello по умолчанию каждые 10 секунд. В коммутируемой сети маршрутизаторы передают приветствия каждые 30 секунд.

Опции (Options). В поле опций указываются возможности протокола OSPF, которые поддерживает данный маршрутизатор. Для всех маршрутизаторов поддержка опций не обязательна. Если эта возможность не используется, маршрутизатор либо отбрасывает опции, либо игнорирует их. Определяются две следующие опции:

- T-bit. Используется для индикации того, что маршрутизатор OSPF может поддерживать разные типы услуг ToS и обеспечивать запрашиваемое качество обслуживания. Маршрутизаторы, поддерживающие ToS, указывают это путем присвоения биту T значения 1. Если бит T=0, это означает, что маршрутизатор не поддерживает функции ToS. Маршрутизаторы, которые поддерживают функции ToS, создают древовидные схемы кратчайших маршрутов, помещая себя в корень дерева. Одно дерево создается только для маршрутов с заданным качеством обслуживания, а другое — для маршрутов, не поддерживающих заданное качество обслуживания.
- E-bit. Маршрутизаторы, работающие с этим битом, могут обрабатывать внешнюю информацию о маршрутах, не управляемых протоколом OSPF. Маршрутизаторы тупиковой области не поддерживают сообщения о корректировках внешних маршрутов и поэтому не работают с этим битом и не распознают его. Маршрутизаторы ASBR всегда используют E bit.

Идентификатор приоритета маршрутизатора (Router Priority ID) обеспечивает выбор маршрутизаторов DR и BDR для каждого сегмента. Маршрутизатор, имеющий ID наивысшего приоритета в области, становится DR. Если в сети не используются данные о приоритете маршрутизатора, значение этого параметра по умолчанию равно 1. Если маршрутизатору присвоен приоритет 0, то он не может претендовать на роль BDR или DR.

Значение таймера регистрации отказа маршрутизатора (Router Dead Interval). Это поле содержит число секунд, которое должно пройти с момента получения от соседнего маршрутизатора последнего пакета-приветствия, после чего тот должен быть объявлен «мертвым». По умолчанию маршрутизатор считает соседа «мертвым», если не получает от него приветствий в течение четырех стандартных интервалов между моментами их передачи, т.е. значение этого интервала устанавливается в четыре раза большим, чем значение *таймера Hello Interval,* значение зависит от топологии уровня 2 сети, использующей OSPF. Как отмечалось выше в этом параграфе, в вещательных сетях маршрутизаторы передают приветствия каждые 10 секунд, поэтому интервал для определения «мертвого» маршрутизатора составляет 40 секунд, а в коммутируемой сети маршрутизаторы передают приветствия по умолчанию каждые 30 секунд, и интервал для определения «мертвого» маршрутизатора составляет, соответственно, 120 секунд.

Назначенный маршрутизатор (Designated Router). В этом поле указывается IP-адрес маршрутизатора DR, если он известен маршрутизатору, передающему пакет. Если этот адрес неизвестен, в поле вводится значение 0.0.0.0.

Резервный назначенный маршрутизатор (Backup Designated Router). В этом поле указывается IP-адрес маршрутизатора BDR, если он известен маршрутизатору, передающему пакет. Если этот адрес неизвестен, в поле вводится значение 0.0.0.0.

Соседний маршрутизатор (Neighbor). В это поле вводится перечень идентификационных номеров всех соседних маршрутизаторов, о которых маршрутизатор, передающий пакет, узнает из полученных от них пакетов-приветствий.

5.6.2. Пакет описания базы данных OSPF

Пакеты с описанием базы данных (тип 2) несут информацию, необходимую для построения топологических баз данных в смежных маршрутизаторах. Как видно из структуры пакета на рис. 5.5, маршрутизаторы обмениваются не содержимым баз данных, а описанием их структуры. Это позволяет маршрутизаторам синхронизировать свои данные о сетевой топологии.

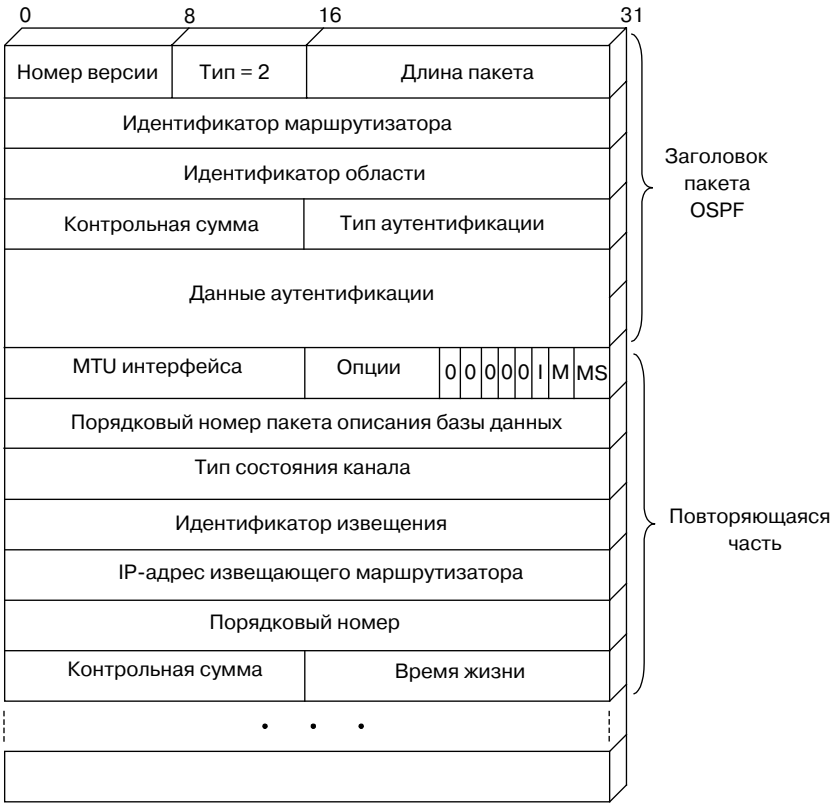


Рис. 5.5. Структура пакета описания базы данных

Опции (Options). Поле опций описывает возможности OSPF, которые поддерживает маршрутизатор, передающий пакет. Это поле имеется и в OSPF-пакетах других типов, и в пакетах разных типов его биты несут разную информацию. Пакет с описанием базы данных использует только три последних бита этого поля:

- I (Init). I=1 означает, что передается первый пакет с описанием базы данных.
- M (More). M=1 означает, что должны последовать другие пакеты с описанием базы данных. Если бит M имеет значение 0, значит это последний пакет.
- MS (главный/подчиненный) указывает, какую функцию несет передающий маршрутизатор — главного (DR) или подчиненного. MS=1 означает главный маршрутизатор, а MS=0 — подчиненный маршрутизатор.

Порядковый номер пакета (Sequence Number). Отправитель упорядочивает последовательность всех пакетов с описанием базы данных, а получатель подтверждает прием каждого пакета. При отправке первого DD-пакета для него выбирается уникальный начальный номер (InitBit = 1); номера всех остальных пакетов последовательно возрастают.

5.6.3. Запросы сведений о состоянии каналов

Пакеты OSPF с запросами сведений о состоянии каналов (тип 3) служат для получения текущей маршрутной информации или для загрузки базы данных из соседнего маршрутизатора. На рис. 5.6 показан формат пакета с запросом сведений о состоянии каналов.

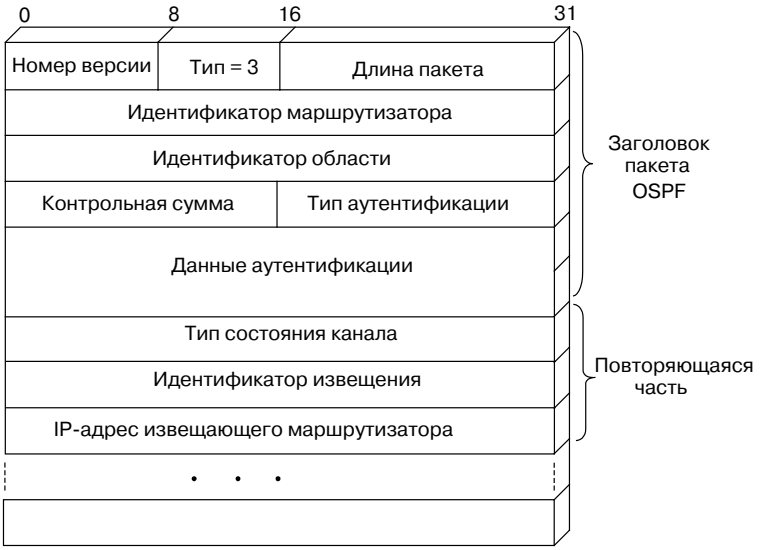


Рис. 5.6. Структура пакета запроса сведений о состоянии каналов

Тип состояния канала (Link State Type). Это поле служит для идентификации нужного LSA.

Идентификатор извещения (Link State ID) содержит уникальный идентификационный номер извещения со сведениями, используемыми другими маршрутизаторами.

Извещающий маршрутизатор (Advertising Router) идентифицирует маршрутизатор, который первоначально генерировал извещение о состоянии каналов.

5.6.4. Корректировка сведений о состоянии каналов

Пакеты OSPF типа 4 с корректировками сведений о состоянии каналов несут маршрутную информацию, переданную в ответ на запрос. Один пакет с корректировкой каналов может включать в себя несколько извещений. На рис. 5.7 показан формат такого пакета.

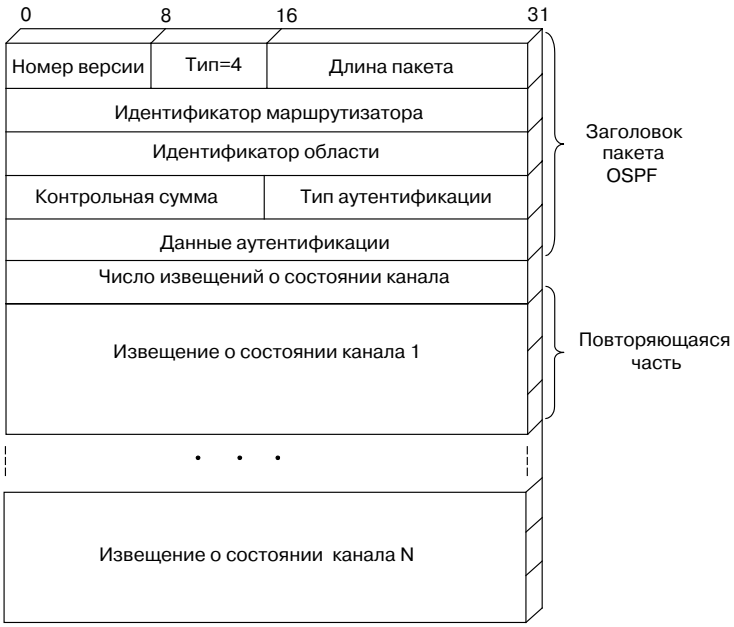


Рис. 5.7. Структура пакета корректировки состояния каналов

Число извещений (Number of Advertisement). В этом поле указывается число извещений, включенных в пакет.

Извещения о состоянии каналов (Link State Advertisements). Это поле представляет собой главное содержимое пакета. Оно включает в себя перечень LSA. Каждое LSA имеет общий заголовок, который сопровождается информацией о типе этого LSA.

5.6.5. Подтверждение извещений о состоянии каналов

Пакет подтверждения (тип 5) просто подтверждает получение маршрутной информации о состоянии каналов в случае надежного ее приема. Этот пакет имеет такой же формат, как и пакет описания базы данных, и содержит перечень заголовков извещений о состоянии каналов. Формат этого пакета представлен на рис. 5.8.

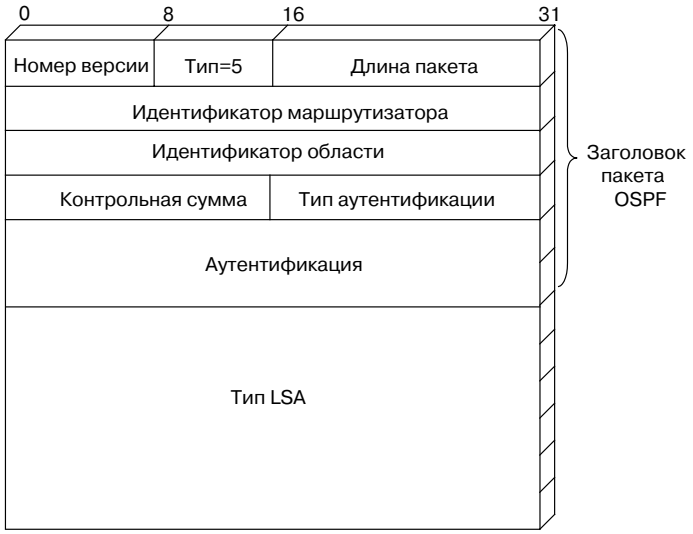


Рис. 5.8. Структура пакета подтверждения

5.7. Извещения LSA

В протоколе OSPF используется десять типов *извещений о состоянии каналов* (в литературе называемых также *объявлениями* или *анонсами) LSA (Link State Advertisement)*. С точки зрения области их распространения эти LSA бывают *внутриобластными (intra-area) извещениями, межобластными (inter-area) извещениями* или *внешними (external) извещениями.* В табл. 5.1 представлены разные типы LSA.

Распространяемые только *внутри области (intra-area)* извещения описывают состояние каналов маршрутизатора и каналов связи сетей внутри одной области OSPF. Это — указанные в табл. 5.1 извещения типа 1 (router link), инициируемые всеми маршрутизаторами, и типа 2 (network link), инициируемые только назначенными маршрутизаторами. Извещения типа 1 маршрутизаторы передают по групповому адресу 224.0.0.6 (групповой адрес для назначенных маршрутизаторов DR и для резервных назначенных маршрутизаторов BDR), указывая сети, с которыми есть прямая связь, и состояния

интерфейсов, соединенных с этими сетями. Для локальных сетей с вещательным трафиком маршрутизатор DR передает извещение LSA типа 2 (network link) по групповому адресу **224.0.0.5** (групповой адрес для всех маршрутизаторов).

Таблица 5.1. Типы пакетов LSA

Тип LSA	Наименование названия
1	Извещение о каналах маршрутизатора (router link advertisement)
2	Извещение о каналах сети (network link advertisement)
3	ABR суммарное извещение о состоянии каналов
4	ASBR суммарное извещение о состоянии каналов
5	Извещение о внешних маршрутах автономной системы AS
6	Вещательные групповые LSA
7	Внешние извещения NSSA (Not-So-Stubby Area)
9	Непрозрачные извещения в пределах сети (Opaque LSA: Link-local scope)
10	Непрозрачные извещения в пределах области (Opaque LSA: Area-local scope)
11	Непрозрачные извещения в пределах автономной системы (Opaque LSA: AS scope)

Межобластные извещения пересылаются пограничными маршрутизаторами области (ABR-Area Border Router) в случаях, когда автономная система разбита на ряд областей, причем обычно маршрутизаторы ABR соединяют эти области с главной опорной областью (backbone 0) для обслуживания всего межобластного трафика. Межобластные извещения суммируют маршруты внутри области OSPF, и маршрутизаторы ABR передают эти извещения в опорную область, где их обрабатывают другие маршрутизаторы ABR, распространяя полученную информацию каждый в своей области. Имеется два типа межобластных извещений: тип 3 и тип 4.

В извещениях типа 3 суммируется информация об областях, соединенных с опорной областью Area 0. Кроме того, маршрутизаторы ABR используют эти извещения для суммирования маршрутов из других подобластей, проходящих через опорную область в их собственную.

Маршрутизаторы ABR передают LSA типа 4 для идентификации маршрутов к пограничным маршрутизаторам автономной системы (ASBR — Autonomous System Boundary Router), которые обеспечивают доступ к внешним сетям, расположенным за пределами этой AS и не поддерживающим OSPF.

Внешние извещения описывают не поддерживающие OSPF маршруты за пределами автономной системы (домена маршрутизации OSPF). Их передают только маршрутизаторы ASBR, то есть маршрутизаторы, находящиеся на границах автономной системы и поддерживающие более чем один протокол маршрутизации, например, OSPF и RIP, IGRP, EIGRP, статическую маршрутизацию и т.д. Внешние LSA бывают двух видов: типа 5 и типа 7. LSA типа 5 позволяют изве-

щать о внешних маршрутах автономной системы все области и маршрутизаторы OSPF. LSA типа 7 передают только маршрутизаторы ASBR, соединенные с областями NSSA (Not So Stubby Area). В таких областях поддерживается более чем один протокол, например, OSPF и RIP, IGRP, EIGRP, статическая маршрутизация и т.д. Маршрутизаторы ASBR можно конфигурировать так, чтобы они принимали сведения о маршрутах, не поддерживающих OSPF, и внедряли их в среду OSPF NSSA посредством LSA типа 7, используемых для передачи маршрутной информации через тупиковые области.

Непрозрачные извещения LSA типов 9-11 описаны в RFC 2370. Они могут быть использованы для распространения информации, необходимой для управления трафиком MPLS, такой как сведения о емкости и топологии маршрутов. Поддержка непрозрачных LSA в маршрутизаторах OSPF не является обязательной, но многие ведущие производители реализуют эту функцию в своей продукции. LSA типа 9 может передаваться в пределах локальной сети или подсети. LSA типа 10 передается в пределах области OSPF, а LSA типа 11 может передаваться в пределах всей AS.

Выше уже объяснялось, с какой целью маршрут сетей OSPF передает *LSA* ко всем другим маршрутизаторам в своей области. Заголовки всех LSA имеют одинаковый формат и содержат одни и те же поля. Длина заголовка составляет 20 байтов. Формат заголовка LSA протокола OSPF показан на рис. 5.9.

Рис. 5.9. Структура заголовка LSA

Заголовок LSA содержит 8 полей:

Возраст LSA (LS Age) — указывает время, в секундах, которое прошло с момента передачи LSA инициирующим маршрутизатором.

Опции (Options) — содержит то же значение опции, что и пакет-приветствие.

Тип LSA (LS Type) — определяет тип LSA: внутри области (1 или 2), межобластное (3 или 4), внешнее (5 или 7) или непрозрачное (9-11).

Идентификатор извещения (Link State ID) указывает, к каким каналам оно относится — маршрутизатора (LSA type 1 и 4), или межсетевым (LSA type 2, 3, 5 или 7).

Извещающий маршрутизатор (Advertising Router) идентифицирует маршрутизатор, передавший данное извещение.

Порядковый номер LSA (LS Sequence Number) гарантирует доставку и получение DD-пакетов описания базы данных, рассмотренных в 5.5.2, в той очередности, в какой они передавались. Каждый раз при отправке LSA маршрутизатор, передающий извещение, включает в заголовок порядковый номер пакета. Получатель использует этот номер для контроля порядка следования принимаемых DD-пакетов.

Контрольная сумма (LS Checksum) служит для проверки OSPF DD-пакета на предмет отсутствия его искажения. Маршрутизатор-отправитель вычисляет эту сумму по специальному алгоритму и помещает результат в поле LS Checksum. Получатель вычисляет по тому же алгоритму контрольную сумму для полученного пакета и сравнивает результаты. Если обнаруживается, что суммы не совпадают, получатель отбрасывает дейтаграмму, так как ее содержимое при пересылке изменилось. Если суммы совпадают, то считается, что заголовок не поврежден, и получатель обрабатывает дейтаграмму.

Длина (Length) — в этом поле указывается длина OSPF-пакета в байтах.

В непрозрачных извещениях на месте поля Link State ID находятся поле Opaque type (8 битов) и поле Opaque ID (остальные 24 бита). Поле Opaque ID идентично полю Link State ID, а поле Opaque type определяет тип Opaque LSA (совсем не то, что тип LSA), причем использование номеров 0-127 назначается рабочей группой IETF, а номера 128-255 резервированы для частного и экспериментального использования. На сегодня определены только три типа Opaque: 1 — для задач управления трафиком, 2 — для Sycamore Optical Topology Descriptions и 3 — для Grace LSA.

5.8. Базы данных OSPF

Все маршрутизаторы OSPF создают и поддерживают три отдельные базы данных: смежности (Adjacency), состояния каналов (Link State) и пересылки (Forwarding).

5.8.1. База данных о смежности

База данных Adjacency, неформально, но весьма удачно именуемая *таблицей соседей*, используется маршрутизатором LSR для хранения информации о соседних LSR в том же домене сети.

С начала работы маршрутизатора LSR в сети (при первоначальном включении, рестарте, изменении конфигурации по команде администратора и т.п.) он рассылает пакеты-приветствия по групповому адресу 224.0.0.5, чтобы представить себя соседним маршрутизаторам домена (также поддерживающим протокол OSPF), которые, в свою очередь, добавляют полученные данные о новом маршрутизаторе в свои таблицы соседей и отвечают своими пакетами-приветствиями, идентифицируя себя.

При знакомстве с OSPF часто возникает путаница между соседними и смежными маршрутизаторами. Здесь можно провести следующую аналогию. Все без исключения сотрудники лаборатории, работающие в общей комнате, являются соседями и общаются между собой. У некоторых из них (а возможно, что и у всех) соседские отношения перерастают в дружеские, предполагающие более тесное общение, более интенсивный обмен информацией. Применительно к маршрутизаторам можно сказать, что в таком случае *соседство* переходит в *смежность*, а для этого все параметры в пакетах-приветствиях должны соответствовать требованиям соседей и приниматься ими. В противном случае базу данных о смежности маршрутизаторы не сформируют.

5.8.2. Топологическая карта сети

База данных со сведениями о состоянии каналов, также имеющая и другое, более точное название *топологическая карта*, содержит сведения обо всех сетях, подсетях и пунктах назначения в пределах области OSPF, представляя собой топологическую карту сети. После формирования таблицы смежности в *процессе обнаружения соседей (neighbor discovery)*, т.е. когда маршрутизаторы OSPF «познакомятся» друг с другом и начнут обмениваться информацией, они переходят к созданию баз данных о состоянии каналов — топологической карты сети для своей области OSPF. В топологической карте указываются каждая сеть и подсеть, а также маршруты к этим сетям и подсетям. Используя топологическую базу данных, каждый маршрутизатор создает древовидную структуру, в которой определяет себя как корень дерева, соединенный с каждым другим маршрутизатором кратчайшим путем.

5.8.3. Таблица маршрутизации

Третья база, называемая *базой пересылки,* которую, по почти сложившейся в этом параграфе традиции, мы будем именовать *таблицей маршрутизации*, хранит наилучшие маршруты к сетям, подсетям и пунктам назначения OSPF. Эта база формируется на основе топологической карты и действительно является таблицей маршрутизации.

Каждый маршрутизатор использует информацию, содержащуюся в топологической базе данных о состоянии каналов, для формирования базы пересылки (таблицы маршрутизации). Сформировав топологическую базу, маршрутизатор запускает алгоритм Дийкстры для вычисления кратчайших маршрутов ко всем известным маршрутизаторам. Затем данные этих маршрутов помешаются в таблицу маршрутизации. Точно так же, когда маршрутизатор обнаруживает новый маршрут или неисправность в ранее зафиксированном маршруте, он немедленно, не ожидая срабатывания таймера, передает информацию об этом всем маршрутизаторам, принадлежащим данному домену. При этом извещения о корректировке, содержащие маршрутную информацию, рассылаются лавинным способом всем маршрутизаторам внутри своей области, уведомляя их о произошедших изменениях. Если область содержит больше пятидесяти маршрутизаторов, извещения о состоянии каналов, лавинообразно распространяющиеся по сети, могут вызвать значительные перегрузки и задержки. Эту неприятность можно преодолеть путем разделения автономной системы OSPF, состоящей из одной области, на несколько областей OSPF, о чем говорилось в 5.4.

5.9. Принципы работы OSPF

Теперь, обладая необходимыми сведениями относительно OSPF-областей, алгоритма SPF, структур OSPF-пакетов, баз данных и LSA-извещений, подведем итоги основных этапов маршрутизации по протоколу OSPF, сформулированных в начале главы.

В качестве первого этапа отмечалось распространение информации о топологии сети, инициирование отношений соседства и смежности. После инициирования каждый OSPF-маршрутизатор начинает обмен LSA, передает пакеты HELLO через все свои интерфейсы, распространяя эту информацию по всем соседним маршрутизаторам, так что каждый из них узнает идентификаторы своих ближайших соседей. Эта топологическая информация начинает распространяться по сети от соседа к соседу, пополняя топологические карты в маршрутизаторах LSR новыми данными, и через некоторое время достигает самых удаленных маршрутизаторов. В результате все маршрутизаторы сети получают сведения о графе сети, которые хранятся в их топологических базах данных.

Впоследствии, при появлении новой связи или нового соседа, маршрутизатор узнает об этом из новых пакетов HELLO. В них указывается достаточно детальная информация о том маршрутизаторе, который передал этот пакет, а также о его ближайших соседях, так что этот маршрутизатор можно однозначно идентифицировать.

Таким образом, на первом шаге каждый маршрутизатор OSPF строит граф связей сети, в котором вершинами графа являются маршрутизаторы, а ребрами — каналы, включенные в интерфейсы маршрутизаторов. Для построения этого графа все маршрутизаторы обмениваются со своими соседями той информацией о сети, которой они располагают в данный момент.

Этот процесс похож на процесс распространения векторов расстояний в протоколе RIP, однако сама информация здесь качественно другая — это информация о топологии сети, передаваемая, к тому же, без модификации, которую делают RIP-маршрутизаторы, а в неизменном виде. В результате распространения топологической информации все маршрутизаторы сети получают идентичные сведения о графе сети, которые хранятся в топологической базе данных каждого маршрутизатора.

Далее, с помощью полученного графа находятся оптимальные маршруты. В протоколе OSPF для этого используется итеративный алгоритм Дийкстры, причем каждый маршрутизатор считает себя центром сети и ищет оптимальный маршрут к каждому известному ему маршрутизатору (или к известной ему сети). В любом найденном таким образом маршруте, в соответствии с принципом одношаговой маршрутизации, запоминается только первый шаг — к следующему маршрутизатору. Если несколько маршрутов к сети назначения имеют одинаковую метрику, то в таблице маршрутизации запоминаются первые шаги всех этих маршрутов. Данные об этих шагах и попадают в рассмотренную в п. 5.8.3 таблицу маршрутизации.

Благодаря иерархической структуре областей уменьшаются перегрузки, связанные с поддержкой огромных таблиц маршрутизации и с пересчетом этих таблиц при изменениях маршрутов. Извещения о корректировках передаются только в случае, если в сети происходят изменения. Эти извещения рассылаются всем маршрутизаторам OSPF, что сокращает время сходимости.

Количественная оценка числа итераций для перехода OSPF-сети в установившееся состояние соответствует величине $L \cdot log_2 L$, где L — количество каналов (ребер). Эта оценка явно превосходит аналогичный показатель использующего дистанционно-векторный алгоритм Беллмана-Форда протокола RIP, равный $N \cdot L$, где N — число маршрутизаторов.

Еще один аспект функционирования OSPF связан со *стабильностью.* В OSPF-сетях могут возникать периоды нестабильной работы. Например, при отказе связи, когда информация об этом не дошла до какого-либо маршрутизатора, он продолжает отправлять пакеты

к сети назначения, считая эту связь работоспособной. Однако такие периоды непродолжительны, и пакеты не зацикливаются в маршрутных петлях, а просто отбрасываются при невозможности передать их через неработоспособную связь. Каждая запись в топологической базе данных имеет срок жизни, как и маршрутные записи протокола RIP. С каждой записью связан таймер, который служит для контроля времени жизни записи. Если какая-либо запись в топологической базе маршрутизатора, полученная от другого маршрутизатора, устаревает, то первый из них может запросить ее новую копию с помощью специального сообщения Link-State Request протокола OSPF, на которое должен поступить ответ Link-State Update от маршрутизатора, непосредственно тестирующего эту связь, о чем говорилось в параграфе 5.5.

5.10. SDL-диаграмма поведения маршрутизатора OSPF

В заключение главы, по сложившейся для книг этой серии традиции, построим SDL-диаграмму поведения OSPF-маршрутизатора, отображающую вышеизложенные принципы (рис. 5.10). Для читателя, не знакомого с основами SDL, рекомендуем пролистать главу 2 из книги «Сигнализация в сетях связи».

Итак, прежде чем начать обмен маршрутной информацией, маршрутизаторы должны сформировать таблицу соседей, последовательно проходя через следующие состояния:

- Нерабочее (Down)
- Инициирование (Init)
- Двусторонний обмен данными (Two-Way)
- Выбор DR и BDR (Exstart)
- Обмен (Exchange)
- Загрузка (Loading)
- Полная готовность (Full)

Нерабочее состояние 0 (Down) — состояние маршрутизатора перед включением его администратором. При включении, если маршрутизатор исправен и поддерживает протокол OSPF, он переходит в *состояние 1 инициирования (Init)*, когда администратор активизирует протокол OSPF или устанавливает интерфейс в исходное состояние.

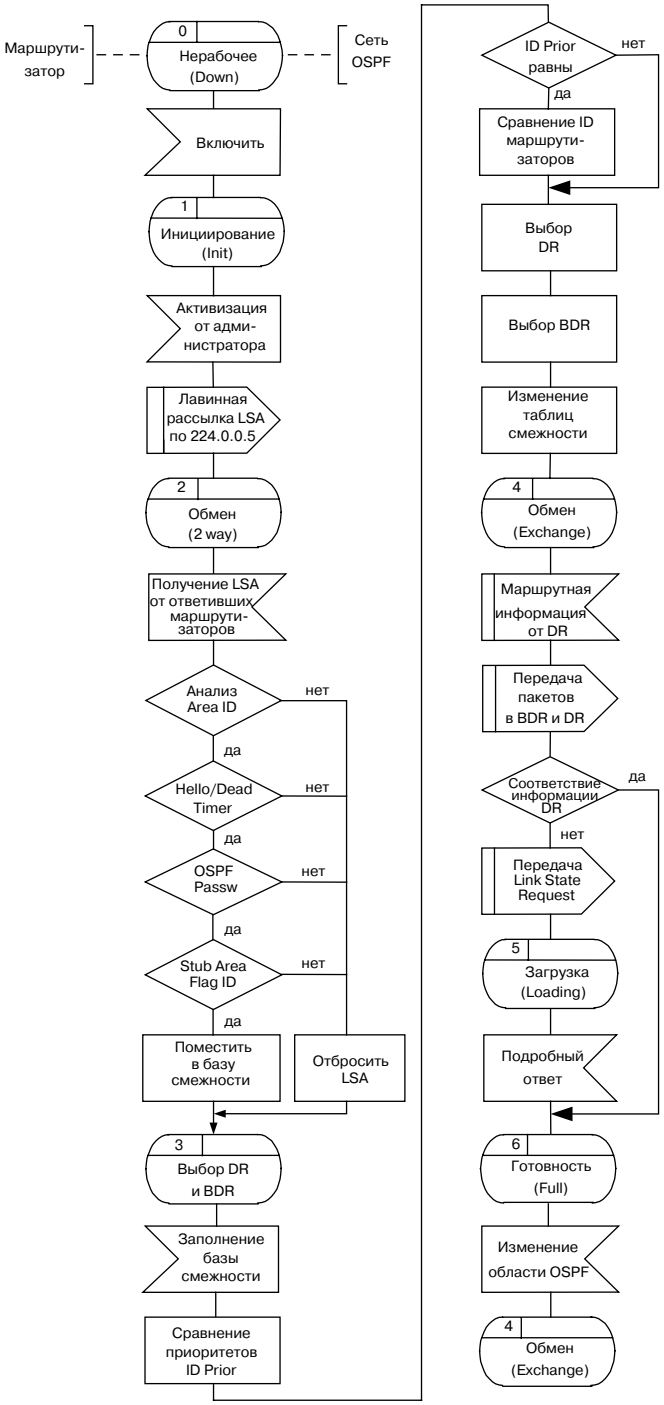


Рис. 5.10. SDL-диаграмма поведения маршрутизатора OSPF

В *состоянии 1* маршрутизатор извещает о себе всех своих соседей, рассылая по групповому адресу 224.0.0.5 пакеты-приветствия, в которых он представляет себя и свои параметры всем остальным маршрутизаторам. В этом состоянии между соседями еще не существует отношений смежности. Точно так же, не существует назначенных DR и резервных назначенных маршрутизаторов BDR, так как еще не выполнен процесс их выбора, а только переданы извещения о присутствии маршрутизатора всем остальным OSPF-маршрутизаторам данного сегмента. Последние получают эти извещения, из которых узнают о новом соседе, и затем включают сведения о нем в свои базы данных о смежности.

Передав все эти извещения, маршрутизатор переходит в *состояние 2 двустороннего обмена (Two-Way),* получает ответные извещения от других локальных маршрутизаторов и начинает формировать смежные соседские взаимоотношения. Отношения смежности могут формироваться только в случае совпадения следующих полей в пакетах-приветствиях:

- ID зоны (OSPF Area ID).
- Значение таймеров приветствия и «жизни» маршрутизатора (Hello/Dead timers).
- Аутентификация — обмен паролями (OSPF Password).
- ID тупиковой области (Stub Area Flag ID).

После того как маршрутизаторы сформируют смежные взаимоотношения с соседними маршрутизаторами, они будут передавать пакеты-приветствия через установленные интервалы времени (по умолчанию — через 10 секунд) для поддержки смежности. Если маршрутизатор не получит от своего соседа пакета-приветствия в течение четырех таких интервалов, он сочтет соседний маршрутизатор неработающим («мертвым» — dead) и удалит его из базы данных о смежности.

После установления двухсторонних взаимоотношений все маршрутизаторы сегмента обладают достаточной информацией для выбора назначенного маршрутизатора (DR — Designated Router) и резервного назначенного маршрутизатора (BDR — Backup Designated Router). Происходит переход в *состояние 3 выбора DR и BDR*. В этом состоянии все маршрутизаторы формируют отношения ведущий/ведомый и выбирают маршрутизаторы DR и BDR с целью обмена маршрутной информацией. После выбора DR и BDR они становятся в сегменте центральным узлом сбора всех сообщений о корректировках, а все остальные маршрутизаторы становятся по отношению к DR и BDR ведомыми. В процессе выбора участвуют все маршрутизаторы сегмента. Для выбора DR и BDR маршрутизаторы используют два параметра, заимствованных из предыдущих пакетов-приветствий: ID приоритета (рассматривается в первую

очередь) и/или ID маршрутизатора (рассматривается во вторую очередь). Маршрутизатор с наивысшим приоритетом становится DR, маршрутизатор со следующим после него приоритетом становится BDR. Если же все маршрутизаторы имеют одинаковый приоритет, то маршрутизатором DR становится маршрутизатор с наивысшим идентификационным номером, а маршрутизатор со следующим по величине ID становится BDR.

После того как маршрутизаторы DR и BDR выбраны, они ведут сбор всех извещений о маршрутах, получаемых от локальных маршрутизаторов, создают топологические базы данных (Link State database), распространяют эту информацию по всем маршрутизаторам сегмента. Ведомые маршрутизаторы изменяют свои смежные взаимоотношения со всеми другими маршрутизаторами таким образом, что маршрутизаторы DR и BDR становятся получателями всех извещений о маршрутах.

С момента своего существования маршрутизаторы DR и BDR принимают и обрабатывают локальные извещения о маршрутах типа 1, поступающие от всех остальных маршрутизаторов сегмента. Однако только маршрутизатор DR несет ответственность за распространение этой информации среди локальных маршрутизаторов; он рассылает ее по групповому адресу 224.0.0.5. Маршрутизатор BDR остается в резерве до тех пор, пока DR не окажется неспособным выполнять вышеназванные функции. Тогда маршрутизатор BDR становится DR. Когда же вышедший из строя DR снова становится работоспособным, он выполняет функции BDR.

Принцип выбора маршрутизатора DR для каждого сегмента позволяет уменьшить число смежных связей, необходимых во всей объединенной сети, и сокращает время обработки OSPF-трафика, обусловленного рассылкой групповых извещений всеми маршрутизаторами.

Завершив выбор DR и BDR, маршрутизатор переходит в *состояние 4 обмена (Exchange).* В этом состоянии все подчиненные маршрутизаторы обмениваются маршрутной информацией с маршрутизаторами DR и BDR. Они передают извещения о корректировках обоим этим маршрутизаторам, используя групповой адрес 224.0.0.6. И DR, и BDR воспринимают изменения базы данных, но, как уже было упомянуто, только DR управляет синхронизацией и распространением этой информации. На этом этапе маршрутизаторы создают топологические базы данных, содержащие описание всех маршрутизаторов и сетей внутри области OSPF, как это изложено в параграфе 5.8.

Если маршрутизатор получает от DR информацию, которая противоречит текущей топологической карте, то чтобы получить информацию, необходимую для построения новой топологической

карты, маршрутизатор передает запрос данных о состоянии каналов (Link State Request), переходит *в состояние 5 загрузки (Loading)* и запрашивает у локального DR более подробные сведения. Получив необходимые сведения, он запускает алгоритм Дийкстры для создания таблицы маршрутизации.

Если же никаких расхождений нет, маршрутизатор пропускает этот этап и сразу переходит к созданию таблицы маршрутизации на основании базы данных о состоянии каналов. Маршрутизатор вычисляет наилучшие маршруты к пунктам назначения, вводит их в базу пересылки (таблицу маршрутизации) по алгоритму Дийкстры на основе топологической карты маршрутов и переходит в *состояние 6 полной готовности (Full).*

Маршрутизаторы OSPF выполняют описанные действия всякий раз, когда в области OSPF происходят изменения. При обнаружении изменений, например, ввода нового канала или неисправности в существующем канале, маршрутизатор OSPF генерирует извещение о корректировке (входит в состояние 4 обмена) и лавинным способом рассылает его в пределах области. Маршрутизаторы DR всех сегментов получают это извещение, обрабатывают его и распространяют информацию каждый в своем сегменте.

Этим исчерпывается описание протокола OSPF в нашей книге. В двух следующих главах будет рассмотрена роль двух других популярных протоколов маршрутизации IS-IS и BGP, но перед этим хотелось бы подчеркнуть, что основную роль в MPLS играет маршрутизация именно с помощью протокола OSPF.

Глава 6
Протокол IS-IS

6.1. Еще раз о маршрутизации по состоянию каналов

Протокол маршрутизации OSI под названием *протокол обмена данными между промежуточными системами IS-IS (Integrated Intermediate System-to-Intermediate System)* использует тот же принцип *маршрутизации по состоянию каналов*, что и рассмотренный в предыдущей главе протокол OSPF. Но если OSPF является разработкой IETF, то протокол IS-IS был создан ISO (International Standard Organization) на базе сетевой архитектуры DECNet Phase V для стыковки с протоколом *CLNP (Connectionless Network Protocol)*, являющимся основным сетевым протоколом уровня 3 в разработанной ISO *архитектуре CLNS (Connectionless Network Service).*

Как раз в терминологии ISO маршрутизаторы называются «промежуточными системами» (intermediate system, IS), а хосты — «конечными системами» (end system, ES). Существует также протокол ES-IS, с помощью которого маршрутизаторы узнают о подключенных к ним хостах, а хосты — о маршрутизаторах. Протокол ES-IS реализуется Hello-сообщениями ESH, периодически передаваемыми хостами, и Hello-сообщениями ISH, периодически передаваемыми маршрутизаторами (в ЛВС сообщения передаются на специально выделенные групповые адреса Ethernet).

Разработка протокола IS-IS в ISO оказалась весьма успешной, и этот протокол, обеспечивая, в частности, соответствие между двумя популярными архитектурами — неоднократно упоминавшейся на страницах этой книги *семиуровневой моделью OSI* и разработанной компанией Digital Equipment и реализованной в DECNet *сетевой архитектурой DNA (Digital Network Architecture),* занял свое место в той и другой архитектуре. Предшественником протокола IS-IS явился протокол маршрутизации DECNet под названием *DRP (DECNet Routing Protocol)*, входивший в состав пакета протоколов DECNet Phase IV.

Оказалось, что протокол IS-IS очень хорошо работает в весьма больших сетях, содержащих более 500 маршрутизаторов. Подобно

OSPF, протокол IS-IS разделяет сеть на области, чтобы не распространять информацию о маршрутах среди всех маршрутизаторов сети, обеспечивая разумные размеры их таблиц маршрутизации, а тем самым — быструю сходимость поиска маршрута. По этим причинам, в частности, протокол IS-IS стандартизирован и в рамках IETF в RFC 1142 «OSI IS-IS Intra-domain Routing Protocol» (сам протокол) и в RFC 1195 «Use of OSI IS-IS for routing in TCP/IP and dual environments» (использование этого протокола для маршрутизации в сетях TCP/IP и в двухпротокольной среде).

В технологии MPLS протокол IS-IS используется почти таким же образом, как и рассмотренный в главе 5 протокол OSPF. Оба они относятся к классу протоколов IGP и служат для создания и поддержки топологической карты, используемой протоколом LDP.

Ранее уже отмечалось, что классификация протоколов маршрутизации делит их на категории в зависимости от используемых ими алгоритмов поиска оптимального маршрута. Одна из общих категорий получила в этой классификации обозначение *маршрутизации по состоянию каналов*. Отметим, что практически каждый маршрутизатор, представленный на современном рынке сетевого оборудования, умеет обращаться, как минимум, с одним из многих существующих протоколов динамической маршрутизации по состоянию каналов.

Протоколы динамической маршрутизации используются в LSR с целью идентифицировать соседние LSR одного сетевого домена и периодически обновлять информацию о топологии сети при помощи рассмотренных в предыдущей главе *извещений о состоянии каналов***.** Этими извещениями маршрутизаторы периодически обмениваются для того, чтобы знать состояние соседних маршрутизаторов (и их каналов) на текущий момент времени. Такой периодический обмен обладает своими преимуществами, равно как и недостатками, однако прежде чем перейти к их подробному обсуждению, еще раз отметим, что именно отличает протоколы маршрутизации по состоянию каналов OSPF и IS-IS от протоколов статической маршрутизации типа RIP.

Главную отличительную черту протоколов маршрутизации по состоянию каналов составляет постоянно проводящийся поиск кратчайшего пути. Это является и главным преимуществом использования такой маршрутизации, т.к. для передачи данных между двумя конечными пунктами используется кратчайший *на данный момент* маршрут.

Как часто бывает в жизни, названное преимущество протокола IS-IS является в то же время и его существенным недостатком. Этот недостаток связан с так называемой *лавинной рассылкой пакетов (flooding)*, вызываемой внезапным изменением состояния каналов

(либо канал неожиданно стал недоступен, либо, наоборот, возобновил свою работу после перерыва). *Flooding* характеризуется обменом между маршрутизаторами огромным количеством служебных пакетов, т.к. каждый маршрутизатор, соседний с данным, приняв очередное извещение об изменении состояния каналов и обновив свои таблицы маршрутизации, пересылает его дальше.

6.2. Проблема flooding в протоколе IS-IS

В IS-IS присутствуют аналогичные OSPF механизмы обнаружения соседей с помощью пакетов Hello, синхронизации баз данных и оповещения об изменении состояния связи путем рассылки пакетов (flooding). Аналогами OSPF-пакетов LSU (Link State Update) в IS-IS являются рассматриваемые ниже пакеты *LSP (Link State Packet)*.

Подчеркнем весьма неудачную для книги ситуацию, что LSP означает коммутируемый по меткам тракт, и пакет протокола IS-IS. Впрочем, второй вариант аббревиатуры используется только в главе 6.

Лавинная рассылка этих LSP-пакетов радикально изменяет требования к сетевой пропускной способности с учетом размера сети и интенсивности работы в ней протокола IS-IS. Необходим запас пропускной способности, достаточной для того, чтобы справиться с лавинной рассылкой, и должны выполняться соответствующие требованиях к аппаратным ресурсам маршрутизаторов, к быстродействию их процессоров и к объему памяти.

Это связано с возможностью передачи между маршрутизаторами сети большого количества служебных сообщений, мешающих нормальной передаче полезных данных, ради чего, собственно, и создана сеть. В данном случае этот дополнительный трафик обусловлен тем, что каждый маршрутизатор сети обязан оповестить о произошедших изменениях своих соседей, а те, в свою очередь, должны оповестить своих, и так до тех пор, пока все маршрутизаторы не синхронизируют свои данные о потенциально доступных маршрутах.

Для предотвращения возможных перегрузок при лавинной рассылке LSP-пакетов протокол IS-IS оснащен рядом встроенных механизмов контроля.

Один такой механизм получил весьма романтичное название *расщепление горизонта (Split Horizon).* Заключается этот механизм в том, что маршрутизатор никогда не передаст LSP-пакет по тому каналу, по которому тот был принят. Такое ограничение исключает зацикленную передачу пакета между двумя маршрутизаторами. Проиллюстрируем этот механизм все на том же примере сети MPLS, представленном здесь на рис. 6.1.

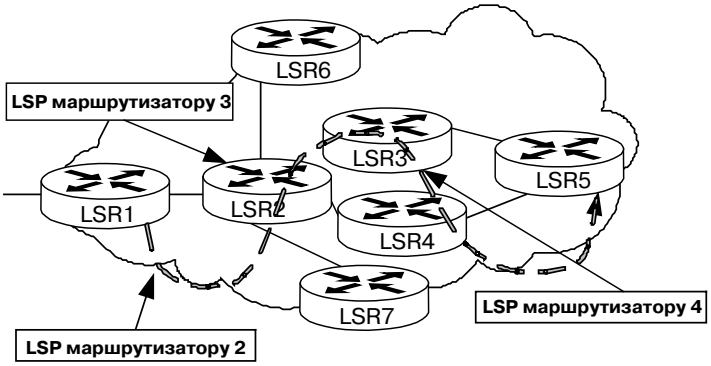


Рис. 6.1. Пример фрагмента MPLS-сети

После получения маршрутизатором LSR2 пакета LSP, переданного от LSR1, этот второй маршрутизатор обновляет информацию в своей таблице маршрутизации и передает собственное извещение маршрутизатору LSR3. При этом отсутствует целесообразность передачи LSP-пакета от LSR2 к LSR1, т.к. именно LSR1 начал процесс обмена служебной информацией и, естественно, не нуждается в приеме той информации, которую сформировал сам относительно своих собственных каналов.

Точно таким же образом маршрутизатор LSR3 блокирует впоследствии канал между собой и маршрутизатором LSR2. И следующий маршрутизатор LSR4, получивший служебную информацию по цепочке от маршрутизатора LSR3, точно так же блокирует порт между собой и LSR3. При помощи такого блокирования канала после получения по нему служебной информации общее количество служебной информации, передаваемой для обновления таблиц маршрутизации, сводится к минимуму.

Обратим внимание, что на рис. 6.1 маршрутизатор LSR4 может получить одну и ту же информацию об обновлении сети на два свои порта: переданную от LSR2 и от LSR3. Целесообразно, разумеется, в такой ситуации обрабатывать только одно из дублирующих друг друга извещений. Но неверно было бы блокировать только один канал (например, от LSR2 к LSR4) и выполнить дальнейшую пересылку LSP-пакета с данными об обновлении от LSR4 к LSR3. Разумно обрабатывать только один из дублирующих друг друга LSP-пакетов и блокировать все каналы, по которым они поступили и по которым бессмысленно передавать их обратно. Этому способствуют два специальных поля в заголовке LSP-пакета: *время жизни (Life Remaining)* и *очередность (Sequence).* Первое поле содержит значение таймера задержки, запускаемого маршрутизатором в процессе обработки LSP-пакета и до срабатывания которого обработка каких-либо иных LSP-пакетов в этом LSR и по этому ка-

налу не производится. Второе поле используется для определения очередности обработки содержащейся в LSP-пакете информации и для избежания обработки пакета, пришедшего вне очереди.

6.3. Метрики IS-IS

Основная метрика, используемая в IS-IS,— это некоторое число, не превышающее 1024 для маршрута и 64 — для канала. Смысл и числовые значения этой метрики для каждого канала и маршрута определяет системный администратор. Метрика маршрута вычисляется как сумма метрик составляющих его каналов.

Кроме того, можно задать три дополнительные метрики: *задержка* (*delay*), отражающая длительность задержки в канале, *стоимость передачи по каналу* (*expense*), отражающая коммуникационные затраты, и *ошибки* (*error*), отражающая коэффициент ошибок в канале.

IS-IS поддерживает возможность задавать в поле QoS заголовка пакета CLNP соотношение между этими четырьмя метриками, что будет рассмотрено ниже. По умолчанию IS-IS-метрика любого интерфейса, включенного в IS-IS-систему на маршрутизаторах Cisco, равна 10. Метрика связи с внешней IP-сетью равна 0 и, как и обычные метрики, увеличивается на 10 с каждым промежуточным маршрутизатором. Под внешними IP-сетями понимаются сети, включенные в IS-IS-систему командой passive-interface без активизации протокола IS-IS в соответствующем интерфейсе. Чтобы изменить метрику интерфейса, дается команда router(config-if)#isis metric M [level-1 | level-2], где метрика M может принимать значения от 0 до 63. По умолчанию метрика изменяется для связей как на уровне 1, так и на уровне 2. Если необходимо изменить метрику связей только какого-то одного уровня, то этот уровень указывается в конце команды. Если требуется изменить метрику сети, включенной в базу данных командой passive-interface, необходимо сначала активизировать IS-IS в соответствующем интерфейсе, а потом дать команду isis metric. Ранее данная команда passive-interface предотвратит передачу сообщений IS-IS и установление отношений смежности через интерфейс.

6.4. Адресация IS-IS

Напомним, что обычные OSI-протоколы используют так называемые NSAP-адреса точек доступа к сетевой услуге (Network Service Access Point). Эта форма адреса точно определяет местоположение конечной системы относительно области, которой принадлежит искомый узел. А так как IS-IS — это протокол, совместимый с OSI,

то в нем тоже используются NSAP-адреса. Но, в то же время, IS-IS интерпретирует содержимое NSAP-адреса несколько иначе, чем это происходит в прочих OSI-протоколах.

Формат адреса устройства стандартен для многих сетевых протоколов, и NSAP-адреса не составляют исключения. Подобно большинству сетевых адресов, NSAP-адреса разделены на две части с целью более точно указать местоположение узла на сетевом уровне. Первая, *доменовая часть IDP (Initial Domain Part),* — это составляющая адреса, соответствующая домену, которому принадлежит узел. Сама эта составляющая тоже содержит две части: AFI и IDI. Первая из них, *идентификатор формата и полномочий AFT* (Authority and Format Identifier), определяет полномочия, присвоенные адресом домену, а вторая, идентификатор начального *домена IDI* (Initial Domain Identifier), определяет общую информацию о домене.

Вторая, *специфическая часть NSAP-адреса DSP* (Domain Specific Part), содержит адрес узла внутри домена, которому он принадлежит.

Адрес области, заключающий в себе IDP, однозначно определяет область, которой принадлежит узел-адресат. Все узлы, расположенные в одной и той же области, обладают одинаковым адресом области. Администратор сети присваивает области этот адрес на этапе разработки сети в виде, по меньшей мере, одного шестнадцатеричного октета.

6.5. Маршрутизация IS-IS

Принципы маршрутизации IS-IS очень похожи на используемые в протоколе OSPF. Для синхронизации баз данных маршрутизации IS-IS использует пакеты CSNP (Complete Sequence Number Packet) и PSNP (Partial Sequence Number Packet), по своему назначению примерно аналогичные пакетам DD (Database Description) и LSR (Link State Request) протокола OSPF. Протокол IS-IS поддерживает и двухуровневую иерархическую систему сетей (периферийные области и магистральная область 0), но принцип организации этой системы отличается от принципа ее организации в OSPF.

Иерархическая двухуровневая маршрутизация в IS-IS-системе строится следующим образом. Периферийные области образуют *уровень 1*, а магистраль (backbone) — *уровень 2*. В отличие от OSPF, каждый IS-IS-маршрутизатор принадлежит области целиком (то есть граница между областями проходит не внутри маршрутизатора, а через соединения между маршрутизаторами).

Номер области содержится в NSAP-адресе маршрутизатора. Маршрутизатор может принадлежать только одной области. Он может быть маршрутизатором первого уровня, или маршрутизатором

второго уровня, или и то, и другое одновременно (L1-2). Признаком маршрутизатора второго уровня является специальный бит (attached bit), который такой маршрутизатор вводит в пакеты LSP, рассылаемые им внутри области.

Маршрутизатор первого уровня поддерживает полную базу данных о состоянии связей в своей области (L1-database) и, соответственно, хранит полный набор данных о маршрутах внутри нее. Такой маршрутизатор не имеет никаких сведений о сетях, расположенных за пределами его области. При необходимости отправить пакет за пределы области он пересылает этот пакет ближайшему маршрутизатору 2-го уровня, находящемуся в его области. Т.о., область IS-IS концептуально аналогична тупиковой области OSPF.

Маршрутизаторы второго уровня образуют магистраль. Она не является отдельной областью, а представляет собой совокупность маршрутизаторов второго уровня, каждый из которых находится в своей области. В одной области могут находиться несколько маршрутизаторов второго уровня. Маршрутизатор 2-го уровня может быть единственным маршрутизатором области (область из одного маршрутизатора). Магистраль должна быть связной, то есть каждый маршрутизатор 2-го уровня должен быть непосредственно связан с одним или несколькими маршрутизаторами этого же уровня. Маршрутизаторы 2-го уровня поддерживают базу данных о состоянии связей магистрали (L2 database).

Маршрутизатор 2-го уровня вносит в базу данных магистрали список узлов, находящихся в его области. Получив пакет, адресованный в другую область, такой маршрутизатор находит в базе данных магистрали ближайший маршрутизатор этого же уровня, находящийся в области адресата, и отправляет ему полученный пакет. Чтобы получить информацию о префиксах своей области от маршрутизаторов первого уровня, маршрутизатор второго уровня должен поддерживать и базу данных L1. Следовательно, в каждой области должен быть хотя бы один маршрутизатор L1-2.

Назначенная промежуточная система, или, другими словами, *назначенный маршрутизатор (Designated Router),* полностью аналогична тому DR, который выбирается в OSPF. Назначенная промежуточная система исполняет роль инициатора всех сетевых событий. Основной функцией этой системы является рассылка LSP-пакетов, включающих в себя информацию для всей сети. Система генерирует полную картину сетевого окружения на основе сбора обновленных сведений о состоянии каналов и рассылает эту информацию всем прочим промежуточным системам сети.

Когда промежуточные системы IS получают LSP-пакеты от назначенного маршрутизатора, происходит сравнение всей LSP-информации в локальных базах данных маршрутизаторов 1-го и 2-го

уровней с принятыми данными, что обеспечивает быструю сходимость. Выбор же назначенной промежуточной системы происходит с учетом сравнения приоритетов разных IS сети.

После того как все IS сравнили свои приоритеты, система с наивысшим приоритетом становится назначенным маршрутизатором. В случае, когда наивысший приоритет имеют сразу несколько IS, проблема решается сравнением их MAC-адресов. Назначенной становится система с наибольшим значением MAC-адреса. После завершения выбора система, которая получила статус назначенной, занимается сбором и распространением LSP-информации по всей локальной сети, известной в этом аспекте как *псевдо-узел.*

Если одна IS обладает интерфейсами, покрывающими пространство нескольких локальных сетей, то такая IS будет участвовать в выборе назначенной системы во всех этих сетях. Другими словами, выбор происходит в локальной сети, к которой физически подключена данная система. И если система принадлежит более чем одной локальной сети, то автоматически становится во всех этих сетях кандидатом на роль назначенной и может получить статус назначенной промежуточной системы сразу в нескольких сетях.

Если необходимо исключить возможность выбора той или иной системы в качестве назначенной, нужно присвоить ей приоритет, равный нулю. Вообще же приоритет может иметь любое значение от 0 до 127.

Итак, с точки зрения назначенной промежуточной системы множество прочих IS сети представляет собой псевдо-узел, который рассматривается ею как единый логический объект. Протокол IS-IS использует псевдо-узлы для получения информации о сети в целом.

Каждая из IS, входящих в состав псевдо-узла, получает LSP-информацию только от одной назначенной системы. Исключение составляет только лавинная информация, поступающая от тех устройств, с которыми эта IS имеет прямые связи. Но такая LSP-информация касается состояния каналов лишь отдельно взятой промежуточной системы, а не псевдо-узла в целом. Об этом уже говорилось при рассмотрении лавинной рассылки на рис. 6.1 и отмечалось, что если каждый маршрутизатор будет передавать всем другим маршрутизаторам в сети данные об обновлении, то может произойти бесконечное зацикливание служебной информацией. Концепция псевдо-узла разрешает эту проблему.

Работа назначенной системы (так же, как и назначенного маршрутизатора в OSPF) заключается в сборе всей служебной информации и в создании главной таблицы маршрутизации с учетом полной топологии сети. В дальнейшем назначенная система передает данные об обновлении на адрес псевдо-узла, а каждая из систем,

входящих в этот псевдо-узел, рассматривает получаемые данные с точки зрения себя самой и своих непосредственных соседей. Этот механизм обеспечивает доставку сведений об обновлении всем системам, гарантируя при этом отсутствие зацикливания информации.

Каждой IS присваивается уникальный адрес, отличающий ее от прочих. И в то же время, для адресации к совокупности IS, объединенных в псевдо-узел, нужен общий, вещательный адрес.

Существует только четыре Ethernet-адреса, которые могут быть использованы в качестве вещательных. Эти адреса различаются своим назначением. Другими словами, каждый из них используется для адресации к определенной группе устройств. В табл. 6.1 приведены вещательные адреса IS-IS и их назначение.

Таблица 6.1. Вещательные адреса Ethernet для протокола IS-IS

Адрес	Описание
0180C2000014	Используется только для связи с маршрутизаторами 1-го уровня
0180C2000015	Используется только для связи с маршрутизаторами 2-го уровня
09002B000005	Применяется для связи со всеми промежуточными системами. Это и есть адрес псевдо-узла
09002B000004	Применяется исключительно для связи со всеми конечными системами

Пакет с любым вещательным адресом принимают и обрабатывают все IS той группы, которой этот адрес присвоен. Поэтому назначенная промежуточная система избавлена от необходимости многократной передачи одного и того же пакета — он передается только один раз на адрес псевдо-узла, и его получают все устройства.

В сетевом окружении IS-IS используются два разных способа маршрутизации и, соответственно, два метода обработки дейтаграмм маршрутизаторами. Сфера действия маршрутизаторов 1-го уровня ограничена своей областью, а маршрутизаторы 2-го уровня отвечают за маршрутизацию как внутри, так и вне области, домена, локальной сети.

Рассмотрим процесс маршрутизации с точки зрения маршрутизатора 1-го уровня. Поскольку конечная система не способна самостоятельно выбирать маршрут доставки пакета, она передает его ближайшему маршрутизатору для дальнейшей пересылки к месту назначения. Далее, маршрутизатор 1-го уровня анализирует указанный в заголовке пакета адрес места его назначения, (который содержит две важнейших части — адрес области и адрес узла). В первую очередь, маршрутизатор интересуется, принадлежит ли искомая система данной локальной области, т.е. сравнивает адрес области назначения с адресом своей области. Если адрес области назначения совпал с адресом области маршрутизатора 1-го уровня, то этот маршрутизатор анализирует вторую часть адреса назначения, т.е. адрес конечной системы-получателя, однозначно

указывающий искомую систему. В случае, если маршрутизатор не знает системы с указанным адресом, он просто отбрасывает пакет. Если же адреса областей искомой системы и маршрутизатора 1-го уровня не совпали, маршрутизатор сканирует свою таблицу маршрутов в поисках соседнего маршрутизатора, способного доставить такой пакет. Далее пакет пересылается подходящему маршрутизатору, и, опять таки, если маршрут найти не удалось, то пакет отбрасывается.

Процесс маршрутизации с точки зрения маршрутизатора 2-го уровня довольно сильно отличается от вышеописанного. Неизменными остаются лишь два первых шага, выполняемых маршрутизатором после получения пакета. Так, если область, которой принадлежит искомая система, совпадает с областью маршрутизатора 2-го уровня, он отсылает пакет к месту назначения, если же искомая в этой области система ему неизвестна, пакет отбрасывается как ошибочный. Так как маршрутизатор 2-го уровня соединяет несколько областей, он определяет область назначения и сравнивает ее адрес с адресами областей соединенных с ним маршрутизаторов 1-го уровня. Если такой маршрутизатор 1-го уровня найден, пакет пересылается к нему для окончательной обработки. Однако если маршрутизатор 2-го уровня не связан с искомой областью непосредственно, он анализирует префикс. А если для доставки пакета требуется внутренний маршрут, то этот пакет пересылается соответствующему маршрутизатору 1-го уровня.

6.6. Пакеты IS-IS

Протокол IS-IS использует пакеты трех базовых форматов:

- *Пакеты-приветствия IS-IS (IS-IS Hello packets),*
- *Пакеты состояния каналов LSP (Link state packets),*
- *Пакеты порядковых номеров SNP (Sequence numbers packets).*

В эти пакеты инкапсулируются данные маршрутизации, передаваемые по сети. Однако далеко не каждый пакет содержит инкапсулированную информацию, адресованную одной системой другой системе. Некоторые пакеты выполняют исключительно служебные функции и используются для внутренних нужд сети, помогая протоколу IS-IS определить конфигурацию и топологию устройств, включенных в сетевое IS-IS-окружение.

Каждый из трех форматов пакетов IS-IS содержит три логические части. Первой частью является 8-байтовый заголовок, общий для пакетов всех трех типов. Заголовок пакета содержит информацию, анализируя которую, принимающий узел определяет тип принятого пакета, включая использованный для его передачи протокол. Но, прежде всего, заголовок пакета содержит адрес назначения,

по которому должен быть доставлен пакет. В частности, это может быть Ethernet-адрес. Второй частью является специфическая для пакетов каждого типа часть с фиксированным форматом. Третья логическая часть также является специфической для типа пакета, но имеет переменную длину.

Формат пакетов протокола IS-IS определен в документе RFC 1142, посвященном описанию самого протокола, но для использования протокола в сетях IP и MPLS/IP он был расширен, что нашло свое отражение в документе RFC 1195. Был изменен формат основного заголовка и добавлены некоторые специфические поля переменной длины для каждого из трех типов служебных пакетов.

6.6.1. Пакеты-приветствия Hello

Пакеты-приветствия Hello (рис. 6.2.) используются маршрутизаторами IS-IS для извещения соседних маршрутизаторов о своем присутствии. С их помощью маршрутизатор устанавливает с этими маршрутизаторами соседские отношения. Точно так же это происходит и в протоколе OSPF. Содержимое приветствия следует за стандартным заголовком, но отделено от него шестью резервными полями, содержащими нули.

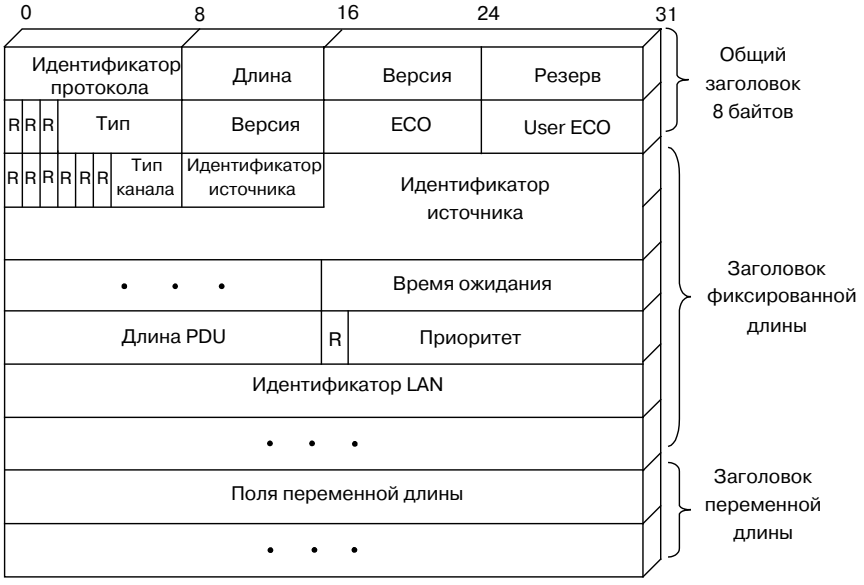


Рис. 6.2. Пакет-приветствие IS-IS

Первым полем в общем заголовке является *идентификатор протокола (Protocol Identifier*), который идентифицирует протокол IS-IS. Это поле содержит константу 131 (в шестнадцатеричной системе 83), которую выделила в качестве стандартного значения для протокола IS-IS международная организация ISO.

Следующим в общем заголовке является поле *длины (Length)*. Оно содержит значение длины заголовка и, если это значение лежит в пределах от 1 до 8 (включительно), определяет размеры части ID идентификатора точки доступа к сетевой услуге NSAP (Network Service Access Point). Если поле содержит нуль, то размер части ID равен 6 байтам. Если поле содержит одни единицы, то часть ID равна 0.

За полем длины следует поле *версии (Version)*, которое содержит номер текущей версии протокола (например, в текущей спецификации IS-IS этот номер равняется единице). В заголовке имеется также еще одно поле *версия*.

Резервное поле *(Reserved)* резервировано для будущего использования, заполняется нулями и игнорируется. Имеются еще три поля, резервированных для будущего использования *(Three Reserved fields)*, пока что все они содержат нули. Поле *тип пакета (Packet Type)* определяет тип пакета IS-IS (Hello, LSP или SNP). Поля *ECO* и *User ECO* заполняются нулями и игнорируются при приеме.

Формат общего заголовка пакета одинаков у всех IS-IS-пакетов. Но поля, следующие за заголовком, разнятся в зависимости от типа пакета и содержат передаваемые от системы к системе данные.

Специфическая информация, составляющая основное содержание сообщения-приветствия, начинается с поля *тип канала (Circuit type),* указывающего тип маршрутизатора, ставшего источником обмена. Для маршрутизатора 1-го уровня это значение равно 01, 2-го уровня — 10, 1-2 уровня — 11.

Далее следует поле *идентификатор источника (Source ID)*, передавшего сообщение.

Поле *время ожидания (Holding Time)* определяет время, в течение которого ожидается приветствие от соседнего маршрутизатора. Если это время истекло, а приветствие так и не получено, маршрутизатор, от которого оно ожидалось, считается нерабочим.

Длина PDU указывает размер PDU-блока в байтах.

Поле *приоритет (Priority)* определяет приоритет маршрутизатора. Так же, как и в OSPF, приоритет используется для выбора назначенного маршрутизатора (Designated Router). При этом маршрутизаторы 1-го и 2-го уровней выбирают каждый своего представителя. Выбор основан на поиске активного маршрутизатора с высшим приоритетом. Но, в отличие от OSPF, в данном случае отсутствует понятие резервного назначенного маршрутизатора (Backup Designated Router). В случае выхода из строя назначенного маршрутизатора, на его место выбирается новый. Очевидно, что для соединений типа точка-точка такие назначенные маршрутизаторы не требуются вовсе, так как в подобных соединениях есть только два устройства — источник и приемник.

Идентификатор локальной сети или канала (LAN ID или Circuit ID) определяет идентификатор локальной сети для вещательных сетей с множественным доступом; такой идентификатор является производным от адреса назначенного маршрутизатора либо 1-го, либо 2-го уровня.

Поля *переменной длины (Variable length fields)* могут включать в себя информацию об аутентификации, адресах соседних устройств и т.д.

Для использования IS-IS в IP и MPLS/IP были введены, как уже было отмечено выше, следующие дополнительные поля:

Поле *Protocols Supported* содержит набор NLPID идентификаторов протоколов сетевого уровня (Network Layer Protocol Identifiers), сообщения которых способна обрабатывать данная IS. Набор NLPID описан в ISO/TR 9577.

Поле *IP Interface Address* содержит IP-адрес интерфейса, взаимодействующего с SNPA (Subnetwork Point of Attachment), через который должен быть передан PDU.

Поле *Autentification Information* содержит информацию аутентификации PDU.

Итак, промежуточная система IS периодически передает своим соседям пакеты-приветствия с целью уведомить их о своей готовности принимать данные. Помимо этого, пакеты-приветствия содержат сведения о назначенном администратором приоритете, типе и адресе промежуточной системы, передавшей это приветствие. По сути, для промежуточных систем приветствия играют ту же роль, что и аналогичные им приветствия в протоколе OSPF. Обмен приветствиями позволяет наладить связь с соседними системами в том же сегменте сети.

Когда промежуточная система IS принимает пакет-приветствие, она обязательно передает в ответ свой пакет-приветствие. Это является подтверждением функционирования обеих систем и доказательством того, что за время, прошедшее с прошлого обмена приветствиями, сетевое окружение не подверглось изменениям. После успешного обмена приветствиями обе промежуточные системы вносят информацию друг о друге в список своих соседей, и тем самым формируют между собой отношения смежности в соответствующей базе данных. Отношения смежности в протоколе IS-IS полностью аналогичны подобным отношениям в протоколе OSPF. Маршрутизатор обязан определить смежность со своими соседями для обмена информацией о маршрутизации и для успешной пересылки данных. При этом строятся две независимые базы данных, одна — для маршрутизаторов 1-го уровня, другая — для маршрутизаторов 2-го уровня.

6.6.2. Пакеты состояния каналов LSP

Пакеты состояния каналов являются основными во всех протоколах, использующих данные об этих состояниях. С их помощью маршрутизаторы обмениваются всей информацией о произошедших в сети изменениях. Промежуточные системы могут обмениваться информацией о маршрутизации сразу же, как только они установили свою смежность.

Информация о маршрутизации распространяется именно в этих LSP-пакетах, и во время лавинной рассылки все IS используют их для обновления своих таблиц маршрутизации. Результатом обмена LSP-пакетами является полное обновление базы данных о состоянии каналов и топологической карты сети. Создаются две раздельные базы данных о маршрутизаторах 1-го и 2-го уровня. А так как LSP-пакет содержит в себе всю информацию, имеющую отношение к маршрутизации, маршрутизатор обязан корректно передать ее от одной IS к другой.

Поля, идущие следом за 8-байтовым заголовком, полностью совпадающим с заголовком пакета-приветствия, имеют следующее назначение.

Поле *длина PDU (PDU Length)* определяет размер PDU-блока в байтах.

Поле *оставшееся время жизни (Remaining lifetime)* определяет время в секундах, оставшееся до момента, когда истечет срок действия PDU-блока.

Поле *идентификатор пакета состояния канала (LSP ID)* определяет системный идентификатор маршрутизатора-отправителя и LSP-пакета.

Значение поля *порядковый номер (Sequence Number)* устанавливается отправителем и используется получателем для контроля очередности следования пакетов с данными об обновлении.

Поле *контрольная сумма (Checksum)* используется для контроля корректной передачи заголовка.

Бит *P (Partition — P)* присутствует в пакетах, сформированных маршрутизаторами как 1-го, так и 2-го уровней, но используется только маршрутизаторами 2-го уровня. Если бит имеет значение 1, это свидетельствует о том, что передавший пакет маршрутизатор поддерживает восстановление раздела. Так как большинство маршрутизаторов лишены такой аппаратной функции, то зачастую значение этого бита равно нулю.

Поле *вложения (ATT — Attachments)* определяет количество IS, подключенных к маршрутизатору-отправителю пакета, а также поддерживаемые им метрики. Выше уже упоминались все четыре

вида метрик: ошибки, стоимость, задержка и метрика по умолчанию. Метрика *ошибки* отображает надежность канала. Метрика *стоимость* показывает денежную стоимость использования канала. Метрика *задержка* измеряет временную задержку передачи пакетов по каналу. И, наконец, *метрика по умолчанию* — это произвольное значение от 1 до 64, присвоенное каналу администратором. Каждая из этих метрик может быть использована маршрутизатором для вычисления кратчайшего пути между отправителем и получателем.

Значение бита *переполнение (OL — Overload)* обычно равно нулю. Значение 1 присваивается этому биту маршрутизатором-отправителем в качестве признака нехватки памяти для своевременной обработки всей поступающей информации. Маршрутизатор-получатель, обнаруживший бит OL со значением 1, старается разгрузить передавшее пакет устройство до получения от него нового пакета с нулевым значением этого бита.

Бит *тип промежуточной системы (IS Type)* определяет тип маршрутизатора отправителя — 1-й или 2-й уровень.

Поля *переменной длины (Variable length fields)* могут содержать адреса соседних IS, сведения о типах поддерживаемых протоколов (CLNP или IP, или оба одновременно), информацию об аутентификации, список достижимых систем во внутренней и внешней сети и т.д. Сведения о достижимости внутренних и внешних систем, содержащаяся в LSP-пакете, информируют все маршрутизаторы о системах, известных в качестве действующих получателей. Эти сведения включают в себя IP-адреса таких систем, наравне с используемой метрикой и привязанной к ним маской. LSP-пакеты содержат данные для обновления таблиц маршрутизации. И когда IS получает такой пакет, она обязательно сравнивает свежеполученные данные с информацией, хранящейся в базе данных о маршрутах для маршрутизаторов либо только 1-го, либо также и 2-го уровня, в зависимости от типа обновления.

Полями, которые дополнительно включаются в формат пакета для работы в IP и MPLS/IP сетях, являются все поля, добавленные в формат пакетов-приветствий, а также поле *IP Internal Reachability Information,* содержащее IP-адреса внутри домена маршрутизации, доступные через один или более интерфейсов данной IS.

Для пакетов 2 уровня добавляются еще поле *IP external Reachablity Information*, содержащее IP-адреса вне домена маршрутизации, которые доступны через интерфейсы данной IS, и поле *Inter-Domain Routing Protocol Information* с информацией, которая прозрачно переносится на втором уровне для нужд любого междоменового протокола, работающего в границах промежуточных систем.

После получения и обработки LSP-пакета, IS вычисляет, используя описанный в предыдущей главе алгоритм Дийкстры и с учетом обновленной базы данных, новые оптимальные маршруты, которые заносятся в таблицу маршрутизации.

6.6.3. Пакеты порядкового номера SNP

Пакеты порядкового номера (рис. 6.3.) обеспечивают гарантию обновления в каждой IS базы данных о состоянии каналов при получении очередных LSP-пакетов. SNP-пакеты делятся на полные и частичные. Полный пакет *CSNP (Complete Sequence Number Packet)* включает в себя список всей LSP-информации, содержащейся в базе данных о состоянии каналов. Частичный пакет *PSNP (Partial SNP)* содержит лишь некое подмножество LSP-информации. В сетях с множественным доступом CSNP-пакеты передаются назначенным маршрутизатором для описания всего содержимого базы данных о состоянии каналов. О назначенных маршрутизаторах упоминалось в предыдущей главе, посвященной OSPF, но еще раз отметим, что в IS-IS резервные назначенные маршрутизаторы отсутствуют. В то же время, PSNP-пакеты используются на регулярной основе, как в вещательном режиме, так и в режиме точка-точка (point-to-point) для запросов повторной передачи утерянных (т.е. не принятых) LSP-пакетов, или для запросов дополнительной информации об обновлении в случае обнаружения неточностей в базе данных. А так как для маршрутизаторов 1-го и 2-го уровня ведутся раздельные базы данных, то и CSNP- и PSNP-пакеты тоже бывают двух разных уровней.

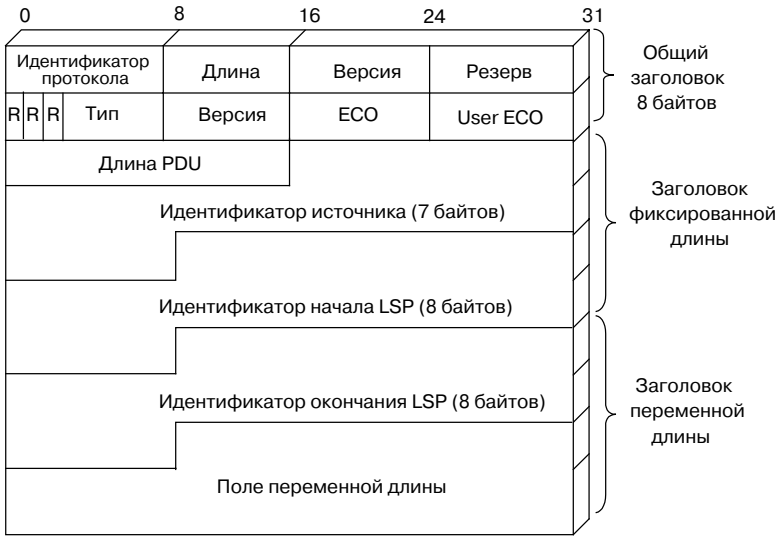


Рис. 6.3. IS-IS пакет порядкового номера SNP

Представленный на рис. 6.3 пакет SNP содержит после стандартного 8-байтового общего заголовка следующие поля.

Двухбайтовое поле *длина PDU (PDU Length)*, которое определяет размер PDU-блока в байтах.

Поле *идентификатор источника (Source ID)*, которое содержит системный *идентификатор отправителя* в случае вещательного режима или идентификатор канала в случае режима точка-точка и состоит из 7 байтов.

Поле *идентификатор начала LSP (Start LSP ID)*, которое содержит 8 байтов и присутствует только в CSNP-пакетах, где оно используется для определения начального диапазона LSP-информации, заключенной в пакете.

Поле *идентификатор окончания LSP (End LSP ID)*, которое точно так же содержит 8 байтов и присутствует только в CSNP-пакетах, где используется для определения конечного диапазона LSP-информации, заключенной в пакете.

Поля *переменной длины (Variable length fields)*, куда включаются LSP-записи базы данных, которые повторяются в зависимости от номера LSP, содержащегося в пакете согласно табл. 6.2. Эти поля доставляют принимающей IS информацию о корректности полученных данных об обновлении и о сроке действия этого обновления.

Таблица 6.2. Поля переменной длины LSP пакетов.

LSP-записъ	Описание
Оставшееся время жизни	Показывает время, в течение которого IS не имеет права получать данные об обновлении, пришедшие после уже полученных
Идентификатор SP	Системный идентификатор
Порядковый номер SP	Порядковый номер пакета с данными об обновлении
Контрольная сумма	Проверка целости полученной LSP-информации

Единственным полем, добавленным в формат LSP-пакета документом RFC 1195, является поле параметра *Authentication Information.*

Глава 7
Протокол маршрутизации BGP

7.1. Использование протокола BGP в MPLS

Третий из используемых технологией MPLS протоколов маршрутизации (OSPF, IS-IS, BGP-4) называется *Border Gateway Protocol (BGP).* Его первая версия BGP-1 появилась в 1989 году, а повсеместное внедрение BGP-4 началось с 1993 года. Подробно четвертая версия протокола BGP-4 (а именно ее мы и будем здесь рассматривать) представлена в написанном И. Ректером и Т. Ли документе RFC1771, а также в его многочисленных дополнениях и расширениях. Рассмотрим только основные функции этой последней версии протокола, и те, которые непосредственно используются в технологии MPLS.

В частности, это относится к многопротокольному расширению протокола BGP-4, мало освещенному в существующей литературе, но нашедшему применение при построении MPLS-VPN.

Напомним описанное в главе 1 разделение функций технологии MPLS на два компонента — управление и пересылка пакетов,— изображенное на рис. 1.1 в виде двух плоскостей. Управляющий компонент использует протоколы маршрутизации OSPF, IS-IS, BGP-4 для обмена маршрутной информацией между маршрутизаторами. На основе этой информации в каждом маршрутизаторе формируется и модифицируется сначала таблица маршрутизации, а затем, с учетом информации о смежных системах в каждом интерфейсе,— таблица пересылки пакетов. Когда система получает пакет, пересылающий компонент анализирует информацию, содержащуюся в его заголовке, ищет соответствующую запись в таблице пересылки и направляет этот пакет в выходной интерфейс.

Но если рассмотренные в двух предыдущих главах протоколы OSPF и IS-IS выполняют эту задачу в пределах одной автономной системы AS, которая представляет собой, по сути, самодоста-

точную независимую сеть, не имеющую полученных каким-либо логическим путем сведений о других сетях в составе всей сети MPLS, то роль протокола BGP-4 гораздо шире. Его основное назначение как раз и состоит в том, чтобы передавать от одного BGP-маршрутизатора другим BGP-маршрутизаторам информацию о наличии других автономных сетей и об их структуре, формируя тем самым иерархическую схему маршрутизации, связывающую разные узлы и автономные сети в единую MPLS/IP-сеть и позволяющую свободно устанавливать связь между собой системам, неизвестным друг другу.

Необходимость декомпозиции глобальной MPLS/IP-сети на автономные системы обусловлена очевидными мощностными соображениями: если большое количество маршрутизаторов попытается вступить во взаимодействие, трафик превысит все мыслимые границы. Поэтому администраторы сетей обычно пользуются проверенными эмпирическими правилами, согласно которым количество маршрутизаторов в локальной сети не должно превышать 50–60, а в транзитной глобальной сети это число еще меньше — до 10–15 маршрутизаторов. Эти значения зависят, разумеется, от производительности сетевых маршрутизаторов, интенсивности трафика, допустимых задержек передачи пакетов, стабильности характеристик трафика в сети, периодичности обновления маршрутной информации и т.п. Очевидно, что при слабом сетевом оборудовании и большом трафике количество используемых маршрутизаторов должно быть еще меньше. Но даже в условиях высокой пропускной способности сетевого оборудования, нетребовательности к задержкам пакетов и ограниченном трафике, даже в этих условиях для сети MPLS/IP сколько-нибудь значительного размера междоменовая маршрутизация с помощью протокола BGP является необходимой.

BGP специфицируется как сеанс связи между двумя узлами, а так как в сети будет параллельно выполняться много BGP-сеансов, один маршрутизатор может быть вовлечен в несколько таких сеансов. В ходе BGP-сеанса между одноранговыми узлами протокола BGP происходит обмен сообщениями по TCP-соединению. Среди них — сообщения открытия BGP-сеанса, сообщения извещения однорангового узла о том, что активны новые маршруты, а старые — нет, напоминания одноранговому узлу о том, что соединение активно, а также сообщения, информирующие о любых необычных ситуациях или о сбоях. Все эти сообщения рассматриваются далее.

Версия 4 протокола BGP значительно отличается от предыдущих реализаций BGP и фактически включает в себя два отдельных протокола: протокол *EBGP — External Border Gateway Protocol,* — используемый для маршрутизации между автономными системами, и протокол *IBGP — Internal Border Gateway Protocol,* — используемый для маршрутизации внутри автономной системы.

Второе принципиальное отличие протокола BGP от OSPF и IS-IS заключается в том, что он относится не к категории *протоколов маршрутизации по состоянию каналов*, а к категории *дистанционно-векторных протоколов* (как и протокол RIP, упоминавшийся в главе 5). В протоколах каждой из этих двух категорий применяются свои алгоритмы определения маршрута и свои методы вычисления этого маршрута на основании значений метрик, которые указываются для каждого маршрута в таблице маршрутизации. Принцип вектора расстояния подразумевает выбор маршрута, исходя из кратчайшего расстояния между системами, определяемого числом пересылок. Протоколы на основе вектора расстояния обычно учитывают только число пересылок (hops) в маршруте, о чем уже говорилось выше. Для вычисления маршрута в таких протоколах обычно применяется *алгоритм Беллмана-Форда*.

Прежде чем перейти к рассмотрению этого алгоритма, который, кстати, полезно сравнить с описанным в главе 5 алгоритмом Дийкстры, отметим, что кроме обычных параметров дистанционно-векторных протоколов в BGP используется дополнительный механизм, именуемый *маршрутно-векторной маршрутизацией (path-vector routing).* Это обусловлено тем, что ни маршрутизация с учетом состояния каналов, ни дистанционно-векторная маршрутизация в чистом виде для внешней маршрутизации не эффективны.

7.2. Алгоритм Беллмана-Форда

Метод *дистанционно-векторной маршрутизации* иногда называют также методом Беллмана или методом Форда-Фалкерсона по имени исследователей, которые первыми опубликовали идею алгоритма. Сама эта идея довольно проста. Маршрутизатор хранит в таблице список всех известных маршрутов с указанием в каждом элементе таблицы сети получателя и целого числа — количества пересылок до этой сети. Время от времени каждый маршрутизатор отсылает копию своей таблицы другим маршрутизаторам, к которым он имеет прямой доступ. Получив такую копию от LSR-B, маршрутизатор LSR-A анализирует полученный набор адресатов и расстояний до них. Маршрутизатор LSR-A заменяет данные в своей таблице, если маршрутизатору LSR-B известен более короткий, чем имеющийся в ней, маршрут к получателю, или если в его списке есть неизвестный ему до сих пор получатель, а также если LSR-A выполняет маршрутизацию через LSR-B, но расстояние от LSR-B до получателя изменилось.

На основе этой таблицы, согласно алгоритму Беллмана-Форда, и рассчитывается значение метрики (например, стоимости маршрута, задержки и т.п.) для каждой пересылки в сети и поиск

минимального суммарного числа пересылок. Обратим внимание на то, что понятие *дистанционный вектор* связано с характером периодически передаваемой протоколом информации. В сообщениях содержится пара чисел {R, D}, где R — вектор, определяющий получателя, а D — расстояние до этого получателя, т.е. один маршрутизатор сообщает другому о своей возможности достичь получателя R за D пересылок.

При маршрутизации по протоколу BGP пересылка возможна как между внутренними маршрутизаторами (расположенными внутри одной AS), которые работают под управлением протокола IBGP, так и между внешними маршрутизаторами, соединяющими разные автономные системы AS, когда маршрутизация выполняется под управлением протокола EBGP.

Применяемый в BGP маршрутно-векторный механизм помогает решить проблему традиционной дистанционно-векторной маршрутизации, возникающую в условиях функционирования между автономными системами. Дело в том, что в разных AS могут применяться разные метрики измерения длины маршрутов, что может привести к проблемам интерпретации одних и тех же числовых значений во внешних BGP-маршрутизаторах. Поэтому механизм маршрутно-векторной маршрутизации предусматривает отказ от метрики маршрута. Тогда маршрутизаторы просто обмениваются информацией об автономных системах, к которым и через которые у них есть доступ.

7.3. Нумерация автономных систем в BGP

Как и для рассмотренных ранее протоколов маршрутизации, автономные системы являются основной структурной единицей для протокола BGP. Да и не только для этих протоколов. Автономные комбинации узлов, с которыми работают протоколы, называются по-разному — *доменами* в протоколе PNNI, *равноправными группами (Peer-Groups)* в протоколе IS-IS и т.п.,— однако природа автономных систем сетевых объектов одна и та же. *Автономная система AS* — это определенная группа маршрутизаторов, каким-либо образом связанных между собой в локальной или в глобальной сети, как это понималось и в предыдущих главах. В контексте BGP к этому уместно добавить, что определение автономной системы тесно связано с понятием *точек доступа в IP-сеть PoP (Point of Presence)*, которых автономная система имеет, как правило, две или более для распределения нагрузки и обеспечения надежности.

Номер автономной системы *ASN (Autonomous System Number)* является идентификатором определенной AS в объединенной интрасети и отличает ее от других автономных систем, также под-

держивающих протокол BGP. Обычно номера ASN прямо связаны с IP-адресами сегментов сети, которой эти AS принадлежат, что дает гарантию доставки данных от внешних автономных систем через граничные шлюзы данной системы.

Диапазоны номеров AS описаны в документе-инструкции RFC 1930. В большинстве случаев ASN назначается провайдером Интернет из подмножества принадлежащих ему номеров в пределах от 1 до 65535. В этом подмножестве выделены две основные категории ASN: *общедоступные номера* в диапазоне от 1 до 64511, отводимые Интернет-провайдерам и крупным корпоративным сетям, и *частные номера AS*, которые не используются в Интернет, а служат для IBGP-маршрутизации в больших сетях, поддерживающих протокол BGP. Номер из диапазона частных ASN 64512—65535 выбирается самим пользователем.

Существует три класса автономных систем AS:

- системы с множественной адресацией (multihomed),
- тупиковые, имеющие только один выход в Интернет (single-homed),
- многоканальные транзитные сети (multihomed transit).

Системы с множественной адресацией (multihomed) — это системы, имеющие несколько соединений с внешними автономными системами. Эти системы еще называют *многоканальными нетранзитными (multihomed nontransit).* Автономная система с множественной адресацией может принимать маршрутную информацию от всех соединенных с ней систем, но сама она выполняет только внутреннюю маршрутизацию.

Многие автономные системы на самом деле являются *тупиковыми (stub)* или *одноканальными (single-homed)* и имеют лишь один выход во внешнюю сеть. Соответственно, они не требуют никаких дополнительных правил для управления ими и обслуживания большого списка BGP-маршрутов в шлюзе, у которого имеется только один выход в Интрасеть.

Третий тип автономной системы — *транзитная сеть*. Транзитные AS — это системы с множественной адресацией, которые принимают информацию от внешних автономных систем и выполняют маршрутизацию этой информации к другим внешним AS.

7.4. Маршрутизаторы BGP

Имеется три типа маршрутизаторов BGP: спикеры, пограничные шлюзы и равноправные маршрутизаторы BGP.

Спикерами BGP (BGP speakers) являются все маршрутизаторы автономной системы BGP. При настройке администратор дол-

жен присвоить им номер той AS, которой они принадлежат. Если IP-адрес спикера BGP не соответствует ASN, то такой спикер не может работать в этой автономной системе. Более того, те спикеры BGP, IP-адреса подсетей которых относятся к данной автономной системе, но путь к ним лежит через спикер, ей не принадлежащий, не смогут получать данные от этой автономной системы.

Спикеры BGP, соединяющие две или несколько автономных систем, называются *пограничными шлюзами (Border Gateways)*. Они нужны автономным системам только в том случае, если AS использует для связи с другими автономными системами MPLS/IP сети протокол EBGP. Справедливо и обратное утверждение: каждая автономная система может иметь несколько пограничных шлюзов. Так бывает, если автономная система связана с одной или с несколькими внешними AS через несколько спикеров BGP. Задача пограничного шлюза — извещать о внутренних маршрутах автономной системы (и о других известных ему маршрутах) любой внешний спикер BGP, с которым этот шлюз связан.

Согласно принципам сетевой маршрутизации, во время сеансов связи спикеры BGP обмениваются маршрутной информацией о топологии и метрических характеристиках соответствующих участков сети. Такой обмен происходит между *равноправными маршрутизаторами (BGP peer)* автономной системы.

Равноправные маршрутизаторы BGP не обязательно должны иметь прямые связи друг с другом; однако между двумя спикерами BGP всегда должен существовать стандартный способ коммуникации для того, чтобы они могли инициировать сеанс связи. Когда BGP устанавливает сеанс связи двух равноправных маршрутизаторов, между которыми нет прямого соединения, такая связь называется *одноранговой связью с пересылкой по протоколу EBGP (EBGP multihop peering).* Используя внешние соединения по протоколу EBGP, спикер BGP может инициировать сеанс связи с другими спикерами, находящимися на расстоянии нескольких пересылок. Во всех таких случаях для организации сеансов BGP использует протокол TCP, а все сеансы равноправной связи создаются и активизируются через TCP-порт 179.

При одноранговой связи спикеры BGP обмениваются полными копиями таблиц маршрутизации во время первоначального двустороннего сеанса, включая повторные запуски. При перезапусках маршрутизатор каждый раз обменивается полными копиями таблиц маршрутизации со всеми одноранговыми маршрутизаторами, так что этот процесс может быть довольно продолжительным. Маршрутизаторы BGP могут хранить данные о многих (от сотен до тысяч) объявляемых маршрутах. Во время первого сеанса обмена данными о корректировке для нового спикера BGP ему должны

быть переданы все сведения об этих маршрутах, в связи с чем, в зависимости от количества объявляемых маршрутов в том или ином маршрутизаторе, начальный обмен данными о корректировке может быть довольно интенсивным. Сведение к минимуму числа перебоев в работе сети/маршрутизатора позволяет уменьшить нагрузку сети, обусловленную трафиком обмена полными копиями таблиц маршрутизации.

В остальных случаях после установления соединения между равноправными спикерами BGP они обмениваются лишь данными о вносимых в таблицы изменениях. Поэтому чем продолжительнее функционирует маршрутизатор в автономной системе, тем менее интенсивным становится его обмен с другими маршрутизаторами данными о корректировке.

BGP является протоколом с устойчивым состоянием, и поэтому маршрутная информация, обмен которой между одноранговыми узлами проведен успешно, не нуждается в обновлении. Эта информация считается действительной до тех пор, пока один из одноранговых узлов не уведомит другой о том, что это не так, или пока не закончится BGP-сеанс между ними.

Ключевым принципом, лежащим в основе протокола BGP, является то, что когда один одноранговый узел информирует своего партнера о том, что IP-адрес доступен по сообщенному маршруту, партнер может быть уверен, что равноправный узел уже успешно использует этот маршрут для передачи собственного трафика. При этом подразумевается, что маршруты, о которых узел извещен, могут использоваться всегда, когда о них объявляется. Наряду с возможностью использовать маршрут, предоставляется набор атрибутов, связанных с IP-префиксом.

7.5. Протокол EBGP

Протокол EBGP (Exterior Border Gateway Protocol) используется для установления соединения между спикерами BGP разных автономных систем, включая коммуникации между Интернет-провайдерами и точками доступа PoP, а также между большими корпоративными сетями и провайдерами услуг.

Во время одноранговых сеансов, рассмотренных в предыдущем параграфе, маршрутизаторы BGP получают информацию о своем окружении от других маршрутизаторов — спикеров разных автономных систем. Для того чтобы любой спикер, находящийся вне некоторой AS, мог успешно обмениваться с этой AS информацией, он должен быть осведомлен о ее местоположении и знать ее адрес. Для этого спикеры BGP объявляют об известных им маршрутах к другим автономным системам, т.е. о маршрутах, не входящих в их

собственные AS. Запись маршрута BGP содержит сведения о том, как доставить от одного пункта к другому информацию, предназначенную для адреса xxx.xxx.xxx.xxx. Маршруты указывают сетевой IP-адрес (если сеть не находится в автономной системе, которой принадлежит устройство-отправитель) и адрес маршрутизатора, который будет использован для следующей пересылки. Адрес маршрутизатора для следующей пересылки — это адрес пограничного шлюза, который будет использован для доставки информации к пункту назначения. Такая информация позволяет спикерам BGP сообщать о маршрутах, с которыми они не соединены непосредственно, и обеспечивать доставку данных от разных AS практически к любой другой автономной системе.

Когда спикер имеет информацию о маршруте, которую необходимо передать другим спикерам BGP (внутренним или внешним), он извещает их об этом маршруте. Таким образом, *уведомление или извещение о маршруте* — это способ, пользуясь которым, один спикер представляет другим спикерам сведения о маршруте или о корректировке маршрута, что позволяет спикерам BGP знать друг о друге и о сетях, которые они представляют. Маршруты, о которых извещают спикеры BGP, делятся на два типа — динамические и статические.

Динамический маршрут — это маршрут, о котором спикер узнает в результате обмена с другими маршрутизаторами сведениями о корректировке. Такой процесс обмена обеспечивает динамическое перераспределение маршрутов BGP (в противоположность статическому перераспределению маршрутов BGP).

К *статическим маршрутам* относятся маршруты, которые программируются в спикере BGP администратором.

7.6. Протокол IBGP

Очевидно, что BGP-маршрутизаторы, находящиеся в одной AS, тоже должны обмениваться между собой маршрутной информацией. Это необходимо для согласованного отбора внешних маршрутов в соответствии с политикой данной AS и для организации транзитных маршрутов через автономную систему. Такой обмен производится по протоколу BGP, который в этом случае называется IBGP (Internal BGP).

Отличие IBGP от EBGP состоит в том, что при извещении о маршруте соседнего маршрутизатора, находящегося в той же AS, в *список номеров автономных систем AS_Path* не вводится номер собственной автономной системы (AS_Path — список номеров автономных систем, через которые должна пройти дейтаграмма по пути в указанную сеть). Действительно, если номер AS будет вве-

ден в список, и сосед передаст сведения об этом маршруте далее (опять с добавлением номера той же AS), то одна и та же AS будет присутствовать в AS_Path дважды, что и расценивается как цикл. Чтобы не возникло циклов, маршрутизатор не должен сообщать по протоколу IBGP о маршруте, сведения о котором получены также по этому протоколу, поскольку нет способов определить зацикливание при извещении о BGP-маршрутах внутри одной AS.

Следствием этого является необходимость создавать полный графа IBGP-соединений между пограничными маршрутизаторами одной автономной системы, т.е. каждая пара маршрутизаторов должна устанавливать между собой соединение по протоколу IBGP. При этом число соединений может оказаться очень большим, и для его уменьшения применяются разные решения: разбиение AS на области (конфедерации), применение серверов маршрутной информации и др.

Сервер маршрутной информации (аналог назначенного маршрутизатора в OSPF), обслуживающий группу BGP-маршрутизаторов, принимает данные о маршруте от одного члена группы и рассылает их всем остальным. Членам группы нет необходимости устанавливать BGP-соединения попарно; вместо этого каждый из них устанавливает соединение с сервером. Группой маршрутизаторов могут быть, например, все BGP-маршрутизаторы одной AS, однако маршрутные серверы могут применяться для уменьшения также и числа внешних BGP-соединений в случае, когда в одной физической сети находится много BGP-маршрутизаторов из разных AS. Отметим, что сервер маршрутной информации — не обязательно маршрутизатор, то есть, в общем случае, узел, в котором есть модуль BGP, может и не быть маршрутизатором. В технических документах этот факт подчеркивается тем, что для обозначения BGP-узла используется термин BGP-спикер, а не BGP-маршрутизатор.

7.7. Конфедерации BGP

Как уже говорилось в предыдущем параграфе, все BGP-спикеры в автономной системе связаны по схеме «каждый с каждым», т.е. с увеличением числа спикеров N число физических соединений между ними возрастает как N(N-1)/2. Кроме использования сервера маршрутизации, есть другой метод уменьшения числа физических соединений между спикерами BGP — формирование внутри автономной системы так называемых *конфедераций BGP (BGP Confederations)*. Одна большая AS может быть разделена на несколько подсистем AS, которые сохраняют тот же номер ASN. Всем конфедерациям внутри одной AS, чтобы их можно было различать, присваиваются идентификаторы — номера, выбираемые в соответствии со схемой нумерации автономной системы.

Так как конфедерации представляют собой обычные AS, только меньшего размера, одноранговые члены IBGP внутри каждой из них должны быть связаны каждый с каждым, а чтобы создать из конфедераций полную AS, в каждой из них необходимо назначить одноранговые маршрутизаторы IBGP, которые будут обеспечивать связь между конфедерациями. В каждой конфедерации эти маршрутизаторы должны работать и как спикеры IBGP, и как спикеры EBGP. Но поскольку спикеры EBGP работают в среде IBGP, они следуют всем правилам, установленным для IBGP-спикеров.

7.8. Карты маршрутов

Карты маршрутов BGP дают администратору возможность выбрать из огромного числа маршрутов BGP в Интернет, какие маршруты будут перераспределяться его спикерами. Карты маршрутов работают только на уровне извещений о корректировках и не фильтруют маршруты, поступающие в маршрутизатор. В карте маршрутов содержатся метрики, по которым выполняется алгоритм маршрутизации. Эти метрики представляют собой определяемые администратором значения, которые позволяют выбирать для передачи данных наиболее подходящий из альтернативных маршрутов, ведущих к одному и тому же пункту назначения. Как правило, выбирается маршрут, который имеет самое низкое совокупное значение метрик, или, другими словами, самую низкую стоимость. Однако администратор может присвоить какому-либо маршруту более приоритетное значение метрики. Для технологии MPLS это означает возможность присваивать трафику определенного типа, идущему от определенного источника, нужный приоритет и возможность изменять этот приоритет путем динамического изменения метрик.

Наиболее распространенным применением маршрутной карты является изменение такого атрибута маршрутов BGP, как *локальный приоритет (Local Preference)*.

Когда во время однорангового сеанса нарушается соединение спикера BGP с каким-либо другим маршрутизатором, числящимся в его таблице маршрутизации, то говорят, что происходит *флеппинг (flapping) маршрута*. Причиной флеппинга может быть сбой в работе процессора или любой другой отказ в канале или во всей сети. Явление флеппинга в одном из маршрутов, когда все спикеры, использующие этот маршрут, продолжают перераспределять информацию о нем по сети, может привести к перегрузке маршрутизатора, передающего пакеты по неисправному маршруту. Чтобы избежать этого, разработана система *демпфирования флеппинга (flap dampening)*, принцип работы которой состоит в том, что маршруты, на которых флеппинг происходит в течение определенного

времени, удаляются из таблиц маршрутизации. После восстановления работоспособности маршрута он вводится в рабочий режим по истечении определенной выдержки времени.

7.9. Метрики маршрутов

Рассмотренные выше маршрутные карты помогают администратору изменять значение метрик, которые применительно к протоколу BGP называются *атрибутами* и определяют используемый в нем алгоритм маршрутизации. Ниже приводятся основные атрибуты BGP.

Атрибут *AS-Path* содержит список всех автономных систем, через которые пакет проходит к пункту назначения, и включается в сообщение о корректировке.

Атрибут *следующая пересылка Next-Hop* указывает адрес того пограничного маршрутизатора, который должен быть следующим на пути к получателю.

Атрибут *происхождение Origin* указывает, от какого отправителя поступило данное сообщение о корректировке, и является ли маршрут внутренним (IBGP) или внешним (EBGP). Атрибут Origin может иметь одно из трех значений:

- IGP, когда информация сетевого уровня о достижимости NLRI является внутренней по отношению к автономной области, где был сформирован маршрут. Если атрибут Origin имеет значение IGP, то данные о маршруте получены от внутреннего протокола маршрутизации, такого как RIP, OSPF или IBGP, и спикер BGP может считать, что маршрут находится в его AS.
- EGP, когда информация сетевого уровня о достижимости NLRI получена через протокол внешнего шлюза EGP. Если атрибут Origin имеет значение EGP, то данные о маршруте получены во внешнем сообщении о корректировке, и для доступа к маршруту спикер BGP будет использовать протокол EBGP.
- INCOMPLETE, когда информация сетевого уровня о достижимости NLRI получена другими средствами. В большинстве случаев атрибут Origin со значением INCOMPLETE означает, что данные получены в результате перераспределения маршрутов и определяют, каким образом получена информация о маршруте и как она помещается в таблицу маршрутизации.

К этому нужно сделать одно замечание. Не следует путать смысл аббревиатур IBGP, EBGP и аббревиатур IGP, EGP. Аббревиатуры IBGP и EBGP обозначают специфические протоколы в рамках BGP, а IGP и EGP обозначают Interior Gateway Protocol (протокол внутренних шлюзов) и Exterior Gateway Protocol (протокол внешних шлюзов).

Атрибут *локальное предпочтение Local_Preference* определяет точку выхода из AS с учетом преимущества одного маршрута перед другими (когда имеется несколько равноценных точек выхода к пункту назначения). Маршрутизатор сравнивает значения атрибута Local Preference у разных маршрутов, чтобы выбрать из них наиболее предпочтительный. Чем выше *Local_Pref* маршрута, тем выше вероятность его использования. Если атрибут *Local_Pref* не определен, значение этого атрибута по умолчанию равно 100.

Атрибут *Aggregator* — необязательный атрибут, значение которого содержит в двух первых байтах номер последней AS, сформировавшей объединенный маршрут, а в следующих четырех байтах — адрес BGP-спикера, сформировавшего этот маршрут. При создании объединенного маршрута характеристики нескольких отдельных маршрутов комбинируются таким образом, что для них может быть определен один общий маршрут.

7.10. База данных маршрутизации

База данных маршрутизации RIB (Routing Information Base) BGP-спикера состоит из трех частей: *Adj-RIBs-In,* где хранится маршрутная информация о входящих маршрутах, которая была получена в сообщениях о корректировке, *Loc-RIB,* где содержится локальная маршрутная информация, выбранная из Adj-RIBs-In на основе локальной политики маршрутизации BGP-спикера, и *Adj-RIBs-Out,* где помещается информация, которую BGP-спикер выбрал для передачи другим маршрутизаторам в процессе одноранговой связи. Эта информация будет включена в рассылаемые BGP-спикером сообщения о корректировке.

7.11. Сообщения BGP

7.11.1. Общий заголовок

Все сообщения BGP начинаются общим заголовком, за которым следует заголовок, специфический для сообщений каждого типа. Общий заголовок указывает, что это — сообщение протокола BGP, а также определяет тип информации, содержащейся в сообщении. На рис. 7.1 показаны поля общего заголовка BGP.

Маркер (Marker), имеет длину 16 байтов и указывает на принадлежность сообщения протоколу BGP. Принимающий маршрутизатор BGP сам определяет длину пакета, чтобы прогнозировать содержание поля маркера. Если в полученном пакете содержание поля маркера отличается от прогнозируемого, этот пакет подлежит уничтожению.

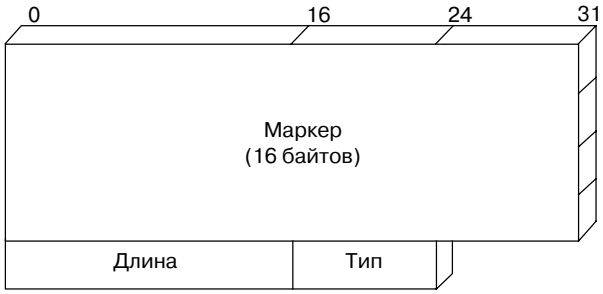


Рис. 7.1. Структура общего заголовка BGP

Общий заголовок содержит четыре поля:

Длина (Length), размером в 2 байта, определяет общую длину пакета BGP в байтах.

Тип (Type) — однобайтовое поле, указывающее тип сообщения.

Далее следует переменное число байтов *пересылаемых данных,* составляющих содержание самого сообщения. Максимальный размер сообщения BGP составляет 4096 байтов. Протоколом BGP предусмотрены сообщения четырех типов, рассматриваемые ниже в этом параграфе.

7.11.2. Запрос соединения OPEN

Сообщение о запросе BGP-соединения OPEN — это сообщение типа 1. Его передает маршрутизатор, которому нужно организовать сеанс связи с другим равноправным маршрутизатором BGP. Принявший это сообщение маршрутизатор подтверждает установление соединения, передавая сообщение *KEEPALIVE*. На рис. 7.2 показана структура заголовка сообщения *OPEN*.

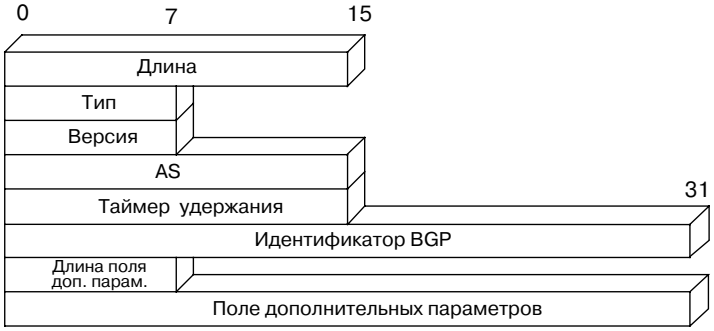


Рис. 7.2. Структура заголовка сообщения OPEN

В заголовке сообщения OPEN предусмотрены следующие поля.

Поле *версия (Version),* длиной 1 байт, указывает применяемую отправителем версию протокола BGP, чтобы встречный марш-

рутизатор смог определить, используют ли оба маршрутизатора совместимые протоколы. Обычно маршрутизаторы начинают с самой последней версии протокола, которую они могут поддержать, а затем, сбрасывая сеансы, постепенно понижают версию, пока не достигнут консенсуса.

Поле *автономная система (AS),* длиной 2 байта, содержит уникальный номер той AS, в которую входит спикер-отправитель.

Поле *таймер удержания (hold-time)*, тоже 2-байтовое, определяет максимальный интервал времени между приемом сообщений KEEPALIVE и UPDATE. При приеме и того, и другого сообщения таймер сбрасывается. Если маршрутизатор, передавший сообщение, не получает ответа в течение времени, указанного в этом поле, считается, что получатель отключен.

Поле *идентификатор BGP,* длиной 4 байта, однозначно определяет отправителя и обычно содержит MAC-адрес маршрутизатора и его ASN.

Необязательное поле, размером 1 байт, содержит *значение длины поля дополнительных параметров*, а если дополнительные параметры не задаются, то в этом поле указывается значение 0.

Необязательное *поле дополнительных параметров (Optional Parameter)* содержит любые параметры, которые маршрутизатор-отправитель может передать получателю. В четвертой версии BGP сообщение OPEN содержит только один дополнительный параметр — параметр аутентификации, используемый при необходимости аутентифицировать передаваемые пакеты.

7.11.3. Сообщение об обновлении UPDATE

Принцип обновления маршрутной информации составляет основу протокола BGP. *Сообщение UPDATE* рассылается маршрутизаторам BGP с целью внесения изменений в таблицы маршрутизации. На рис. 7.3 представлена структура этого сообщения, в которое входят три блока: *информация сетевого уровня о достижимости (NLRI — Network Layer Reachability Information), атрибуты маршрутов* и *недостижимые маршруты.*

Эти три информационных блока разнесены по следующим полям (рис. 7.3). Первые два поля относятся к недостижимым маршрутам. Первое поле, размером 2 байта, содержит значение *длины списка отменных (withdrawn) маршрутов.* Если в сообщении нет таких маршрутов, то это поле имеет значение 0. Далее следует поле *отмененных маршрутов*, которое содержит информацию об отменяемых маршрутах, обозначаемых префиксами IP-адресов, и имеет переменную длину.

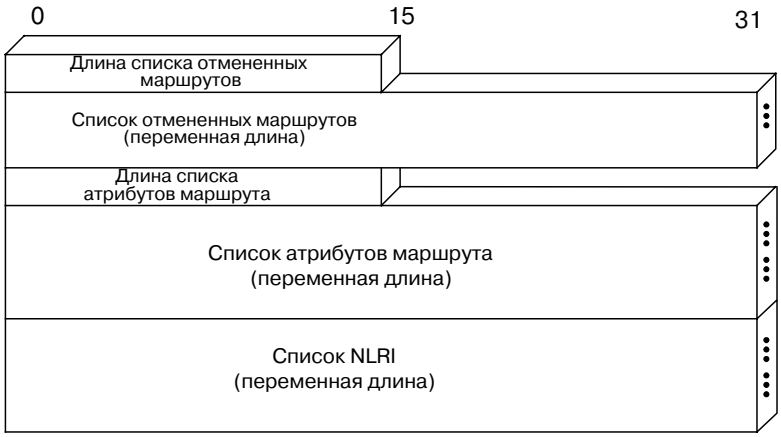


Рис. 7.3. Структура заголовка сообщения BGP «корректировка»

Третье поле сообщения — 2-байтовое поле, указывающее *общую длину списка атрибутов маршрута.* За ним следует поле переменной длины *атрибуты маршрутов*, которое содержит метрики, используемые протоколом BGP с целью назначить нужные показатели для определенных маршрутов. И, наконец, поле переменной длины с информацией сетевого уровня о достижимости NLRI содержит IP-префиксы маршрутов, которые должны быть добавлены в таблицу маршрутизации.

7.11.4. Уведомление NOTIFICATION

Сообщениями-уведомлениями BGP-маршрутизаторы обмениваются при возникновении ошибок. Если маршрутизатор сталкивается с такой проблемой, он передает сообщение NOTIFICATION, содержащее 3 поля, и затем прекращает сеанс связи. Поля уведомления содержат код ошибки, дополнительный код ошибки и данные об ошибке.

Рис. 7.4 Формат сообщения NOTIFICATION

Дополнительный код ошибки — это один октет, содержащий дополнительную информацию об ошибке. Для ошибки каждого типа существует свой набор дополнительных кодов, а если ошибке невозможно присвоить какой-либо дополнительный код, это поле заполняется нулями — «неспецифическая ошибка».

Таблица 7.1. Коды и дополнительные коды ошибок

Код ошибки	Ошибка		
1	Ошибка в заголовке	Доп. код	Значение
		1	Соединение не синхронизировано
		2	Недопустимая длина
		3	Недопустимый тип
2	Ошибка в сообщении OPEN	Доп.код	Значение
		1	Не поддерживаемый номер версии
		2	Недопустимая одноранговая AS
		3	Недопустимый BGP-идентификатор
		4	Не поддерживаемый дополнительный параметр
		5	Невозможность аутентификации
		6	Невозможное время удержания
3	Ошибка в сообщении UPDATE	Доп.код	Значение
		1	Ошибка в списке атрибутов
		2	Не распознаваемый обязательный атрибут
		3	Пропущенный обязательный атрибут
		4	Ошибка в флаге атрибута
		5	Ошибка в длине атрибута
		6	Недопустимый атрибут в Origin
		7	Зацикливание в пределах AS
		8	Невозможный атрибут Next_Hop
		9	Ошибка в необязательном атрибуте
		10	Ошибка в поле сети
		11	Ошибка в AS_PATH
4	Срабатывание таймера удержания		
5	Ошибка конечного автомата		
6	Остановка по причине, отличной от вышеуказанных		

7.11.5. Сообщение подтверждения связи Keepalive

Сообщение подтверждения связи KEEPALIVE передается, чтобы подтвердить прием сообщения OPEN, переданного равноправным маршрутизатором AS, или для удержания уже существующего, но бездействующего сеанса BGP, например, в случае, когда время удержания близко к завершению. Сообщение KEEPALIVE содержит только заголовок и обеспечивает сброс таймера удержания соединения.

7.12. Синхронизация BGP

Напомним, что BGP — это протокол маршрутизации с использованием пограничных шлюзов. Следовательно, сетевой среде необходим протокол IGP для маршрутизации данных в той части внутренней сети, которая не обслуживается протоколом BGP, например, OSPF, IS-IS или какой-либо другой внутренний протокол. Каждый из таких протоколов использует свои сообщения об обновлении. Если один маршрутизатор использует одновременно протоколы OSPF и BGP, и оба они получают данные для корректировки таблиц из разных источников, то необходима синхронизация, дающая маршрутизатору возможность определить, какие данные об обновлении следует включить в таблицу маршрутизации. Благодаря синхронизации BGP, маршрутизатор BGP может воздержаться от передачи сообщений об обновлении до тех пор, пока все маршрутизаторы не сообщат о получении данных об обновлении от IGP.

7.13. Многопротокольные расширения BGP

Протокол BGP-4 ориентирован на передачу информации маршрутизации для протокола IPv4. Многопротокольные расширения MP-BGP (MultiProtocol BGP) часто называют BGP-4+ и иногда ошибочно причисляют к групповому протоколу Multicast BGP. Многопротокольные расширения BGP-4 позволяют ему передавать информацию маршрутизации для нескольких протоколов сетевого уровня, например, IPv6, IPX, и, что особенно важно в контексте MPLS, передавать адреса VPN-IPv4. При этом обеспечивается обратная совместимость — маршрутизатор, поддерживающий расширения, может взаимодействовать с маршрутизатором, не поддерживающим эти расширения, причем только три элемента передаваемой BGP-4 информации связаны со спецификой IPv4:

- атрибут NEXT_HOP (выражен как адрес IPv4),
- AGGREGATOR (содержит адрес IPv4),
- NLRI (выражен как префикс адреса IPv4).

Чтобы BGP-4 мог поддерживать маршрутизацию для нескольких протоколов сетевого уровня, его необходимо дополнить возможностью ассоциировать конкретный протокол сетевого уровня с информацией о следующем переходе (next hop) и возможностью ассоциировать такой протокол с NLRI. Информация о следующем участке (атрибут NEXT_HOP) является значимой только в сочетании с извещением о достижимых пунктах назначения, а в сочетании с извещением о недостижимых пунктах назначения (вывод маршрутов из обслуживания) информация атрибута NEXT_HOP о следующей пересылке бессмысленна. Это значит, что извещение о достижимых пунктах назначения должно быть объединено с извещением

о следующем участке на пути к этим пунктам назначения, и что извещение о достижимых пунктах назначения должно быть отделено от извещения о недостижимых пунктах назначения.

Чтобы обеспечить обратную совместимость и упростить внедрение многопротокольных возможностей, в RFC 2858 добавлено два новых атрибута — NLRI, допускающий работу в многопротокольном режиме *(MP_REACH_NLRI),* и NLRI, запрещающий работу в многопротокольном режиме, *(MP_UNREACH_NLRI).*

Первый атрибут, MP_REACH_NLRI, используется для передачи информации о достижимых пунктах назначения и о следующем участке, который должен использоваться для пересылки данных в эти пункты назначения.

Второй атрибут, MP_UNREACH_NLRI, используется для передачи информации о недостижимых пунктах назначения. Оба эти атрибута являются необязательными и не транзитивными. Таким путем BGP-спикер, который не поддерживает многопротокольные возможности, просто игнорирует информацию, передаваемую в этих атрибутах, и не пропускает ее к другим BGP-спикерам.

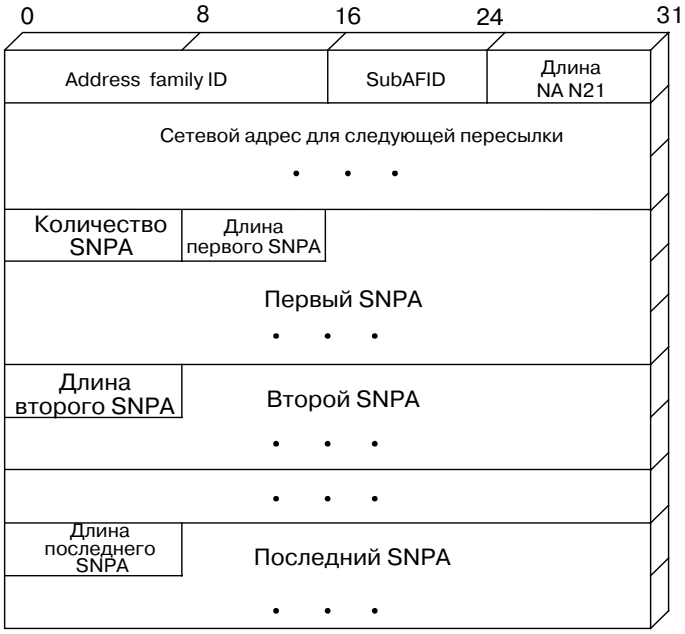


Рис. 7.5. Формат MP_REACH_NLRI

Атрибут *Multiprotocol Reachable NLRI — MP_REACH_NLRI (код типа 14)* можно использовать для следующих целей:

- известить маршрутизатор о подходящем маршруте к одноранговому объекту,
- разрешить маршрутизатору объявлять сетевой адрес маршрутизатора, который должен использоваться в качестве следующего узла на пути к пункту назначения, включенному в список поля Network Layer Reachability Information атрибута MP_NLRI,
- дать маршрутизатору возможность сообщать о некоторых или обо всех пунктах Subnetwork Points of Attachment (SNPAs), которые существуют внутри локальной системы.

Address Family Identifier (AFID) — это поле переносит идентификатор протокола сетевого уровня, ассоциированного с последующим сетевым адресом. Значения, определенные в настоящее время для этого поля, специфицированы в RFC 3232.

*Subsequent Address Family Identifie (*SubAFID*)* — это поле переносит дополнительные сведения о типе информации NLRI, передаваемой в атрибуте. Значения для этого поля представлены в табл. 7.2.

Таблица 7.2. Коды поля SubAFID

Значение	Тип информации
1	Информация NLRI, используемая для одноадресной пересылки
2	Информация NLRI, используемая для многоадресной пересылки
3	Информация NLRI, используемая как для одноадресной, так и для многоадресной пересылки

Length of Next Hop Network Address — поле, длиной 1 октет, значение которого выражает длину поля Network Address of Next Hop, измеренную в октетах.

Network Address of Next Hop — поле переменной длины, которое содержит сетевой адрес следующего маршрутизатора на пути к пункту назначения.

Number of SNPAs — поле, длиной 1 октет, несущее сведения о количестве отдельных SNPA, которые должны перечисляться в следующих полях. Значение 0 может использоваться для того, чтобы указать, что в этом атрибуте SNPA не перечисляются.

Length of Nth SNPA — поле, длиной 1 октет, значение которого выражает длину поля Nth SNPA of Next Hop, измеренную в полуоктетах.

Nth SNPA of Next Hop — поле переменной длины, содержащее SNPA маршрутизатора, сетевой адрес которого содержится в поле Network Address of Next Hop. Длина поля является целым числом октетов, а именно, округлением до целого значения половины длины SNPA, выраженной в полуоктетах; если SNPA содержит нечетное число полуоктетов, значение этого поля дополняется полуоктетом, содержащим нули.

Network Layer Reachability Information — поле переменной длины с перечислением NLRI для подходящих маршрутов, о которых объявляется в этом атрибуте.

Сообщение UPDATE, которое переносит атрибут MP_REACH_NLRI, должно также переносить атрибуты ORIGIN и AS_PATH как в EBGP, так и в IBGP. Более того, в IBGP такое сообщение должно также переносить атрибут LOCAL_PREF. Если сообщение принято от внешнего однорангового объекта, локальная система должна проверить, равно ли крайнее слева значение AS в атрибуте AS_PATH номеру автономной системы однорангового объекта, передавшего это сообщение. Если это не так, локальная система должна передать сообщение NOTIFICATION с компонентами Error Code UPDATE Message Error и Error Subcode, имеющими значение Malformed (неверный) AS_PATH.

Сообщение UPDATE, которое не переносит NLRI, кроме кодированного в атрибуте MP_REACH_NLRI, не должно переносить и атрибут NEXT_HOP. Если такое сообщение содержит атрибут NEXT_HOP, то принимающий сообщение BGP-спикер должен игнорировать этот атрибут.

Multiprotocol Unreachable NLRI — MP_UNREACH_NLRI (Type Code 15) — это атрибут, который можно использовать для вывода из обслуживания нескольких непригодных маршрутов.

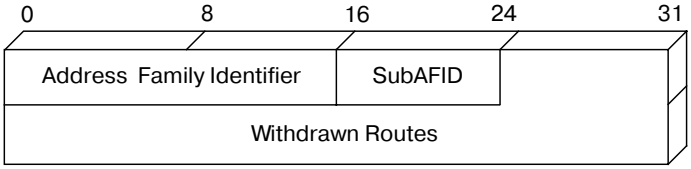


Рис. 7.6. Формат MP_UNREACH_NLRI

Здесь к вышеперечисленным добавляется поле *Withdrawn Routes* — поле переменной длины, содержащее список NLRI для маршрутов, которые должны быть выведены из обслуживания. Сообщение UPDATE, которое содержит MP_UNREACH_NLRI, не должно переносить другие атрибуты тракта.

Кодирование NLRI производится следующим образом. Информация Network Layer Reachability кодируется как один или несколько объектов в форме <длина, префикс>, поля которых описаны ниже.

Поле *длина (Length)* указывает длину префикса адреса в битах. Нулевая длина указывает префикс, который соответствует всем (как это специфицировано семейством адресов) адресам (с самим префиксом, состоящим из нулевых октетов).

Поле *префикс (Prefix)* одержит префикс адреса с «хвостовыми» битами, которые дополняют поле до величины, кратной октету. Заметим, что значения «хвостовых» битов могут быть любыми.

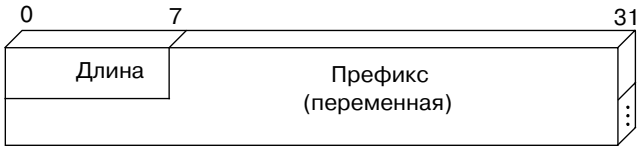


Рис. 7.7. Формат <длина-префикс>

Процедуры BGP Capability Advertisement. BGP-спикер, который поддерживает многопротокольные расширения, должен использовать процедуры BGP Capability Advertisement для того, чтобы определить, может ли он пользоваться этими расширениями при работе с тем или иным одноранговым объектом. Поле *Capability Code* имеет значение 1, что указывает Multiprotocol Extensions capabilities. Поле *Capability Length* имеет значение 4. Значение поля *Capability Value* определяется, как показано ниже:

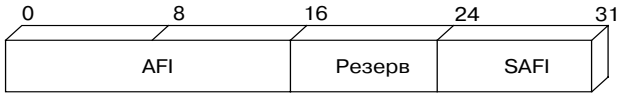


Рис. 7.8. Формат поля Capability Value

Поле *AFI — Address Family Identifier* (16 битов), кодируется так же, как в Multiprotocol Extensions.

Res. — резервное (8 битов) поле. Отправитель должен присвоить ему значение 0, а на приеме оно должно игнорироваться.

SAFI — (Subsequent Address Family Identifier) длиной 8 битов кодируется таким же образом, как в Multiprotocol Extensions.

Чтобы обеспечить двухсторонний обмен информацией о маршрутизации для определенного <AFI, SAFI> между двумя BGP-спикерами, каждый такой спикер должен известить другой (посредством механизма Capability Advertisement) о возможности поддерживать эти маршруты <AFI, SAFI>.

Ну и, наконец, об обработке ошибок. Если BGP-спикер принимает от соседа сообщение UPDATE, которое содержит атрибут MP_REACH_NLRI или атрибут MP_UNREACH_NLRI, и определяет, что этот атрибут не корректен, спикер должен удалить все маршруты BGP, сведения о которых приняты от этого соседа с такими же AFI, SAFI, как у переданного в некорректном атрибуте MP_REACH_NLRI или MP_UNREACH_NLRI. Во время сеанса BGP, в котором было принято это сообщение UPDATE, спикер должен игнорировать принимаемые данные о всех маршрутах с такими AFI, SAFI. Кроме того, спикер может прекратить сеанс BGP, в котором было принято сообщение UPDATE. Сеанс должен быть завершен кодом/дополнительным кодом уведомления BGP (сообщения NOTIFICATION), обозначающим *Update Message Error/Optional Attribute Error* — ошибка в сообщении об обновлении/ошибка в необязательном атрибуте.

Глава 8
Виртуальные частные сети и туннели

8.1. Виртуальные частные сети VPN

Несомненно, что понятие *виртуальная частная сеть VPN (Virtual Private Network)* читателю знакомо. Эта концепция появилась задолго до MPLS, а ее реализации вполне успешно выполнялись еще на базе сетей с коммутацией каналов средствами общеканальной сигнализации №7.

Общие предпосылки VPN традиционно иллюстрируются следующими типовыми примерами. Представим себе гипотетическую сеть предприятия в условиях современной глобализованной экономики. Офисы разбросаны по всему миру, сотни или тысячи сотрудников работают в командировках, сотни или тысячи сотрудников работают дома,— и все это необходимо объединить в единую сеть. Притом не просто объединить, а организовать доступ к информационным ресурсам, разделить полномочия, обеспечить надежность и безопасность. Естественно, можно проложить свои каналы, установить свои маршрутизаторы и устройства доступа, т.е. организовать собственную частную сеть связи. Предприятие, имеющее такую сеть, не зависит от Операторов сети общего пользования, само решает проблемы безопасности, доступа к услугам, может строить все что угодно и на чем угодно.

Но везде, где есть плюсы, должны быть и минусы. Любые телекоммуникационные системы, а тем более — сеть, требуют технического обслуживания, ремонта и забот, не говоря уж о развитии, модернизации и эксплуатационном управлении, о прокладке собственных кабелей связи через океан или о запуске собственных спутников связи. Даже в какой-нибудь очень богатой газодобывающей

компании рано или поздно здравый смысл приведет к взаимодействию с существующими Операторами связи с целью эффективной организации своей сети. И тут же особое значение приобретут проблемы защиты информации, аутентификации и авторизации пользователей. Наверное, в результате примерно таких рассуждений и появилась концепция виртуальных частных сетей.

Преимущества VPN очевидны: почти нет расходов на поддержание работоспособности сети (достаточно абонентской платы за арендованные каналы), удобны организация и перестроение сетевой структуры (закрылся офис в каком-то городе — расторгли договор на аренду каналов связи с этим городом). Да и сама проблема аренды каналов теряет актуальность, ведь мы ориентируемся на протокол IP, а это значит, что ресурсы разных Интернет- и сервис-провайдеров — к нашим услугам. Именно поэтому виртуальные частные сети становятся хорошей основой для поддержки электронной коммерции, хостинга приложений и других мультимедийных услуг. Но, решив более выгодно проблему организации частной сети, ее владелец сталкивается с проблемой безопасности. Доступ к ресурсам Интернет свободен, его может получить любой человек, находясь в любом месте планеты. А про хакеров, про взломы сетей и про промышленный шпионаж теперь говорят чуть ли не каждый день. Поэтому при организации VPN используются специальные устройства и механизмы защиты. Для того чтобы получить доступ к VPN, пользователь проходит через разнообразные *брандмауэры*, где производится его аутентификация и авторизация. Кроме того, для передачи VPN-трафика по сети общего пользования применяются механизмы туннелирования и шифрования трафика.

Наиболее распространенный метод создания туннелей VPN — инкапсуляция сетевых протоколов (IP, IPX, Apple Talk и т.п.) в PPP и последующая инкапсуляция образованных пакетов в протокол туннелирования. Обычно в качестве последнего выступал сам IP или, гораздо реже, технологии ATM и Frame Relay. Такой подход назывался туннелированием второго уровня, поскольку «пассажиром» здесь являлся протокол именно этого уровня.

С помощью туннелирования пакеты данных передаются через общедоступную сеть, как по обычному двухточечному соединению. Для каждой пары «отправитель–получатель» организуется своеобразный туннель — безопасное логическое соединение, позволяющее инкапсулировать данные одного протокола в пакеты другого. Новым моментом является пересылка пакетов через безопасный туннель, организованный внутри пакетной сети общего пользования. Виртуальная частная сеть предоставляет удаленным пользователям доступ к корпоративной сети непосредственно через Интернет. Пользователю достаточно установить соединение с Интернет-провайдером, и соответствующее программное

обеспечение на его компьютере организует туннель через сеть провайдера к Интернет-шлюзу корпоративной сети. При этом провайдер вовсе не в курсе того, что организуется туннель, и не предпринимает никаких действий для его поддержки. Аутентификация пользователей в корпоративной сети и обеспечение целости и конфиденциальности туннелируемых данных выполняются брандмауэром с помощью *протокола шифрования «точка-точка» корпорации Майкрософт MPPE (Microsoft Point-to-Point Encryption)* или уже упоминавшегося выше *протокола IPSec*.

Существует множество разновидностей виртуальных частных сетей. Их спектр варьируется от сетей Интернет-провайдеров, позволяющих управлять обслуживанием клиентов непосредственно на их площадях, до корпоративных VPN, разворачиваемых и управляемых самими корпорациями. Принято выделять три основных вида виртуальных частных сетей:

- VPN с удаленным доступом (Remote Access VPN),
- внутрикорпоративные VPN (Intranet VPN) и
- межкорпоративные VPN (Extranet VPN).

Принцип работы *VPN с удаленным доступом* прост: пользователи устанавливают соединения с местной точкой доступа к глобальной сети (PoP), после чего их потоки данных туннелируются через Интернет, что позволяет, в частности, избежать платы за междугородную или международную связь. Затем все эти потоки концентрируются в соответствующих узлах и передаются в корпоративные сети. Вследствие использования в качестве объединяющей магистрали открытой сети Интернет механизмы защиты информации становятся жизненно важными элементами такой технологии.

Intranet VPN представляет собой самый простой вариант VPN: корпорации, нуждающиеся в организации для своих филиалов и отделений доступа к централизованным хранилищам информации, используют вместо выделенных линий дешевую связь с этими хранилищами через Интернет.

Extranet VPN — это сетевая технология, которая обеспечивает прямой доступ из сети одной компании к сети другой компании, способствуя тем самым повышению надежности связи, устанавливаемой для делового сотрудничества. Сети Extranet VPN, в целом, похожи на внутрикорпоративные виртуальные частные сети, с той лишь разницей, что проблема защиты информации является для них более острой. Когда несколько компаний принимают решение работать вместе и открывают друг для друга свои сети, они должны заботиться о том, чтобы их новые партнеры имели доступ только к определенной информации. Конфиденциальная же информация должна быть надежно защищена от несанкционированного использования. Именно поэтому в межкорпоративных сетях боль-

шое значение должно придаваться контролю доступа посредством брандмауэров (Firewalls). Важна и аутентификация пользователей, призванная гарантировать, что доступ к информации получат только те, кому он действительно разрешен.

В дополнение к типам технологий, используемым для реализации VPN, эти сети разделяются также по уровням модели OSI, в которых они работают. Существуют две основные группы — сети VPN уровня 2 и сети VPN уровня 3,— хотя некоторые специалисты считают, что указатели URL с префиксами https:// позволяют создавать VPN уровня 4. Сети MPLS-VPN в этой классификации образуют новый тип VPN уровня 2.5, что уже обсуждалось в главе 1.

Если рассмотренные в этом параграфе основы концепции VPN наложить на принципы работы MPLS, изложенные в предыдущих главах книги, то выводы аналитиков (исследования компании Intronetics, например) о том, что наибольшее распространение в качестве основы VPN получила технология MPLS, будут очевидны. Действительно, технология MPLS — это экономичная поддержка масштабируемых услуг VPN в IP-сети. Возможности многопротокольной коммутации по меткам, предусматривающие работу без создания виртуального соединения, средства обеспечения QoS и рассматриваемый в следующей главе инжиниринг трафика создают огромные возможности наращивания VPN в инфраструктуре предприятия без ущерба для производительности сети. Виртуальным частным сетям MPLS-VPN и посвящена вся эта глава, однако прежде необходимо рассмотреть вопросы туннелирования в MPLS-сетях.

8.2. Туннелирование в MPLS

Организация в виртуальном канале туннелей реализуется во многих современных протоколах и базируется на инкапсуляции пакетов, направляемых по туннелю, в пакеты, следующие по «основному» виртуальному каналу и имеющие тот же адрес назначения, что и инкапсулируемые в них пакеты. При этом туннели могут создаваться как по принципу hop-by-hop, так и по принципу использования заранее определенных маршрутов.

Применительно к MPLS мы будем говорить о так называемых *LSP-туннелях*, которые образуются не путем инкапсуляции пакетов, а с помощью средств коммутации по меткам. LSP-туннель представляет собой последовательность маршрутизаторов, где первый маршрутизатор является входным конечным пунктом туннеля, а последний — его выходным конечным пунктом. Чтобы направить пакет в LSP-туннель, маршрутизатор входного конечного пункта туннеля помещает метку, назначенную для этого туннеля, на верх существующего в пакете стека меток (заметим, что

предпоследний маршрутизатор LSP-туннеля может уничтожить верхнюю метку в стеке до передачи пакета к выходному конечному пункту туннеля).

LSP-туннель создается внутри LSP, как это показано на рис. 8.1. Начало и/или конец туннеля, как правило, не совпадают с началом и/или концом этого LSP, туннель обычно бывает короче LSP, в котором он создан. Так, на рис. 8.1. изображен LSP, составленный LSR1, LSR2, LSR3, LSR4, LSR5, LSR6 и LSR7, и в данном LSP из семи маршрутизаторов построен LSP-туннель из четырех маршрутизаторов, в котором LSR2 является входным пунктом туннеля, а LSR5 — выходным.

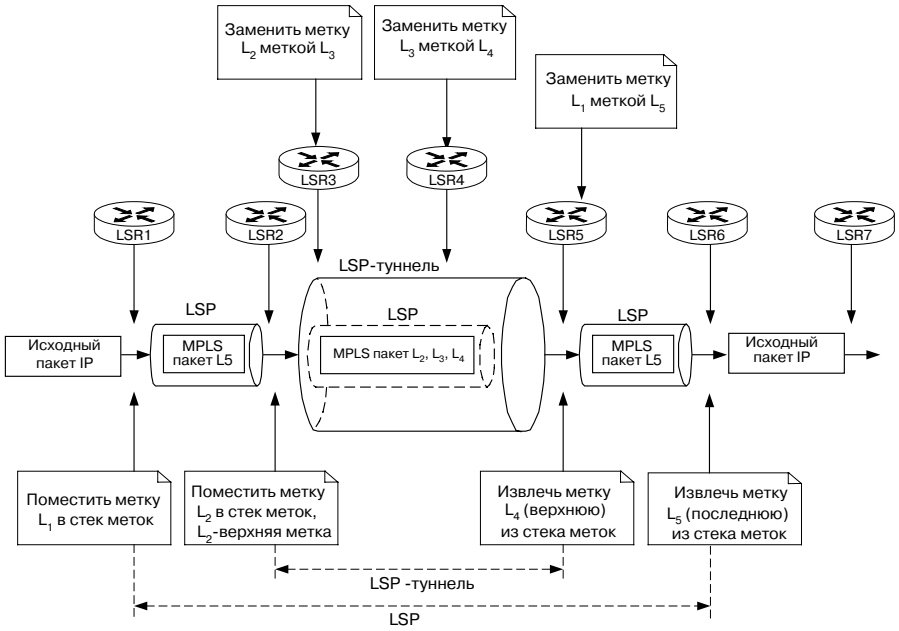


Рис. 8.1. LSP-туннель между LSR2 и LSR 5

В пределах одного LSP может быть создано несколько LSP-туннелей одного уровня с несовпадающими входными и/или выходными конечными пунктами. Более того, внутри любого из этих LSP-туннелей можно создавать LSP-туннели следующего уровня. Количество таких уровней, по тем или иным причинам, не может быть сколь угодно велико, однако иерархичность организации туннелей в MPLS вполне очевидна. Обеспечивается она с помощью стека меток, уже упоминавшегося в главе 2 и более подробно рассматриваемого ниже. Механизм стека меток позволяет, в частности, использовать MPLS для выполнения одновременно как маршрутизации внутри сети одного Интернет-провайдера, так и маршрутизации между доменами. Напомним, что описанный в главе 2

стек меток MPLS организован по принципу LIFO (last-in, first-out), то есть метка, полученная последней, находится на верху стека, и только она обрабатывается при пересылке пакета. То обстоятельство, что эта метка могла прежде не быть верхней, т.е. что над ней могли находиться другие метки, а также что в данный момент под этой меткой в стеке имеются другие метки, не имеет значения. Как было показано в параграфе 2.1, чтобы указать, что *глубина стека* превышает 1, используется *бит S* в заголовке метки. Одно из двух возможных значений этот бит имеет в самой нижней метке стека, а второе — во всех остальных метках. Одной из причин появления в стеке еще одной метки, может быть то, что некий LSR_i, находящийся на пути следования пакета с определенным FEC_j, имеет полные или частичные сведения о том, к какому FEC_k будет отнесен и с какой меткой L_l будет пересылаться этот пакет тому LSR_n, который для текущего FEC_j является последним (то есть конечным). В этом случае LSR_i может поместить в пакет метку L_l, а над ней — метку, которая в нем связана с FEC_j. Но основное назначение стеков меток состоит в том, чтобы поддерживать в MPLS-сети упоминавшуюся выше древовидность множества трактов LSP, заканчивающихся в одном входном LSR, и в том, чтобы использовать метки при создании LSP-туннелей. Об LSP-туннелях еще будет говориться в этом же параграфе несколько позже, но прежде рассмотрим некоторые другие аспекты использования меток.

С поддержкой древовидности дело обстоит, в принципе, довольно просто. Если в одном LSP сливается несколько потоков (каждый поток — со своим FEC и со своей меткой), то этот LSP не заменяет метки, связанные с названными потоками, а оставляет их, помещая сверху метку нового FEC, который соответствует объединенному потоку пакетов, образующемуся в результате слияния. Поскольку дерево ветвится, в каком-то другом LSR, находящемся ближе к корню, происходит еще одно слияние нескольких объединенных потоков, и в стеке появляется еще одна метка.

Под *LSP уровня m* понимается LSP, который образован последовательностью маршрутизаторов $<LSR_{вх}, LSR_2, \ldots, LSR_{n-1}, LSR_{вых}>$ со следующими свойствами:

- входной маршрутизатор $LSR_{вх}$ помещает в стек меток обрабатываемого пакета такую по счету метку, что стек приобретает глубину *m*;
- при всех i ($1<i<n$) пакет, поступающий к LSR_i, имеет стек меток глубины *m*;
- в процессе транспортировки пакета от $LSR_{вх}$ к LSR_{n-1} глубина его стека меток никогда не бывает меньше *m*;
- при всех i ($1<i<n$) LSR_i передает пакет LSR_{i+1} средствами MPLS.

Иными словами, маршрут LSP *уровня m* представляет собой последовательность маршрутизаторов, которая начинается с входного LSR, помещающего в пакет метку уровня *m*, содержит проме-

жуточные LSR, каждый из которых принимает решение о пересылке пакета на основе метки уровня *m*, и заканчивается выходным LSR, где решение о пересылке принимается на основе метки уровня *m* — *k*, где *k*>0, или на основе обычных (не-MPLS) процедур пересылки. Отметим еще раз, что от LSR_{n-1} к LSR_n можно передавать пакеты со стеком меток глубины (*m*-1), поскольку метка уровня *m* выходному LSR не требуется. Это позволяет избавить выходной LSR от операций анализа ненужной ему метки и не требует никаких дополнительных операций в предпоследнем LSR, кроме простого уничтожения верхней метки в стеке.

Сказанное выше про LSP уровня *m* справедливо и по отношению к LSP-туннелю уровня *m*. Подчеркнем лишь, что в этом случае входной, промежуточные и выходной LSR являются таковыми не для всего LSP, а для LSP-туннеля, причем конечные пункты LSP-туннеля и LSP, в котором организован туннель, могут не совпадать. А поскольку в отношении правил работы с метками LSP-туннель ничем не отличается от обычного LSP, в MPLS имеется возможность создавать туннели внутри туннелей, т.е. в сети MPLS могут создаваться LSP-туннели практически любой сложности.

Путем создания туннелей через несколько сетевых сегментов достигается уникальная возможность MPLS обеспечивать транспортировку пакета, не специфицируя явно промежуточные маршрутизаторы в этих сегментах. Чтобы показать, как это делается, рассмотрим схему, представленную на рис. 8.2.

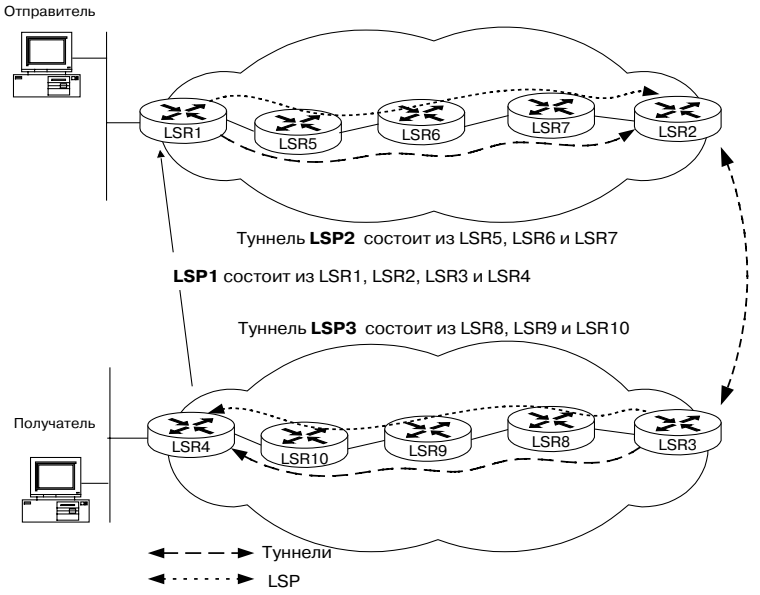


Рис. 8.2. Туннели в сети MPLS

Пусть здесь все пограничные маршрутизаторы (LSR1, LSR2, LSR3 и LSR4) используют рассмотренный в главе 7 протокол BGP (далее в этой главе мы обсудим также расширение протокола BGP для построения VPN на основе MPLS) и создают коммутируемый по меткам тракт LSP1. Маршрутизатор LSR1 знает о том, что следующий пункт назначения — LSR2. В свою очередь, LSR2 знает о том, что для него следующий пункт назначения — LSR3, а LSR3 знает, это же о LSR4. Эти пограничные LSR будут использовать протокол LDP для получения и хранения меток от выходного LSR4 вплоть до входного LSR1.

Однако, чтобы данные были переданы от LSR1 к LSR2, они должны пройти через несколько транзитных маршрутизаторов LSR (в представленном на рис. 8.2 случае — через три: LSR5, LSR6 и LSR7). Между LSR1 и LSR2 создается отдельный тракт LSP2 в виде LSP-туннеля, связывающего эти два LSR и проходящего через LSR5, LSR6 и LSR7. Метки в LSP-туннеле отличаются от меток, которые LSR создали для LSP1. Это повторено и между LSR3 и LSR4, т.е. во втором сегменте создается тракт LSP3, тоже представляющий собой LSP-туннель.

Для передачи пакета от LSR1 к LSR4 через два сетевых сегмента используется изложенная выше концепция стека меток. Поскольку пакет должен следовать через LSP1, LSP2 и LSP3, он будет переносить одновременно две разные метки. Пары меток, используемые в каждом сегменте, следующие: первый сегмент — метка для LSP1 и метка для LSP-туннеля LSP2, второй сегмент — метка для LSP1 и метка для LSP-туннеля LSP3 и т.д. Маршрутизатор LSR4, приняв пакет, до отправки его адресату удаляет из стека обе метки.

Рассмотрим теперь отличительные особенности туннелей MPLS. Благодаря особенностям технологии MPLS эти туннели позволяют передавать блоки данных любого протокола (IP, IPX, кадры Frame Relay, ячейки ATM и др.), так как содержимое пакетов вдоль всего пути их следования остается неизменным, а изменяются только метки. В отличие от туннелей MPLS, туннели *IPSec*, например, поддерживают передачу только пакетов протокола IP, а туннельные протоколы 2-го уровня *PPTP* и *L2TP,* которые были предложены в IETF компаниями Microsoft и Cisco в виде конкурирующих спецификаций, позволяют обмениваться данными по протоколам IP, IPX и NetBEUI.

Безопасность передачи данных в туннелях MPLS обеспечивается за счет определенной сетевой стратегии, запрещающей принимать от непроверенных источников как пакеты, снабженные метками, так и маршрутную информацию VPN-IP. Она может быть также повышена применением стандартных средств аутентификации и/или шифрования, например, IPSec. В *протокол IP Security* включены процедуры шифрования IP-пакетов, аутентификации, обеспечения защиты и целости данных при транспортировке, вследствие чего

туннели IPSec обеспечивают надежную доставку информационного трафика. *Протокол туннельного соединения 2-го уровня L2TP (Layer 2 Tunnelling Protocol)* также поддерживает процедуры аутентификации, а *протокол туннельного соединения «точка-точка» PPTP (Point-to-Point Tunneling Protocol),* помимо этих функций, снабжен и функциями шифрования. Применение же меток MPLS позволяет ускорить продвижение всех пакетов по сети, а поскольку MPLS не требует читать заголовки транспортируемых пакетов, используемая в этих пакетах адресация может носить любой характер.

При маршрутизации пакета в MPLS учитываются различные параметры, оказывающие влияние на выбор маршрута. Совместная работа технологии многопротокольной коммутации и механизмов инжиниринга трафика, которые подробно рассматриваются в следующей главе, позволяет предоставить требуемое QoS в каждом LSP-туннеле за счет резервирования ресурсов для этого туннеля в каждом из LSR, через которые он проходит. Помимо этого, появляется возможность отслеживать маршрут, по которому проходит каждый пакет, использующий сформированный туннель, а также возможность диагностики и административного контроля туннелей LSP.

Туннели между двумя точками, различающиеся требованиями к QoS, поддерживает и протокол L2TP. Но технология VPN IPSec не поддерживает QoS в установленном соединении, а протокол PPTP поддерживает только один туннель между двумя точками.

8.3. Виртуальные частные MPLS-сети

8.3.1. Общие предпосылки MPLS-VPN

Все рассмотренные в начале главы сети VPN обладают общими функциональными возможностями: масштабируемостью, наличием средств управления, защитой и способностью обработки частных адресов.

Масштабируемость — крайне полезное для VPN свойство, поскольку эти сети часто нуждаются в расширении с ростом бизнеса предприятия. Сети VPN должны быть *управляемыми*, чтобы их можно было конфигурировать и контролировать в соответствии с быстро изменяющимися бизнес-процессами предприятия. Важным может оказаться также управление учетом предоставляемого обслуживания для начисления платы и для других целей. *Защита* является свойством, которое оправдывает букву P (private) в аббревиатуре VPN и становится ключевым элементом в корпоративных сетевых коммуникациях в условиях существования угрозы хищения информации и атак на интеллектуальную собственность. Поскольку большинство VPN в настоящее время ориентируются на IP, важна

также и возможность *обработки частных адресов*, чтобы гарантировать уникальность IP-адреса.

Обсуждавшееся в предыдущем параграфе понятие LSP-туннелей составляет важный аспект MPLS-VPN, т.к. прилагательное «частный» в MPLS относится к физическому разделению трафика между LSP-туннелями. Это весьма похоже на способ виртуальных каналов, которые организуются для VPN-приложений ATM и FR. В настоящее время в области MPLS-VPN существуют два главных направления: *BGP/MPLS-VPN* и *VPN с виртуальными маршрутизаторами (VR) на базе IP.* Оба эти направления соответствуют общей модели MPLS-VPN, представленной на рис. 8.3.

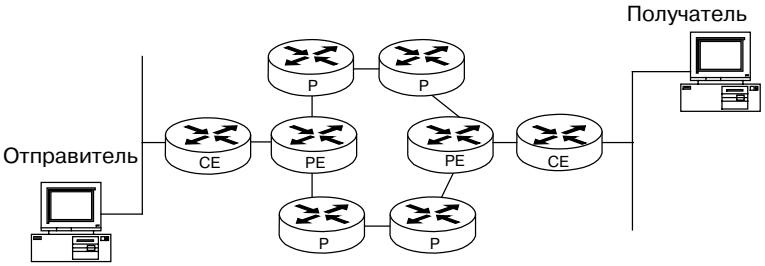


Рис. 8.3. Общая модель MPLS-VPN

Ядро сети строится на базовых маршрутизаторах MPLS, называемых *внутренними маршрутизаторами провайдера P,* и взаимодействует с пользователем VPN не прямо, а через соединение между *пограничным маршрутизирующим устройством заказчика CE (Customer Edge router)* и *пограничным маршрутизирующим устройством провайдера PE (Provider Edge router)*. CE могут быть статически подключены к PE закрепленными каналами или использовать коммутируемые линии связи. Для подключения CE к PE могут использоваться каналы любого типа, например, ATM, FR, Ethernet, PPP, а также такие механизмы туннелирования, как уже упоминавшиеся IPSec, L2TP или *GRE (Generic Route Encapsulation)*.

С точки зрения функциональных возможностей VPN оба метода организации MPLS-VPN сходны. При одном методе для передачи пакетов через ядро MPLS с помощью протокола BGP создаются специальные расширенные адреса, а при другом методе VR хранят отдельные для каждой VPN таблицы MPLS-пересылок. Фактическая же реализация этих двух методов совершенно различна, а выбирает метод провайдер услуг VPN с учетом возможностей оборудования, ситуации в части взаимодействия сетей и других факторов. Создана новая рабочая группа IETF, названная *Provider-Provisioned VPNs (PPVPNs)*, которая разрабатывает структуру и соответствующие спецификации для сетей VPN этих двух типов.

8.3.2. Сети MPLS/BGP-VPN

Модель MPLS/BGP-VPN базируется на уже упоминавшихся в главе 7 расширениях протокола маршрутизации внешнего шлюза BGP, называемых многопротокольными расширениями BGP и касающихся специальных расширенных адресов. Эти адреса используются для обмена информацией о доступности между маршрутизаторами PE только в одной и той же VPN. Каждый маршрутизатор PE в MPLS/BGP-VPN поддерживает отдельную *таблицу маршрутизации VRF (VPN Routing and Forwarding table)*. Каждая VRF содержит все маршруты для одного пользователя одной VPN, что обеспечивает эффективную изоляцию сетей. Такая модель позволяет разным предприятиям использовать перекрывающиеся множества частных IP-адресов.

Основная цель реализации MPLS-VPN этого типа — позволить провайдеру создать нужную заказчику конфигурацию VPN. Таким заказчиком может быть предприятие, группа предприятий, которым нужна EXTRANET, другой сервис-провайдер, или даже другой провайдер VPN, который может использовать это MPLS-приложение с целью построить сеть VPN для своих собственных клиентов. Эта модель VPN позволит заказчикам использовать сетевые услуги масштабируемо и гибко, а провайдеру — упростить менеджмент, защиту и обслуживание VPN. Спецификация MPLS/BGP-VPN впервые была опубликована как RFC 2547 «BGP/MPLS VPNs», а позже пересмотрена и усовершенствована рабочей группой.

8.3.3. Виртуальная сеть на базе IP/MPLS

Другая популярная модель MPLS-VPN связана с использованием маршрутизаторов VR. Маршрутизатор VR является результатом разделения физического маршрутизатора на несколько логических (виртуальных) маршрутизаторов. Каждый VR поддерживает для каждой VPN свою независимую таблицу. При таком решении для поддержки обмена информацией о доступности маршрутизатора VR протокол BGP не требуется.

Каждый VR использует существующие механизмы и инструменты для конфигурации, начисления платы и техобслуживания. Маршрутизаторы VR могут либо совместно использовать физические ресурсы маршрутизации, либо использовать отдельные объекты так, чтобы каждый VR имел свой объект маршрутизации для распределения информации о доступности VPN среди VR, участвующих в работе этой VPN. При этом VPN может использовать любой протокол маршрутизации, а для получения необходимой доступности VPN не требуется никакого специального расширения этого протокола. Такая конфигурация VPN может быть очень гибкой в том отношении, как могут быть соединены маршрутизаторы VR в опорной сети; эта архитектура допускает различные сценарии развертывания опор-

ной сети: провайдер MPLS-VPN может быть владельцем опорной сети, или получать услуги опорной сети от одного или нескольких других провайдеров MPLS.

8.3.4. Организация MPLS-VPN

В общем случае у клиента может быть несколько территориально обособленных IP-сетей, каждая из которых, в свою очередь, может включать в себя несколько подсетей, связанных маршрутизаторами. Такие территориально изолированные сетевые элементы корпоративной сети принято называть *сайтами*. Принадлежащие одному клиенту сайты обмениваются IP-пакетами через сеть провайдера и образуют виртуальную частную сеть этого клиента. Как было показано при обсуждении рис. 8.3, каждый сайт имеет один или несколько пограничных пользовательских маршрутизаторов CE, соединенных с одним или более пограничными маршрутизаторами провайдера PE посредством каналов PPP, ATM, Ethernet, Frame Relay и т.п.

CE-маршрутизаторы разных сайтов не обмениваются маршрутной информацией непосредственно и даже могут не знать друг о друге. Адресные пространства подсетей, входящих в состав VPN, могут перекрываться, т.е. уникальность адресов должна соблюдаться только в пределах одной подсети. Этого удается добиться путем преобразования IP-адреса в VPN-IP-адрес и использования для работы с этими адресами протокола MP-BGP.

Каждый PE-маршрутизатор должен поддерживать столько таблиц маршрутизации, сколько сайтов пользователей к нему подсоединено, то есть в одном физическом маршрутизаторе организуется несколько виртуальных. При этом маршрутная информация, касающаяся какой-то одной VPN, содержится только в PE-маршрутизаторах, к которым подсоединены сайты этой VPN. Таким образом решается проблема масштабирования, неизбежно возникающая в случае, если эту информацию должны хранить все маршрутизаторы сети оператора. Считается, что CE-маршрутизатор относится к одному сайту, но сам сайт может принадлежать нескольким VPN. К PE-маршрутизатору может быть подключено несколько CE-маршрутизаторов, находящихся в разных сайтах и даже относящихся к разным VPN.

8.4. Маршрутизация MPLS-VPN

8.4.1. Таблицы маршрутизации в PE-маршрутизаторах

Как уже говорилось, PE-маршрутизаторы поддерживают для каждого подключенного сайта одну ассоциированную с ним таблицу маршрутизации. Рассмотрим сеть, в которой существуют три

PE-маршрутизатора — PE1, PE2 и PE3. Каждый из них соединяется с одним из CE-маршрутизаторов — CE1, CE2 и CE3, соответственно. При этом CE1, CE2 и CE3 принадлежат сайтам, входящим в одну VPN. В таком случае PE1 использует протокол MP-BGP, чтобы передать PE2 и PE3 сведения о маршрутах, которые он получил от CE1. В свою очередь, PE2 и PE3 вносят данные об этих маршрутах в свои таблицы маршрутизации, ассоциированные с сайтами, в которых находятся CE2 и CE3. Если сайт принадлежит нескольким VPN, то соответствующая ему таблица маршрутизации в PE может содержать данные о маршрутах всех этих VPN.

Если PE-маршрутизатор получит от сайта пакет с адресом, которого нет в ассоциированной таблице маршрутизации, то либо он отбросит такой пакет, либо, если оператор предоставляет услуги доступа в Internet через эту VPN, произойдет обращение к таблице маршрутизации Internet.

Чтобы обеспечивалась изоляция одной VPN от другой, важно, чтобы ни один из маршрутизаторов, составляющих магистральную сеть MPLS (Backbone Router), не принимал пакеты с метками от маршрутизаторов, не относящихся к этой сети, за исключением случая, когда верхняя метка стека уже была присвоена маршрутизатором, входящим в магистральную сеть MPLS, и притом обнаруживается, что использование этой метки приведет к уходу пакета из сети до того, как будут обработаны остальные метки стека и произведен анализ IP-заголовка.

Ассоциированные таблицы маршрутизации в PE используются только для пакетов, которые получены от сайтов, непосредственно соединенных с PE, и, как результат, может существовать несколько разных маршрутов к одной системе, причем маршрут будет определяться сайтом, из которого пакет попал в магистральную сеть.

В некоторых случаях сайт клиента может быть разделен им на несколько виртуальных сайтов, например, с использованием VLAN. Такие виртуальные сайты могут входить в разные VPN, и PE должен поддерживать таблицы маршрутизации для каждого виртуального сайта, как это описано в документе RFC 2917.

8.4.2. Распространение маршрутной информации по протоколу BGP

PE-маршрутизаторы используют протокол BGP для передачи друг другу маршрутной информации. Для каждого адресного префикса BGP-спикер может определить и объявить только один маршрут, но любая VPN может иметь собственное адресное пространство, и один и тот же адрес может использоваться в разных VPN. Для того чтобы устранить это противоречие, было специфицировано новое *семейство адресов VPN-IPv4 Address Family*.

Адрес VPN-IPv4 имеет длину 12 байтов, первые восемь из которых занимает префикс, называемый *различителем маршрутов RD (Route Distinguisher)*, а оставшиеся 4 байта содержат IPv4 адрес (рис.8.4). Даже если в двух VPN будут совпадающие IPv4-адреса, PE-маршрутизатор преобразует их в уникальные адреса VPN-IPv4. Таким образом решается задача определения разных маршрутов к устройствам, имеющим один и тот же IP-адрес, но принадлежащим разным VPN.

Сам RD не является информативным и служит лишь для создания нескольких отдельных маршрутов по общему адресному префиксу IPv4. Он может быть использован также для определения нескольких разных маршрутов к одному и тому же сетевому устройству путем создания двух различных адресов VPN-IPv4, имеющих общую IPv4-часть.

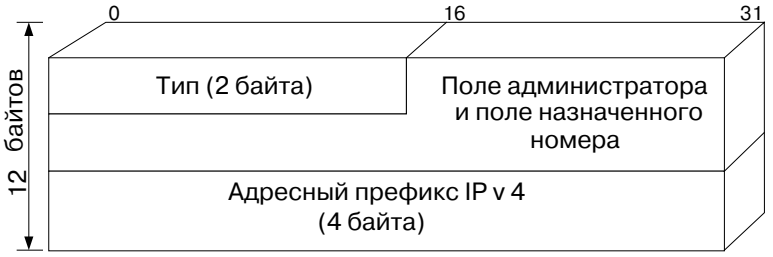


Рис. 8.4. Кодирование Route Distinguisher

Различители маршрутов RD имеют структуру, позволяющую каждому провайдеру создавать и модифицировать собственное «номерное пространство», не конфликтуя при этом с RD, назначенными другими провайдерами. RD состоит из 2-байтового поля типа, поля администратора (Administrator) и поля назначенного номера (Assigned Number). При этом значение поля типа определяет длину обоих следующих полей, а также семантику поля администратора. Структура различителя маршрутов полностью игнорируется протоколом BGP, когда он сравнивает два таких адресных префикса. Она имеет смысл только для провайдера услуг. Если же поля администратора и назначенного номера заполнены нулями, то адрес имеет то же значение, что и обыкновенный IPv4-адрес.

Любая ассоциированная таблица маршрутизации для любого заданного IPv4-префикса будет иметь данные только об одном маршруте VPN-IPv4. Когда адресу назначения ставится в соответствие маршрут VPN-IPv4, то соотносится только IPv4-часть. PE-маршрутизатор должен ассоциировать маршруты, ведущие к определенному CE, с определенным RD. При этом PE может быть сконфигурирован так, чтобы ассоциировать все маршруты, ведущие к одному CE, с одним RD или с разными RD.

8.5. Распространение маршрутной информации

8.5.1. Атрибут целевой VPN

Каждой ассоциированной таблице маршрутизации в PE-маршрутизаторах присваивается один или несколько *атрибутов целевой VPN (Target VPN)*. Когда PE-маршрутизатором создается маршрут VPN-IPv4, он ассоциируется с одним или более атрибутами Target VPN, которые переносятся протоколом BGP как атрибуты маршрута. Каждый маршрут, ассоциированный с Target VPN «T», должен быть объявлен всем PE-маршрутизаторам, которые имеют таблицу маршрутизации, ассоциированную с Target VPN «T». Когда данные о таком маршруте принимаются PE-маршрутизатором, этот маршрут имеет право быть включенным в любую из ассоциированных таблиц маршрутизации, имеющих атрибут Target VPN «T».

Существует набор атрибутов Target VPN, которыми PE-маршрутизатор снабжает маршруты, полученные от сайта «S». Одновременно с этим есть набор атрибутов Target VPN, который PE-маршрутизатор использует для того, чтобы принять решение о праве маршрута, данные о котором получены от другого PE, быть включенным в таблицу маршрутизации, ассоциированную с сайтом «S». Эти два набора атрибутов независимы и могут быть разными.

Функции, выполняемые атрибутом Target VPN, схожи с функциями BGP Communities Attribute, однако формат последнего имеет только 16-разрядное номерное пространство.

Когда BGP-спикер получает данные о двух маршрутах для одного адресного префикса VPN-IPv4, он выбирает один из них согласно традиционным правилам предпочтения маршрутов в протоколе BGP.

Следует отметить, что маршрут может иметь только один относящийся к нему префикс RD, но несколько атрибутов Target VPN. Если существует один маршрут с несколькими атрибутами, масштабируемость в BGP улучшается по сравнению с тем, когда имеется несколько отдельных маршрутов. Всегда остается возможность убрать атрибут Target VPN, создав большее количество маршрутов, но ухудшив характеристику масштабируемости.

Существует несколько реализуемых PE подходов к определению атрибутов Target VPN, которые необходимо ассоциировать с определенным маршрутом. PE может быть сконфигурирован так, чтобы ассоциировать все маршруты, ведущие к некоторому сайту, с определенным атрибутом Target VPN, или так, чтобы ассоциировать часть маршрутов, ведущих к этому сайту, с одним Target VPN, а другую часть — с другим Target VPN. Возможен также вариант, когда CE-маршрутизатор, отправляя маршрутную информацию к PE-маршрутизатору, задает для каждого маршрута один или несколько атрибутов Target VPN. Последний вариант передает поль-

зователю контролирующие механизмы VPN. При этом желательно оставить PE-маршрутизатору возможность исключать любые Target VPN, запрещенные его конфигурацией, а также добавлять такие Target VPN, которые являются для PE обязательными.

8.5.2. Атрибут VPN-источник

Маршрут VPN-IPv4 может быть опционально ассоциирован с атрибутом *VPN-источник (VPN of Origin)*. Этот атрибут однозначно идентифицирует группу сайтов и соответствующий маршрут, который объявлен одним из маршрутизаторов, находящихся в этих сайтах. Одним из вариантов применения этого атрибута может быть идентификация предприятия, владеющего сайтом, к которому ведет маршрут, или сети Intranet, которой он принадлежит. По мнению авторов, точнее было бы назвать этот атрибут источником маршрута, так как определяемая им группа сайтов не обязательно составляет VPN.

8.5.3. Атрибут сайт-источник

Атрибут *Site of Origin* уникальным образом идентифицирует сайт, от которого PE-маршрутизатор получил информацию об определенном маршруте. Все маршруты, данные о которых получены от определенного сайта, должны быть ассоциированы с определенным значением этого атрибута, даже если сайт имеет несколько соединений с одним PE или соединен с несколькими PE. Для разных сайтов должны использоваться разные атрибуты Site of Origin.

8.5.4. Передача маршрутной информации между PE

Если два сайта принадлежат одной автономной системе AS, то их PE-маршрутизаторы могут обмениваться друг с другом маршрутной информацией по протоколу IBGP непосредственно или, когда их много, через *отражатель маршрутов*. Отражатели маршрутов позволяют сообщать полученные от маршрутизаторов IBGP данные о маршрутах другим маршрутизаторам IBGP, т.е. «отражать» их, уменьшая тем самым количество связей внутри AS.

Если же сайты находятся в разных AS (например, если они подсоединены к разным провайдерам), PE-маршрутизатор должен будет передать маршрутную информацию по протоколу IBGP к пограничному маршрутизатору ASBR или к отражателю маршрутов, клиентом которого является ASBR. Затем этот ASBR должен будет передать по протоколу EBGP данные о маршрутах к ASBR, находящемуся в другой AS. Таким образом, становится возможным подключение сайтов к разным провайдерам, но между последними должно быть предварительно заключено доверительное соглашение, так как данные о маршрутах из глобальной сети Интернет по умолчанию не принимаются.

Когда PE-маршрутизатор распространяет данные о маршрутах по протоколу BGP, он использует в качестве параметра BGP_Next_Hop свой собственный адрес. Он также назначает и распределяет метки MPLS. Фактически PE распространяет данные не о маршрутах VPN-IPv4, а о «помеченных» маршрутах VPN-IPv4, но на текущий момент документы IETF, специфицирующие перенос меток MPLS протоколом BGP-4, имеют статус *draft*. Когда PE обрабатывает полученный пакет, имеющий такую метку наверху стека, он извлекает стек и отправляет пакет сайту, к которому ведет маршрут. Обычно это означает, что он пошлет пакет CE-маршрутизатору, объявившему ему этот маршрут. Метка может также определять инкапсуляцию трафика звена данных.

В большинстве случаев метка, присвоенная PE-отправителем, обеспечит доставку пакета прямо к CE, и PE-получателю не придется анализировать адрес назначения. Однако возможно также присвоение метки, однозначно идентифицирующей ту таблицу маршрутизации, к которой PE-получателю необходимо обратиться для дальнейшей пересылки пакета. За счет того, что при построении MPLS-VPN допускается использование любой стандартной архитектуры на основе отражателей маршрутов, а также за счет того, что PE-маршрутизаторы принимают объявления только о таких маршрутах, которые относятся к связанным с ними VPN, а P-маршрутизаторы, т.е. маршрутизаторы MPLS-ядра, вообще не принимают сообщений о VPN-IPv4 маршрутах, достигается исключительная масштабируемость MPLS-VPN решений.

8.6. Пересылка данных по магистральной сети

Маршрутизаторы магистральной сети не имеют сведений о маршрутах к сайтам VPN, а пакеты пересылаются по технологии MPLS с двухуровневым стеком меток. PE-маршрутизаторы (а также ASBR, распространяющие адреса VPN-IPv4) должны добавлять свои адресные префиксы в маршрутные таблицы IGP магистральной сети. Это позволяет MPLS назначить в каждом узле магистральной сети метку для маршрута к каждому из PE.

Получив пакет от одного из CE, PE-маршрутизатор обращается к ассоциированной таблице маршрутизации. Если пакет предназначается CE-узлу, подключенному непосредственно к рассматриваемому PE, он немедленно пересылается этому маршрутизатору. В противном случае анализируется поле BGP_Next_Hop, а также находится и помещается на верх стека соответствующая ему метка. Затем PE ищет IGP-маршрут к BGP_Next_Hop и метку, соответствующую следующей пересылке IGP. Эта метка также помещается на верх стека, и пакет отправляется на адрес следующей пересылки IGP (в случае, если адреса следующего узла для IGP и BGP

маршрутов совпадают, может назначаться только первая метка). После пересылки по MPLS-сети, PE-маршрутизатор удаляет метки и пересылает пакет CE-получателю, анализирующему только его IP-заголовок.

Следует обратить внимание на то, что в силу двухуровневой модели стека меток, маршрутизаторы магистральной сети могут даже иметь маршруты не до CE, а только до PE-маршрутизаторов.

8.7. Передача маршрутной информации между CE и PE

PE-маршрутизатор, подсоединенный к некой VPN, должен знать, как адреса этой VPN распределены по ее сайтам. Если CE-устройство является рабочей станцией (*host*) или коммутатором (*switch*), этот набор адресов будет конфигурироваться на обслуживающем его PE-маршрутизаторе. Если же CE-устройство — маршрутизатор, то существует несколько возможных способов получения PE-маршрутизатором названного набора адресов.

PE преобразует эти адреса в адреса VPN-IPv4, используя различитель маршрутов *RD*. Далее PE использует адреса VPN-IPv4 в качестве информации на входе BGP.

Возможность применения той или иной техники обмена маршрутной информацией между PE и CE зависит от того, принадлежит ли CE так называемой «транзитной VPN» или нет. *Транзитная VPN* — это VPN, содержащая маршрутизатор, который принимает маршрутную информацию от маршрутизаторов, не принадлежащих данной VPN, и не являющихся PE-маршрутизаторами, и передает эту информацию PE-маршрутизатору. В противном случае VPN является *тупиковой (stub VPN)*. Большая часть VPN, включая практически все корпоративные сети предприятий, попадает в разряд тупиковых.

Рассмотрим способы обмена маршрутной информацией. Статическая маршрутизация, т.е. маршрутизация, назначаемая администратором сети, может использоваться только в тупиковых VPN.

Маршрутизаторы PE и CE могут быть одноранговыми узлами протокола RIP. Тогда CE сможет передавать PE информацию о доступных адресных префиксах. Точно так же, PE- и CE-маршрутизаторы могут быть одноранговыми узлами протокола OSPF. В таком случае необходимо, чтобы сайт являлся отдельной зоной OSPF, CE был ABR в этой зоне, а PE — ABR в другой зоне. Этот способ может использоваться только в тупиковых VPN.

PE- и CE-маршрутизаторы могут быть одноранговыми узлами протокола BGP, и CE может использовать BGP (в частности, EBGP),

чтобы извещать PE о доступных адресных префиксах. Эта техника пригодна как для транзитных, так и для тупиковых VPN, и, с чисто технической точки зрения, является наилучшим решением, т.к., в отличие от IGP техники, не требует наличия в PE модулей поддержки нескольких протоколов маршрутизации для общения с разными CE. Это и понятно, т.к. BGP изначально создан для того, чтобы обеспечивать обмен маршрутной информацией между независимо управляемыми системами. Применение BGP позволяет также передавать атрибуты маршрутов от CE к PE.

С другой стороны, работа с протоколом BGP может не поддерживаться администратором CE, конечно, за исключением случаев, когда пользователь услугами VPN сам является Интернет-провайдером.

Если сайт находится не в транзитной VPN, он может не иметь уникального *номера автономной системы ASN*, и этот номер может быть выбран из частного плана нумерации ASN. Петли маршрутов при этом предотвращаются с помощью атрибута *Site of Origin*, описанного выше. Если группа сайтов образует транзитную VPN, то ее удобно представить как *BGP-конфедерацию* для того, чтобы внутренняя структура VPN была скрыта от любого маршрутизатора, не входящего в состав VPN. В этом случае каждый сайт в VPN должен иметь два BGP-соединения с магистральной сетью, одно из которых является относительно BGP-конфедерации внутренним, а второе — внешним. В стандартные внутриконфедерационные процедуры следует внести некоторые изменения для того, чтобы учесть возможность существования в магистральной сети и в сайтах различных вариантов сетевой политики.

Прежде чем распространять данные о маршрутах VPN-IPv4, полученные от сайта, PE должен присвоить этим маршрутам определенные значения атрибутов *Site of Origin, VPN of Origin и Target VPN,* рассмотренных в предыдущих параграфах главы.

Теперь обсудим распространение маршрутной информации от PE к CE. Разумеется, CE-устройство в этом случае должно быть маршрутизатором. В общем случае, PE может извещать CE о любом маршруте, данные о котором есть в его таблице маршрутизации, ассоциированной с тем сайтом, где содержится этот CE. При этом существует одно ограничение: о маршруте, имеющем атрибут *Site of Origin*, нельзя извещать CE-маршрутизаторы, которые принадлежат сайту, идентифицируемому этим атрибутом. Для передачи маршрутной информации от PE к CE используются те же процедуры, что и для передачи ее от CE к PE.

В большинстве случаев будет, однако, удобно в направлении от PE к CE (а в некоторых случаях — также и от CE к PE) назначать маршрут по умолчанию.

8.8. Поддержка MPLS маршрутизатором CE

Если CE поддерживает MPLS и ему требуется импортировать все маршруты, относящиеся к его VPN, PE может сообщить такому CE метки для каждого из этих маршрутов. Принимая затем от CE пакеты с подобными метками, PE будет заменять их соответствующими метками, полученными по протоколу BGP, и добавлять на верх стека метку для следующей пересылки BGP (*BGP Next Hop*).

8.8.1. Виртуальные сайты

Если распространение маршрутной информации между CE и PE производится по протоколу BGP, то используя MPLS, можно организовать несколько *виртуальных сайтов*. При этом CE может сам создавать для каждого из виртуальных сайтов отдельные таблицы маршрутизации, и тогда, если CE получит от PE данные обо всем наборе маршрутов, последнему не придется анализировать адреса ни в одном из полученных от CE пакетов.

Можно сделать и так, чтобы для каждого из виртуальных сайтов PE объявил по одному снабженному метками маршруту по умолчанию. Тогда, получив от CE пакет, снабженный меткой, PE будет знать, от какого из виртуальных сайтов пришел пакет, и к какой таблице маршрутизации нужно обратиться для его дальнейшей пересылки.

8.8.2. VPN Интернет-провайдера

Если VPN принадлежит Интернет-провайдеру, но его CE-маршрутизаторы поддерживают MPLS, то к такой VPN можно относиться как к тупиковой (stub), а не как к транзитной. Так как CE и PE должны обмениваться данными о маршрутах, относящихся только к определенной VPN, маршрутизатор PE может сообщить CE метки для каждого из таких маршрутов. Тогда маршрутизаторы, находящиеся в разных сайтах VPN, могут стать одноранговыми узлами BGP.

8.9. Стандартизация технологии MPLS-VPN

Описанный выше подход к построению VPN в значительной степени основан на документе RFC 2547, имеющем статус информационного. Наряду с ним существует еще один документ в том же статусе — RFC 2917 «A Core MPLS IP VPN Architecture»,— который предлагает архитектуру построения VPN, основанную на виртуальных маршрутизаторах. При этом предлагается не вводить расширения в протокол BGP и использовать эмуляцию локальной сети. Хотя оба документа имеют одинаковый статус, RFC 2917 более далек от стандартизации в силу того, что он меньше проработан.

8.10. Сценарии организации VPN на основе туннелей MPLS

В заключительном параграфе главы, по уже сложившейся в книге традиции, рассмотрим сценарии организации конкретной MPLS-VPN. На рис. 8.5 представлен фрагмент такой сети, объединяющей через сеть MPLS несколько удаленных пользователей и сайтов клиентов. Объединенные сайты и удаленные пользователи одной компании образуют виртуальную частную сеть VPN предприятия. На рис. 8.5 представлены две VPN: предприятия A, включающая в себя два сайта и удаленных пользователей, и предприятия B, включающая в себя три сайта территориально распределенных филиалов.

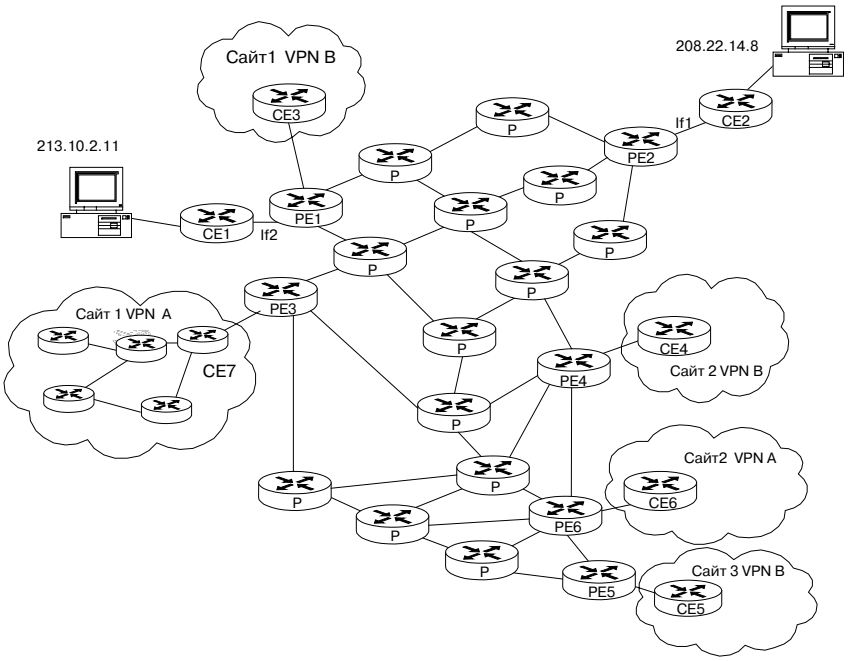


Рис. 8.5. Виртуальная частная сеть на базе технологии MPLS

В MPLS-VPN существуют два основных потока трафика: *поток управления*, который распространяет информацию о маршрутах VPN и другую информацию, необходимую для коммутации меток LSP, и *поток данных*, т.е. поток информации пользователей.

В свою очередь, поток управления состоит из двух потоков. Первый — обмен маршрутной информацией между PE и CE на границах магистрали провайдера и между маршрутизаторами PE через эту магистраль. Второй — информация, необходимая для создания LSP между маршрутизаторами PE.

Как отмечалось ранее в этой главе, механизмом для обмена маршрутной информацией между сайтами одной VPN является многопротокольное расширение протокола MP-BGP (MultiProtocol Border Gateway Protocol). С помощью MP-BGP пограничные маршрутизаторы PE организуют взаимные сеансы и в рамках этих сеансов обмениваются маршрутной информацией своих *таблиц маршрутизации VRF*. Отмеченной в главе 7 особенностью протокола BGP и его расширений является то, что он обменивается маршрутной информацией не со всеми непосредственно связанными с ним маршрутизаторами, как протоколы IGP, а лишь с теми из них, которые указаны в конфигурационных параметрах в качестве соседей.

Т.к. независимость адресных пространств VPN обеспечивается за счет применения разделителя маршрутов RD, PE-маршрутизатор должен уметь соотносить маршруты, ведущие к определенным маршрутизаторам CE, с определенным разделителем RD. Применение RD позволяет решить задачу определения разных маршрутов к устройствам, имеющим один и тот же IP-адрес, но принадлежащих разным VPN. Вопрос о том, кому отправлять маршрутные объявления, а кому нет, зависит от топологии виртуальной сети. Таким образом, кроме тех данных о маршрутах, которые поступают от сайтов, подсоединенных к PE, каждая таблица VRF дополняется данными о маршрутах, получаемыми от других сайтов той же VPN по протоколу MP-BGP.

Перед тем как распространять маршрутные объявления, полученные от сайта, PE должен присвоить маршрутам атрибуты Site of Origin, VPN of Origin и Target VPN. В принципе, маршрутизатор PE может объявлять любой маршрут, зафиксированный в его таблице маршрутизации, которая ассоциирована с сайтом, содержащим рассматриваемый CE, с учетом одного ограничения: маршрут, имеющий атрибут Site of Origin, никогда не должен объявляться CE-маршрутизатором, принадлежащим сайту, который идентифицируется этим атрибутом.

Теперь перейдем к *созданию LSP*. Для передачи трафика по сети MPLS следует создать LSP между маршрутизаторами PE. Создание LSP означает создание таблиц коммутации по меткам во всех маршрутизаторах PE и P, образующих этот LSP. Для каждого подключенного сайта PE-маршрутизатор поддерживает одну таблицу маршрутизации. Эти таблицы используются PE-маршрутизаторами только при получении пакетов от присоединенных к ним сайтов. После того как маршрутизатор CE назначит маршрут, связанный с ним маршрутизатор магистральной сети провайдера записывает локальный маршрут в своей базе LIB, а затем выбирает метку для объявления ее вместе с маршрутом. В одной системе может существовать несколько маршрутов, которые зависят от сайта, передающего информационный трафик.

LSP через сеть провайдера могут быть созданы двумя способами: либо по технологии ускоренной маршрутизации с помощью протоколов LDP, либо на основе рассматриваемой в следующей главе технологии инжиниринга трафика с помощью протоколов RSVP-TE или CR-LDP. Если LSP создается при помощи протокола LDP, то обслуживаться он будет согласно модели *best effort*. Если же провайдер хочет выделить для LSP определенную полосу пропускания или использовать средства регулирования трафика для организации явного маршрута, то используется протокол RSVP, позволяющий поддерживать заданное качество обслуживания и регулирование трафика.

Как правило, для совместимости оборудования разных производителей от всех маршрутизаторов PE требуется поддержка протокола LDP. При использовании провайдером протокола резервирования ресурсов, образованные на основе этого протокола LSP будут иметь больший приоритет, чем LSP, созданные с помощью протокола LDP. Между парой маршрутизаторов могут присутствовать как те, так и другие.

Передачу *трафика пользователя* от одного сайта виртуальной частной сети на другой ее сайт можно разделить на четыре этапа:

- передача потока от CE-маршрутизатора источника до входного пограничного маршрутизатора сети провайдера,
- передача информационного потока от PE до P,
- передача трафика по транзитной области MPLS,
- передача потока от выходного маршрутизатора PE до маршрутизатора назначения CE.

Первый этап. Пакет данных, предназначенный для другого сайта, поступает на пограничный маршрутизатор клиента CE. Маршрутизатор CE принимает исходящий пакет данных своего сайта, определяет маршрут его следования и передает пакет к связанному с ним PE-маршрутизатору.

Второй этап. При получении пакета от маршрутизатора CE, маршрутизатор PE производит поиск маршрута в таблице VRF. Если пакет предназначается узлу CE, подсоединенному непосредственно к рассматриваемому PE, он немедленно отсылается к этому CE. В противном случае маршрутизатор PE обращается к своим таблицам VRF и LIB, формирует стек меток и отправляет пакет к следующему маршрутизатору. После передачи информационного трафика по сети MPLS, PE-получатель удаляет метки и пересылает пакет CE-маршрутизатору, который определяет дальнейший маршрут для пакета на основе анализа его IP-заголовка.

Третий этап. Маршрутизаторы магистральной сети не имеют сведений о маршрутах к сайтам VPN. Пересылка пакетов ведется

по технологии MPLS: внутренние маршрутизаторы P магистральной сети провайдера коммутируют пакет с меткой, заменяя верхнюю метку стека на каждом участке, пока пакет не достигнет предпоследнего маршрутизатора PE, находящегося непосредственно перед тем маршрутизатором, к которому направлен пакет. Этот маршрутизатор извлекает верхнюю метку из стека и отправляет пакет к пограничному PE-маршрутизатору назначения магистральной сети провайдера.

Четвертый этап. Маршрутизатор PE, присоединенный к определенной виртуальной сети, должен уметь определять, как распределены адреса этой сети по ее сайтам. При получении пакета маршрутизатор PE на основе анализа нижней метки стека определяет CE-маршрутизатор назначения и отправляет к нему пакет.

Теперь рассмотрим для представленной на рис. 8.5 структуры VPN предшествующие трафику пользователя *управляющие потоки обмена маршрутными объявлениями* между маршрутизаторами PE и CE и между маршрутизаторами PE. Пример сценария этого обмена изображен на рис. 8.6.

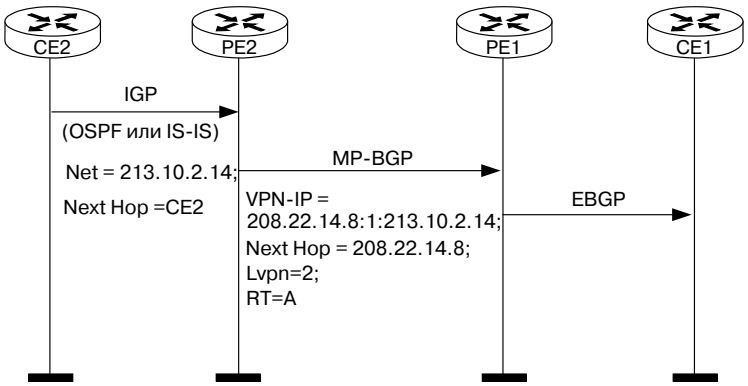


Рис. 8.6. Обмен маршрутной информацией

Маршрутизатор CE2 при помощи протокола внутренней маршрутизации OSPF или IS-IS объявляет префиксы маршрута связанному с ним пограничному маршрутизатору PE2 сети провайдера. Это объявление содержит IP-адрес удаленного пользователя виртуальной частной сети VPN-A и адрес внешнего интерфейса маршрутизатора CE2 (if1): *Net = 213.10.2.11; Next Hop = CE2.*

Маршрутизатор PE2 проверяет соответствие этого маршрута локально сконфигурированным правилам импорта, и если маршрут удовлетворяет этим правилам, вносит данные о нем в соответствующую таблицу VRF.

Как уже отмечалось, маршрутное объявление передается по протоколу MP-BGP только тем маршрутизаторам PE, которые записаны

в конфигурационных параметрах маршрутизатора-отправителя в качестве соседей. Перед тем как направить маршрутное объявление выходному маршрутизатору PE1, маршрутизатор PE2 преобразует его в формат VPN-IP, для чего добавляет к адресу различитель маршрутов RD, изменяет поле Next Hop, записывая в него адрес своего внешнего интерфейса, назначает метку VPN, указывающую виртуальную частную сеть VPN-A и интерфейс маршрутизатора PE2, к которому подключен сайт, и, наконец, назначает атрибуты RT, Site_of_Origin, VPN_of_Origin. В результате маршрутное объявление по протоколу MP-BGP будет иметь следующий вид: *VPN-IP = 208.22.14.8:1:213.10.2.11; Next Hop = 208.22.14.8; Lvpn=2; RT=A.*

Когда выходной маршрутизатор PE1 получает данные о маршруте VPN-IPv4, он проверяет содержащийся в них атрибут RT на предмет совпадения с правилами импорта, применяемыми во всех его (в данном примере VRF1A и VRF1B). Значение атрибута route-target равно A, поэтому данные о маршруте добавляются (после удаления RD) только в таблицу VRF2A, так как для нее определены правила импорта A. Таблица VRF2B остается в неизменном виде, так как правила импорта B говорят о том, что в нее должны помещаться только данные о маршрутах с атрибутом route-target B. В нашем примере маршрутизаторы CE1 и PE1 находятся в разных автономных зонах AS, а потому обмен маршрутными объявлениями между этими маршрутизаторами ведется при помощи протокола EBGP.

Процесс управления созданием LSP-туннелей для представленной на рис. 8.5 структуры виртуальной частной сети выглядит следующим образом. Перед созданием LSP-туннеля в сети MPLS работает протокол LDP, благодаря которому создается топологическая карта сети, распространяются атрибуты ресурсов и создаются обычные тракты LSP (рис. 8.7).

Администратор сети провайдера собирает статистику по этим трактам, после чего, в соответствии с договоренностью о предоставляемом QoS, принимает решение о создании LSP-туннелей. Чтобы создать такой туннель, администратор сети, используя средства рассматриваемого в следующей главе инжиниринга трафика, на основе статистических данных о сети и об LSP определяет маршрут, который удовлетворит требованиям к QoS, и формирует его, вводя в сообщение Path объект ERO, в котором записывает явный маршрут. В нашем случае ERO содержит запись P3, P2, P1, PE1.

Создание туннелей уже рассматривалось выше и будет еще обсуждаться в следующей главе, а здесь мы приведем лишь сценарий (рис. 8.8) все для того же примера. Сообщение Path несет в себе объекты Label Request, Session, Sender Template, Time Values, Sender Tspec.

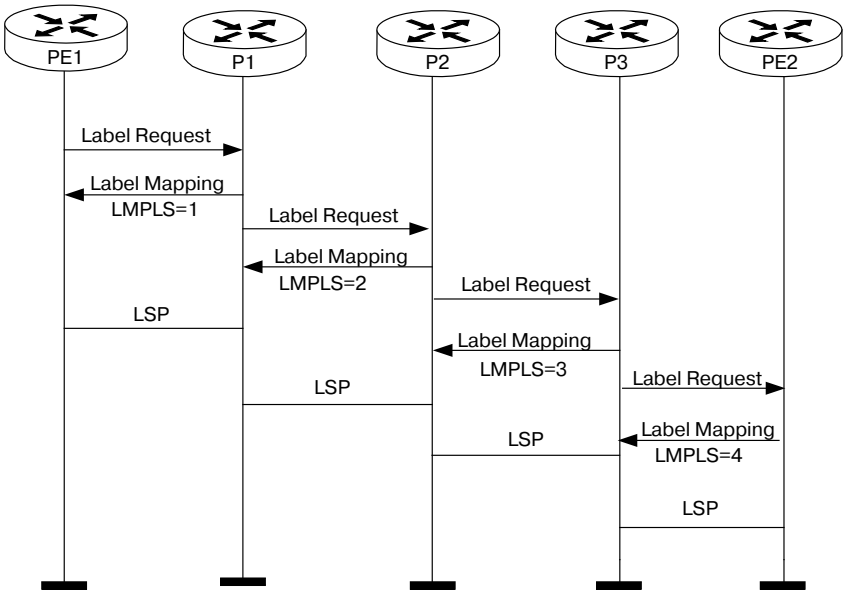


Рис. 8.7. Создание коммутируемого по меткам тракта

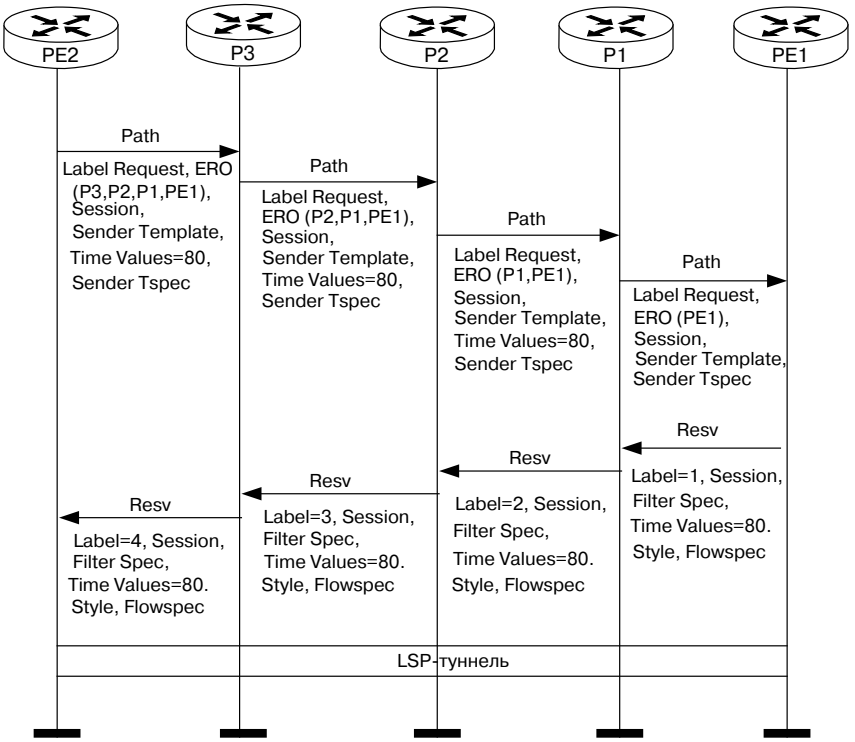


Рис. 8.8. Создание LSP-туннеля

Каждый маршрутизатор, получивший сообщение Path, пересылает его следующему маршрутизатору в соответствии с записью в объекте ERO и фиксирует у себя адрес маршрутизатора, от которого это сообщение было получено. Пройдя таким образом через маршрутизаторы P3, P2, P1, сообщение Path поступает к маршрутизатору PE1. Этот маршрутизатор, приняв сообщение, определяет наличие требующихся ресурсов и правомерность запроса. Если запрос правомерен, и нужные ресурсы имеются, PE1 резервирует их для создаваемого туннеля, после чего передает сообщение резервирования ресурсов Resv маршрутизатору P1. Сообщение Resv, содержащее объекты Label, Session, Filter Spec, Time Values, Style, Flowspec, следует в направлении, обратном направлению следования сообщения Path, т.е. проходит через маршрутизаторы P1, P2, P3 и PE2, каждый из которых производит резервирование ресурсов. Таким образом, организуется LSP-туннель, причем во всех маршрутизаторах, через которые он проходит, резервированы определенные ресурсы для обеспечения требуемого качества обслуживания.

И, наконец, последний сценарий (рис. 8.9) иллюстрирует, все для того же приведенного на рис. 8.5 примера VPN, *продвижение трафика данных*. Здесь представлен процесс передачи через магистральную сеть провайдера MPLS-TE трафика данных от удаленного пользователя в точку доступа, которая находится на сайте, принадлежащем VPN A.

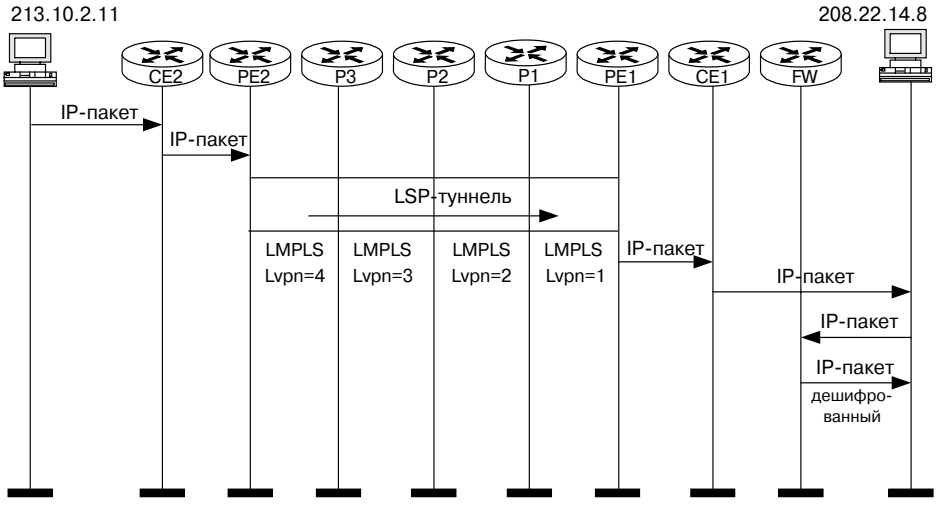


Рис. 8.9. Передача информационного трафика

Удаленный пользователь 213.10.2.11 отправляет IP-пакет. В первую очередь, этот пакет попадает на сервер доступа AS, реализованный вместе с маршрутизатором CE2. Сервер доступа проводит аутентификацию и авторизацию пользователя, после чего по таб-

лице маршрутизации CE2 определяет дальнейший путь следования пакета и пересылает его к маршрутизатору PE2. При получении IP-пакета PE2 обращается к таблице VRF-A и назначает этому пакету метку виртуальной частной сети Lvpn. Далее маршрутизатор обращается к таблице LIB, в которой, в соответствии с требуемым QoS выбирается LSP-туннель. Продвижение пакета по сети провайдера происходит на основании метки верхнего уровня, которой является Lvpn. Каждый раз, когда пакет проходит очередной маршрутизатор P вдоль туннеля, метка Lvpn анализируется и заменяется новой. И только после достижения конечной точки LSP-туннеля (маршрутизатора PE1) метка Lvpn из стека извлекается.

Дальнейшее продвижение пакета основано на метке LMPLS. В зависимости от ее значения пакет направляется в тот или иной выходной интерфейс маршрутизатора PE1: маршрутизатор PE1 обращается к таблице VRF VPN-A, связанной с этим интерфейсом, и извлекает запись о маршруте к пункту назначения. Продвижение пакета к маршрутизатору CE1 обеспечивается при помощи протокола IP. Когда стандартный IP-пакет поступает в маршрутизатор CE1, по таблице маршрутизации производится поиск маршрута, и пакет поступает на FW. Брандмауэр проводит фильтрацию трафика, поступающего от провайдера. После фильтрации пакет дешифрируется и поступает к получателю *208.22.14.8*.

В заключение главы следует отметить, что такое сочетание технологий для организации VPN несомненно является эффективным. В главе рассматривались примеры корпоративной сети, но точно так же можно построить сеть оператора IP-телефонии, имеющую сайты в разных странах, поддерживающую множество взятых из VPN дополнительных услуг и реализующую гарантированное качество обслуживания на основе MPLS. Такая MPLS VPN представляется одной из составляющих перспективных сетей NGN.

Глава 9
Инжиниринг трафика

9.1. Концепция инжиниринга трафика в MPLS

Инжиниринг трафика представляет собой мониторинг и моделирование потоков трафика, а также управление трафиком с тем, чтобы обеспечить нужное качество его обслуживания путем рационального использования сетевых ресурсов за счет сбалансированной их загрузки. В отечественной литературе совокупность механизмов, обеспечивающих выполнение перечисленных функций, называется также *перераспределением* трафика, *конструированием* трафика, *оптимизацией* трафика, *проектированием* трафика, *управлением* трафиком. Более точным является название *управление разнотипным трафиком*, т.к. оно подчеркивает связь рассматриваемых здесь механизмов с задачей обеспечить разное качество обслуживания (QoS) трафика разных типов, но в книге используется наиболее распространенная прямая калька с английского эквивалента — *Traffic Engineering (TE)*. Впрочем, о распространенности в данном случае можно говорить весьма условно: терминология в этой молодой области пока еще находится в стадии становления.

Так или иначе, возможности инжиниринга трафика — главная или, по крайней мере, одна из главных причин, по которым технология MPLS реализуется в сегодняшних сетях связи. Они позволяют снять ограничения, присущие рассмотренным в предыдущих главах протоколам маршрутизации внутреннего шлюза IGP, как дистанционно-векторным (RIP), так и на основе состояния каналов (OSPF и IS-IS). Эти протоколы направляют трафик от отправителя к адресату по кратчайшему маршруту, заранее выбранному в определенной метрике. Для протокола RIP эта метрика представляет собой просто число пересылок (hops) между маршрутизаторами. Протоколы маршрутизации на основе параметров состояния ка-

налов — OSPF и IS-IS — используют более «продвинутые» метрики: предельную пропускную способность каналов, вносимую каналом задержку при передаче пакета и т.п. Однако и они игнорируют текущую ситуацию с перегрузками в сети и тип трафика, который несут маршрутизируемые пакеты. Это же справедливо и в отношении протоколов внешнего шлюза EGP, в частности, рассмотренного в главе 7 протокола BGP-4. Напомним, что протокол BGP-4 используется также и внутри автономных систем, но и в этом качестве он, как и другие рассмотренные выше традиционные протоколы маршрутизации IGP и EGP, непригоден для инжиниринга трафика в силу зависимости используемых им метрик от топологии сети, а не от реальной обстановки, связанной с прохождением и обработкой трафика. При таком подходе на выбранном кратчайшем маршруте может возникнуть перегрузка, что приведет к задержке или потере пакетов, несмотря на наличие в других альтернативных LSP неиспользуемой пропускной способности. К примеру, протокол OSPF может даже увеличить перегрузку, поскольку он стремится направить трафик по перегруженному кратчайшему маршруту и тогда, когда другой, пусть не самый короткий, но вполне приемлемый маршрут загружен меньше.

Классический пример, иллюстрирующий эту проблему маршрутизации по принципу SPF (ortest Path First), демонстрирует элемент сетевой топологии, имеющий из-за внешнего сходства фольклорное наименование «рыба». На рис. 9.1 изображен в таком виде все тот же фрагмент MPLS-сети из 7 маршрутизаторов LSR1 — LSR7, представленный ранее на рис. 1.3 и на аналогичных рисунках в других главах книги. Весь трафик от маршрутизатора LSR2 к LSR5, согласно алгоритму OSPF идет через маршрутизатор LSR6. Таким образом, маршрут LSR2-LSR6-LSR5 перегружен, а ресурс маршрута LSR2-LSR3-LSR4-LSR5 используется неэффективно.

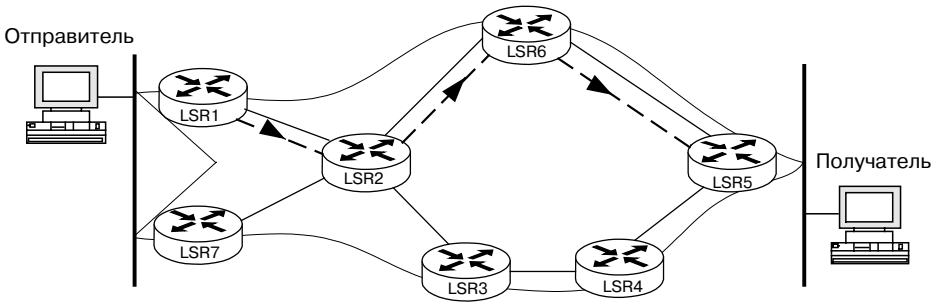


Рис. 9.1. Традиционное распределение трафика

Применение механизмов TE позволяет решить эту проблему, указав два разных пути от LSR1 к LSR5, как это изображено на рис. 9.2.

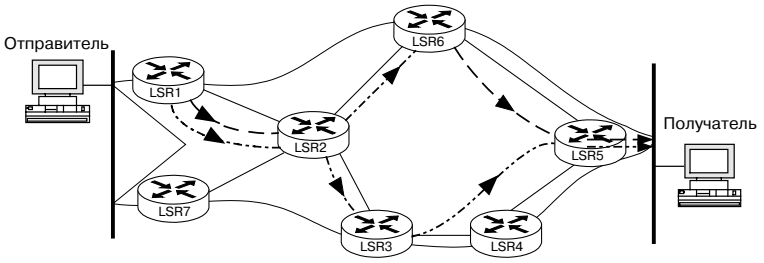


Рис. 9.2. Равномерное распределение трафика TE

Сравнение рис. 9.1 и 9.2 показывает, что TE позволяет лучше использовать сетевые ресурсы за счет перевода части трафика с более загруженного на менее загруженный участок сети. При этом не только лучше используется доступная полоса пропускания, но и достигается более высокое качество обслуживания трафика, поскольку уменьшается вероятность перегрузки в сети. Кроме того, для услуг, которые требуют выполнения заданных норм качества обслуживания QoS, например, заданного коэффициента потерь пакетов и/или задержки/джиттера, инжиниринг трафика позволяет обеспечивать надлежащее QoS путем назначения явно определенных маршрутов. В качестве эксплуатационного инструмента, инжиниринг трафика регулярно оптимизирует использование сетевых ресурсов при изменениях распределения нагрузки в сети.

Таким образом, администратор сети может управлять потоками трафика, проходящими через сеть, и в принудительном порядке направлять по заранее выбранному маршруту пакеты, которые поступают к входному маршрутизатору LSR1 тракта LSP и соответствуют классу эквивалентности пересылки FEC этого LSP, к выходному маршрутизатору LSR7. Кроме того, FEC может определяться с учетом большего числа параметров классификации, чем просто IP-адрес получателя, и использоваться для маршрутизации в соответствии с установленными правилами и распределением нагрузки.

Ключевые характеристики, сопряженные с управлением трафиком, могут быть ориентированы либо на трафик, либо на ресурсы. Задачи TE включают в себя аспекты улучшения показателей QoS информационных потоков, а центральной функцией TE является оптимальное управление пропускной способностью.

Оптимизация рабочих характеристик сети является фундаментальной проблемой управления. В модели процесса TE *трафик-инженер (Traffic Engineer)* действует как контроллер в системе с адаптивной обратной связью. Эта система включает в себя набор взаимосвязанных сетевых элементов, систему мониторинга состояния сети и набор средств управления конфигурацией. Трафик-инженер формулирует политику управления, контролирует состояние сети,

используя систему мониторинга, определяет характеристики трафика и предпринимает управляющие действия, чтобы перевести сеть в состояние, согласующееся с политикой управления. Это может быть выполнено с помощью операций, производимых в порядке отклика на текущее состояние сети, или превентивно, на основе анализа тенденции изменений состояния сети и их прогнозирования с тем, чтобы предотвратить возникновение нежелательных состояний.

В идеале управляющие действия должны включать в себя:

- модификацию параметров управления трафиком,
- модификацию параметров, связанных с маршрутизацией,
- модификацию атрибутов и констант, связанных с ресурсами.

Трафик-инженер здесь означает не должность сотрудника (хотя возможно и такое), а некоторый абстрактный управляющий объект. Участие человека в процессе управления трафиком должно быть минимизировано настолько, насколько это возможно, что обеспечивается автоматизацией вышеперечисленных операций.

Заметим, что инжиниринг трафика — отнюдь не связанное с MPLS изобретение. Задачи управления трафиком решались в отрасли связи всегда, правда, подходы были принципиально различны. Расчеты, производившиеся при проектировании АТС на основе теории телетрафика, как и измерение характеристик нагрузки и качества обслуживания в процессе эксплуатации с последующей реконфигурацией пучков соединительных линий, тоже можно отнести к управлению трафиком. С появлением пакетных сетей и мультимедийного трафика задачи и методы управления трафиком претерпели значительные изменения.

В этих новых условиях применение теории телетрафика можно отнести к проектированию и к эксплуатационному управлению сетями связи. Время, затрачиваемое на решение этих задач (время *реакции*), может измеряться в неделях, днях и часах, соответственно, как это изображено на рис. 9.3.

Для рассматриваемых в этой главе методов инжиниринга трафика MPLS время реакции измеряется в секундах, минутах и часах. В сетевом оборудовании могут применяться еще более быстродействующие методы борьбы с перегрузками, время реакции которых измеряется в секундах и даже в миллисекундах. Перегрузка, связанная с неэффективным размещением ресурсов, может быть уменьшена выбором должной *политики балансировки нагрузки* в разных устройствах сети. Задачей такой политики является также минимизация перегрузки ресурса. Когда перегрузка минимизирована, потери пакетов и задержка доставки уменьшаются, а совокупная пропускная способность возрастает.

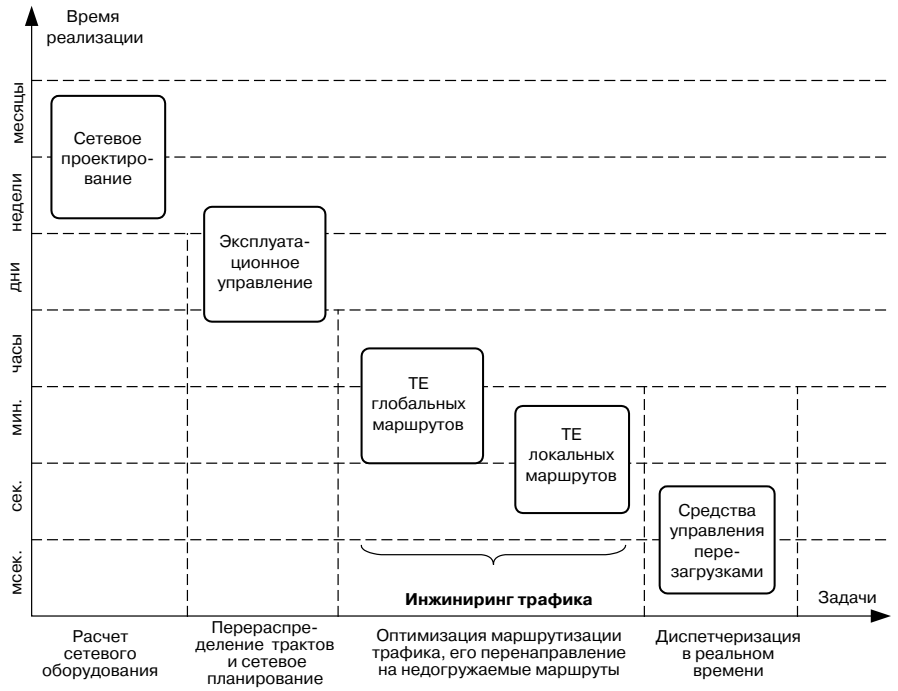


Рис. 9.3. Инжиниринг трафика в сетях связи

Следует сказать о некоторой условности рис. 9.3, главной задачей которого является иллюстрация подходов к анализу трафика в сетях связи и места рассматриваемого здесь инжиниринга трафика MPLS. В качестве компенсации этой условности приведем в заключение параграфа более строгое определение TE согласно RFC 2702: *инжиниринг трафика включает в себя технологию и научные принципы измерения, моделирования и описания трафика, управление трафиком и использование этих знаний и техники для получения определенных рабочих характеристик сети*.

9.2. Протоколы сигнализации

Хотя методы TE появились раньше технологии MPLS, именно эта технология очень хорошо подходит для управления трафиком и может предоставить большую часть функций TE по относительно низкой (в сравнении с конкурирующими решениями) цене. Не менее важно, что в MPLS можно автоматизировать функции управления трафиком. Для всего этого нужно только одно — протоколы сигнализации MPLS должны уметь переносить информацию, которая необходима для работы механизмов управления трафиком, находящихся на прикладном уровне, а также «прокладывать» тракты

LSP по явно заданным маршрутам. При этом в MPLS есть возможность достичь дополнительной гибкости, т.к. маршрут можно задать не только строго, но и не строго, т.е. группа узлов может быть задана как «абстрактный узел», внутри которого существует известная свобода выбора маршрута.

Перед IETF, точнее перед ее рабочей группой MPLS, встала задача выбрать такой протокол. Лучшими оказались два варианта, RSVP-TE и CR-LDP, уже упоминавшиеся в главах 3 и 4. В первом варианте протоколу RSVP необходимо делать в сети MPLS то, что он уже выполнял в сетях IP, а именно — обрабатывать информацию, связанную с QoS, и резервировать ресурсы. Остается лишь добавить возможность распределения меток. Во втором варианте тоже не представлялось сложным добавить к уже используемому для распределения меток в MPLS протоколу LDP несколько новых объектов для переноса информации о QoS. В итоге, рабочая группа так и не смогла придти к единому решению по этому вопросу, и оба протокола были развиты до уровня *proposed standard*.

Как это часто случается в последнее время, практически сразу же вслед за рабочей группой IETF задача выбора протокола сигнализации для инжиниринга трафика в MPLS встала также перед производителями сетевого оборудования и программного обеспечения, а затем — и перед Операторами. Для производителей это вылилось в необходимость создавать в своем оборудовании средства поддержки обоих протоколов и выпускать разные версии, поддерживающие либо тот, либо другой протокол. Операторам пришлось еще сложнее — перед ними возникла проблема выбора, какой из протоколов использовать в своей сети MPLS, а также извечная проблема взаимодействия между сетевыми областями, использующими разные протоколы, например, при слиянии сетей или при покупке сети альтернативного оператора.

Вначале протоколы имели ряд заметных функциональных и технических различий, базируясь на которых, можно было делать выбор в пользу того или иного конкурента. Со временем протоколы эволюционировали и развивались, сглаживая свои недостатки и сохраняя достоинства. Таким образом, выбор между ними все усложнялся. К сравнительному анализу этих двух протоколов мы вернемся в конце главы, после того как рассмотрим принципы и модели TE.

9.3. Атрибуты потоков трафика и сетевых ресурсов

Напомним, что тракт LSP — это логическое соединение, которое создается в MPLS-сети для переноса через нее пакетов, принадлежащих одному FEC. Элементами такого соединения являются

маршрутизаторы LSR и связывающие их звенья. При этом нельзя забывать, что цепочки < LSR_i — звено — LSR_{i+1} — звено — ..... LSR_{n-1} — звено — ISR_n>, по которым проходят разные пакеты одного и того же FEC (т.е. маршруты, которые используются в LSP), в общем случае, могут быть разными.

В области инжиниринга трафика применительно к MPLS используется термин *traffic trunk,* которым обозначают объединение потоков трафика, принадлежащих одному классу и проходящих по одному LSP. Иными словами, *traffic trunk* — это *объединенный поток трафика*, отнесенного к одному FEC.

При инжиниринге трафика в MPLS вводится понятие *наведенный MPLS-граф.* Этот граф образуют набор LSR, представляющих его *узлы*, набор звеньев, представляющих его *ребра*, и набор используемых в LSP маршрутов, представляющих *пути* наведенного графа.

Управление трафиком в MPLS предполагает наличие следующих функциональных средств и возможностей:

- набор атрибутов, которые связаны с объединенными потоками трафика;
- набор атрибутов, которые связаны с ресурсами; эти атрибуты ограничивают возможности выбора маршрутов для LSP и могут рассматриваться как топологические ограничения;
- маршрутизация на основе ограничений, которая используется для выбора маршрута в соответствии с заданными наборами параметров.

Атрибуты, связанные с потоками трафика и с ресурсами, а также параметры, связанные с маршрутизацией, в совокупности представляют собой набор управляющих переменных, которые могут быть модифицированы в результате действий администраторов или автоматических агентов (трафик-инженеров) управления сетью. В рабочей сети всегда желательно, чтобы эти атрибуты можно было менять динамически в реальном времени. Рассмотрим отдельно атрибуты объединенных потоков трафика и атрибуты сетевых ресурсов.

9.3.1. Атрибуты объединенных потоков трафика

Атрибут объединенного потока трафика описывает характеристики этого потока. Значения атрибутов могут быть явно присвоены потокам администратором или заданы неявно базовыми протоколами, когда пакеты классифицируются и сортируются по FEC при входе в домен MPLS. Приведем основные из этих атрибутов, специфицированные IETF:

Атрибуты параметров трафика могут использоваться при сборе данных о потоках трафика (или, более точно, о FEC), которые над-

лежит транспортировать через тракт LSP. Параметры трафика определяют требования к ресурсам этого LSP.

Атрибуты управления и выбора маршрутов определяют правила выбора маршрутов для LSP, а также правила работы с маршрутами, которые уже существуют. Они включают в себя:

- Административно специфицированные маршруты, которые конфигурируются оператором. Такой маршрут может быть определен полностью или частично и являться либо обязательным для использования, либо лишь предпочтительным.
- Иерархия предпочтения для набора маршрутов административно специфицирует иерархию в наборе возможных маршрутов для данного LSP.
- Атрибуты Resource Class Affinity используются для спецификации класса ресурсов, которые следует явно включить в LSP или исключить из него. Это атрибуты политики, которые могут использоваться с целью введения дополнительных ограничений на маршруты для LSP.
- Атрибут адаптивности представляет собой двоичную переменную, определяющую, как должен реагировать тракт на изменение состояния сети: разрешить или запретить адаптивную оптимизацию.
- Распределение нагрузки в параллельных звеньях (трактах). Когда объединенный трафик между узлами таков, что одно звено (один тракт) не сможет его пропустить, и единственным решением является деление трафика на несколько потоков, то эти атрибуты указывают доли трафика, проходящего через каждое (каждый) из параллельных звеньев (трактов).

Атрибут приоритета определяет относительную важность объединенного потока трафика. Приоритеты используются в случае отказов для того, чтобы определить порядок, в котором выбираются из имеющегося списка маршруты для соответствующих LSP, а также в реализациях, допускающих приоритетное обслуживание.

Атрибут Preemption определяет, может ли поток трафика заместить другой поток в данном тракте, и задает условия приоритетного замещения:

- preemptor enabled — может замещать,
- non-preemptor — не может замещать,
- preemptable — допускает замещение,
- non-preemptable — не допускает замещение.

Поток трафика, который допускает замещение, может быть замещен другим потоком, имеющим более высокий приоритет и атрибут *Preemption* со значением preemptor enabled.

Атрибут устойчивости (Resilience) определяет поведение звена (тракта) в случае возникновения ошибок. *Базовый атрибут Resilience* указывает процедуру восстановления, которая должна быть запущена для данного звена (тракта) при возникновении отказа. *Расширенный атрибут Resilience* может использоваться для подробной спецификации действий в случае отказа.

Атрибут Policing определяет действия, которые следует предпринять, когда звено/тракт становится неполноценным, т.е. когда какие-то его параметры выходят за допустимые пределы. Атрибуты *Policing* могут указывать, надо ли лимитировать звено/тракт по полосе пропускания, пометить его или просто пересылать его трафик без каких-либо действий.

9.3.2. Атрибуты сетевых ресурсов

Атрибуты ресурсов сети входят в параметры топологии и служат для того, чтобы определить для маршрутизации потоков трафика ограничения, учитывающие характеристики заданных ресурсов.

Maximum Allocation Multiplier (MAM) является административно задаваемым атрибутом, который определяет долю ресурса, доступную звену (тракту). Этот атрибут используется, в основном, для распределения полосы пропускания. Однако он может быть применен также для резервирования ресурсов LSR.

Атрибуты Resource Class также присваиваются административно и вводят понятие *класс ресурса*. Атрибуты класса ресурса могут рассматриваться как приписанные ресурсам *цвета*, такие, что набор ресурсов с одним *цветом* принадлежит одному классу. Эти атрибуты могут использоваться для реализации разных вариантов политики.

9.4. Маршрутизация на основе ограничений

Маршрутизация на основе ограничений (constraint-based routing) использует в качестве входных данных рассмотренные выше атрибуты, связанные с потоками трафика, атрибуты, связанные с ресурсами, а также другую информацию, характеризующую текущую топологию сети, и дает возможность резервировать по запросу ресурсы для управления трафиком. При этом она может сосуществовать с имеющимися IGP-протоколами маршрутизации, работающими по принципу *hop-by-hop*.

Базируясь на названной информации, процесс маршрутизации на основе ограничений автоматически вычисляет маршрут для каждого потока, исходящего от определенного узла. Результат вычислений представляет собой спецификацию маршрута, который удовлетворяет требованиям, записанным в атрибутах потока

трафика. Ограничения формулируются на основе сведений о доступности ресурсов, административной политики и топологической информации.

Маршрутизация на основе ограничений предназначена для того, чтобы максимально сократить объем работ, связанных с ручной конфигурацией и степень участия администратора в реализации политики управления трафиком.

Известно, что для большинства реальных ситуаций проблема маршрутизации на основе ограничений трудно разрешима. Однако на практике для нахождения приемлемого маршрута может использоваться следующий простой метод. Сначала отбрасываются ресурсы, которые не удовлетворяют требованиям атрибутов потока трафика. Затем для оставшегося графа связей запускается алгоритм поиска кратчайшего пути. Оптимизация обычно сводится к минимизации перегрузки. Но в случае, когда нужно маршрутизировать пакеты нескольких потоков трафика, т.е. пакеты нескольких FEC, предложенный алгоритм не всегда позволяет найти решение, даже если таковое существует.

При реализации маршрутизации на основе ограничений в сетевых устройствах необходимо наличие:

- механизмов обмена информацией о топологическом статусе (данными о доступности ресурсов, информацией о состоянии связи, информацией об атрибутах ресурсов);
- механизмов работы с информацией о топологическом статусе;
- средств взаимодействия между процессами маршрутизации на основе ограничений и процессами традиционных IGP;
- механизмов, обеспечивающих адаптивность звеньев и трактов;
- механизмов, обеспечивающих устойчивость и живучесть трактов.

9.5. Механизмы TE в MPLS

Резюмируя приведенные в предыдущих параграфах базовые сведения о TE, можно сказать, что инжиниринг трафика в MPLS основан на управлении наборами атрибутов, значения которых учитываются при выборе маршрутов для создаваемых в MPLS-сети LSP и LSP-туннелей. Теперь попробуем описать общую картину работы приложения TE в MPLS, не углубляясь, однако, в подробности, рассмотрение которых могло бы составить отдельную книгу. Основными компонентами подсистемы TE являются:

- пользовательский интерфейс, через который администратор сети может управлять политикой TE;
- IGP-компонент, распространяющий информацию о топологии сети и сведения о состоянии сетевых ресурсов;

- маршрутизация на основе ограничений — модуль, который производит расчет маршрута в сети MPLS на основе информации, получаемой от пользовательского интерфейса и IGP-компонента;
- компонент сигнализации для создания и поддержки LSP (или LSP-туннеля), для управления LSP (LSP-туннелем) и для резервирования сетевых ресурсов;
- компонент пересылки данных, в качестве которого выступает сама сеть MPLS.

Напомним, что TE — это механизм оптимизации сети, и потому сетевые узлы, наряду с вычислением маршрутов на основе ограничений, должны уметь рассчитывать традиционные маршруты согласно алгоритму SPF. Функциональная модель такого LSR-TE сети MPLS представлена на рис. 9.4.

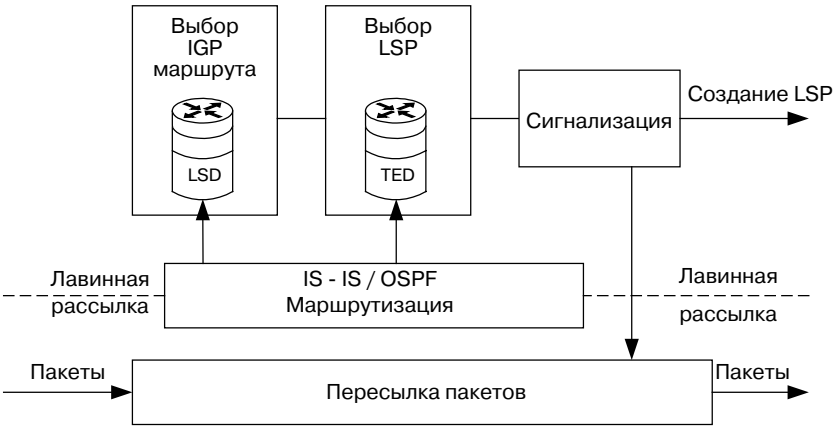


Рис. 9.4. Модель LSR с функциями TE

LSR с функциями TE имеет две отдельные базы данных: одну — традиционную *Link-State Database (LSD)*, а другую — *Traffic Engineering Database (TED)*, где хранятся атрибуты звеньев и топологическая информация. При использовании в сети MPLS механизмов TE вначале необходимо получить статистические данные о трафике. Удобнее собирать статистические данные не по звеньям, а по LSP, так как в этом случае виден объем трафика между LSR для каждой пары узлов сети. Для пояснения приведем пример, представленный на рис. 9.5 и 9.6.

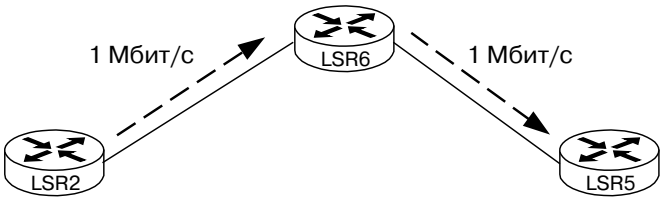


Рис. 9.5. Трафик звеньев

Если собирать статистические данные по звеньям (рис. 9.5), то непонятно, какой объем трафика адресован от маршрутизатора LSR2 маршрутизатору LSR6, а какой — маршрутизатору LSR5. Собирая же статистические данные по LSP, как это показано на рис. 9.6, можно увидеть, например, что по LSP от LSR2 к LSR6 передается 300Кбит/с, по LSP от LSR2 к LSR5 через LSR6 — 700 Кбит/с и по LSP от LSR6 к LSR5 — 300 Кбит/с.

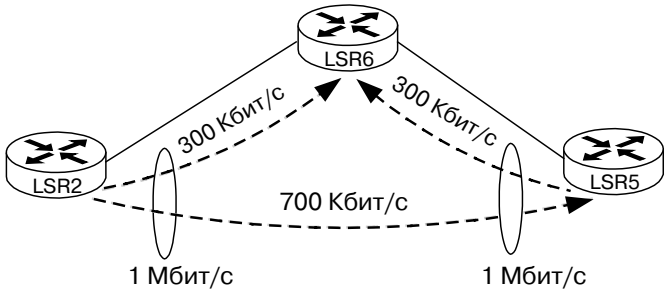


Рис. 9.6. Трафик LSP

Возможна ситуация, когда получать статистические данные не требуется, например, когда администратор заранее знает, какая сквозная пропускная способность ему потребуется между двумя узлами сети. Так или иначе, теперь администратор может сформировать требования к пропускной способности по направлениям, причем это может быть выполнено как вручную, так и с помощью средств автоматизации. Эти требования администратор сообщает системе через пользовательский интерфейс.

В процессе эксплуатации сети MPLS расширенные маршрутные IGP-протоколы, такие как OSPF-TE и IS-IS-TE, распространяют в сети следующую информацию о топологии сети и состоянии ресурсов:

- максимальная пропускная способность звена,
- максимальная пропускная способность звена, доступная для резервирования,
- резервированная на звене пропускная способность,
- текущее использование пропускной способности,
- «цвет» ресурса.

Поскольку состояние сетевых ресурсов изменяется намного чаще, чем топология сети, эти расширенные версии протоколов OSPF и IS-IS создают в сети более интенсивный служебный трафик, нежели базовые протоколы, однако это оправданные затраты.

При маршрутизации, основанной на ограничениях (CSPF — маршрутизации), модуль маршрутизации вычисляет маршрут в сети, используя информацию, хранящуюся в TED. Расчет может быть *тактическим* или *стратегическим*.

Тактический расчет используется при возникновении аварий или перегрузок на маршруте, и вместо маршрута, рассчитанного, например, IGP-протоколом, будет автоматически создан обходный TE-маршрут.

Стратегический расчет может быть разделен на *online* и *offline маршрутизацию*. При стратегической online-маршрутизации все TE-маршруты (и TE-туннели) вычисляются между пограничными узлами MPLS-домена в соответствии с заданными ограничениями, и производится соответствующее резервирование ресурсов, причем вычисления выполняют сами узлы.

Стратегическая offline-маршрутизация отличается от online-маршрутизации только тем, что для вычисления всех маршрутов используется отдельный сервер, имеющий общее видение сети и ее ресурсов. Таким образом, при offline-маршрутизации можно добиться наиболее эффективного использования сетевых ресурсов, так как в сети не будет возникать противоречивых запросов резервирования, а при вычислении маршрута будут учитываться и остальные маршруты (TE-туннели).

Но online-маршрутизация быстрее адаптируется к изменениям, происходящим в сети, поэтому на практике эти два подхода часто используются совместно: LSP рассчитывает offline-сервер, а online-маршрутизация запускается лишь тогда, когда рабочие характеристики LSP перестают удовлетворять предъявляемым к ним требованиям, или происходит заметное изменение состояния сети. Кроме того, процессы offline- и online-маршрутизации могут запускаться с разным периодом, причем первый из них, имеющий больший период, вычисляет маршруты, а второй производит их коррекцию.

Вычисленные таким образом маршруты для LSP позволяют организовать в MPLS-сети сами тракты. Они создаются средствами изображенного в правой части рис. 9.4 компонента сигнализации. Модуль маршрутизации передает в сигнальный модуль данные о последовательности *абстрактных узлов*. При создании каждого LSP происходит обмен протокольными сообщениями, в процессе которого вдоль маршрута распределяются метки и резервируются сетевые ресурсы. Могут также учитываться приоритетность создаваемых LSP, производиться вытеснение низкоприоритетного трафика и обрабатываться ситуации соперничества за ресурсы.

После того как все LSP созданы, подсистема TE продолжает эффективно их поддерживать, используя свои дополнительные возможности. Здесь нельзя не упомянуть о такой функции, как *быстрая ремаршрутизация FRR (Fast ReRoute)*. Поскольку с появлением мощных и производительных маршрутизаторов возможности MPLS, упрощающие процесс маршрутизации, постепенно отходят в тень, основными преимуществами этой технологии становятся, во-первых, гибкое управление прохождением трафика (собственно, основная задача TE) и, во-вторых, именно FRR.

Важность быстрой ремаршрутизации обусловлена тем, что для Оператора весьма опасны потери при выходе из строя звена. Разумеется, о такой ситуации будет информирован оконечный маршрутизатор, он предпримет попытку создать новый LSP (LSP-туннель) в обход поврежденного участка сети, но из-за задержек, возникающих при передаче сигнальных сообщений к оконечному узлу в процессе расчета нового маршрута, могут произойти ощутимые потери данных в неисправном звене. FRR обеспечивает защиту от этих потерь, ремаршрутизируя трафик, проходящий по LSP, в обход поврежденного звена. При этом решение о ремаршрутизации принимается узлом, непосредственно соединенным с неисправным звеном. Такая локальная ремаршрутизация позволяет предотвратить дальнейшую потерю пакетов и выиграть время для того, чтобы информировать оконечный узел и создать новый LSP.

Приведенный на рис. 9.7 пример показывает, как FRR используется для защиты трафика, переносимого между узлами LSR1 и LSR4 при проходе через звено LSR2-LSR3. Для LSP от LSR1 к LSR4 через LSR2 и LSR3 используются метки 25 и 9. Для защиты звена LSR2-LSR3 создается резервный туннель от LSR2 к LSR3, проходящий через узлы LSR5 и LSR6. В этом резервном туннеле будут использоваться метки 38 и 15. Когда LSR2 обнаружит, что звено между ним и LSR3 стало недоступно, он просто отправит трафик, адресованный в сторону LSR3, в резервный туннель. Это выполняется помещением метки 38 наверх стека после выполнения обычной процедуры *Label Swapping* (замены метки 25 меткой 9).

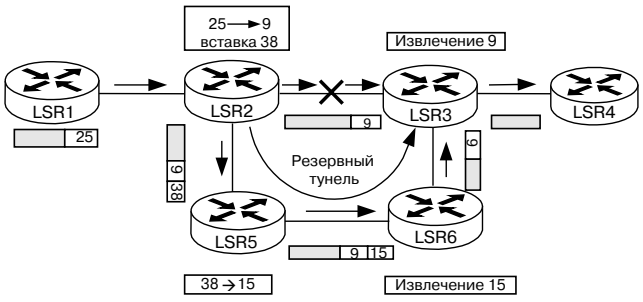


Рис. 9.7. Пример применения FRR

Помимо того, что FRR обеспечивает дополнительную надежность при передаче трафика, она является очень хорошо масштабируемым решением, так как все LSP, проходящие через поврежденное звено, могут быть переведены в единственный резервный туннель, созданный с учетом тех же ограничений, что и при расчете защищаемых LSP.

На момент написания книги документ, специфицирующий оба варианта использования функции FRR, находился в состоянии *draft*-проекта комитета IETF, и до момента стандартизации реализации FRR разными производителями оборудования могут различаться.

9.6. Сравнение протоколов CR-LDP и RSVP-TE

9.6.1. Сравнение функциональных возможностей

Следует сразу же отметить, что оба рассматриваемых протокола отвечают требованиям документа RFC 2702 к сигнализации MPLS. Однако для выполнения этих требований CR-LDP и RSVP-TE используют различные механизмы, хотя и схожего в реализации ряда функций очень много. В этом параграфе рассмотрены основные функции исследуемых протоколов и варианты их реализации в каждом из них.

Возможность присвоения и переноса параметров трафика и QoS в RSVP-TE реализуется посредством передачи в сообщениях непрозрачных данных, предназначенных для подсистемы управления трафиком. CR-LDP может определять *правила для пограничных узлов (edge rules)* и рекомендации для *промежуточных пересылок (per hop behaviors)*, базирующиеся на скорости передачи данных, на полосе пропускания звена и на весах, присвоенных этим параметрам.

О неисправности протокол RSVP-TE извещает сообщением об ошибке, но момент его отправки зависит от установленных значений таймеров для обновления состояния. CR-LDP же пользуется для извещения о неисправности возможностями транспортного протокола TCP.

Восстановление после неисправности протокол RSVP-TE обеспечивает ремаршрутизацией тракта в соответствии с принципом *«make-before-brake»*, когда сначала создается новый LSP, затем в него переводится трафик, и лишь после этого разрушается неисправный тракт. CR-LDP позволяет задать политику обработки неисправности в каждом из узлов, через которые проходит LSP.

Обнаружение закольцованных маршрутов требуется лишь для не строго заданных маршрутов и в RSVP-TE производится при помощи объекта RRO. Протокол CR-LDP использует для этого традиционный для LDP объект Path_Vector_TLV. Оба этих объекта могут также применяться для определения того, какой маршрут использует LSP.

Для управления трактами в обоих протоколах применяется идентификация каждого LSP идентификатором LSP ID. При этом RSVP-TE может идентифицировать и туннель (Tunnel ID), позволяя перевести его из одного LSP в другой.

Оба протокола поддерживают функцию вытеснения высокоприоритетным трафиком низкоприоритетного. В обоих протоколах это достигается присвоением приоритетов удержания и захвата ресурса.

Разумеется, оба протокола могут определять для тракта явно заданный маршрут, причем оба они могут работать с абстрактными узлами и со строго и не строго заданными маршрутами.

9.6.2. Сравнение технических характеристик

Ключевыми различиями между протоколами CR-LDP и RSVP являются используемые тем и другим транспортные протоколы и то, в каком направлении — прямом или обратном — производится резервирование ресурсов. Из различия по этим двум признакам вытекают и многие другие различия этих протоколов. В чем сходны и чем различаются протоколы CR-LDP и RSVP-TE, показывает табл. 9.1.

Таблица 9.1 Сравнение протоколов CR-LDP и RSVP-TE

	CR-LDP	RSVP-TE
Используемый транспортный протокол	TCP	Исходный IP
Надежность операторского класса	Нет	Да
Поддержка трафика «много точек — точка»	Да	Да
Поддержка вещательной рассылки	Нет	Нет
Поддержка слияния LSP	Да	Да
Явная маршрутизация	Со строгими и с не строгими участками маршрута	Со строгими и с не строгими участками маршрута
Ремаршрутизация LSP	Да	Да
Закрепление маршрута	Да	Да, путем записи маршрута
Вытеснение потоков в LSP	Да, на основе приоритета	Да, на основе приоритета
Средства безопасности	Да	Да
Защита LSP	Да	Да
Состояние LSP	Жесткое	Нежесткое
Регенерация состояния LSP	Не требуется	Периодическая, по участкам
Резервирование совместно используемых ресурсов	Нет	Да
Обмен параметрами трафика	Да	Да
Управление трафиком	В прямом направлении	В обратном направлении
Авторизация пользователей	Неявная	Явная
Индикация протокола уровня 3	Нет	Да
Ограничения в зависимости от класса ресурса	Да	Нет

Наиболее очевидным различием протоколов CR-LDP и RSVP, показанным в таблице 9.1, является то, какой *транспортный протокол* используется для передачи запросов меток. RSVP использует для этого протокол IP, не ориентированный на соединение. CR-LDP использует транспортный протокол UDP, но только для обнаружения одноранговых маршрутизаторов сети MPLS; для передачи протокольных сообщений он использует ориентированные на соединение TCP-сеансы. В RSVP требуется, чтобы все принятые пакеты IP, переносящие сообщения протокола RSVP, доставлялись к модулю протокола без ссылки на фактический IP-адрес получателя, содержащийся в пакете. Эта особенность может потребовать внесения незначительных изменений в реализацию IP. Имеются два негативных аспекта использования протоколом CR-LDP транспорта TCP. Во-первых, реализованный в TCP механизм предотвращения перегрузки может заметно тормозить передачу информации между маршрутизаторами. Во-вторых, когда между двумя LSR имеется только одно TCP-соединение, TCP принудительно вводит дисциплину FIFO обслуживания очередей сообщений, при которой для критически важного сообщения нет возможности прийти к адресату раньше менее важного сообщения, которое было отправлено первым. Кроме того, когда теряется какой-либо пакет, все сообщения, следующие за этим пакетом, задерживаются до тех пор, пока не будет успешно выполнена его повторная передача.

Протокол CR-LDP наследует все функции *обеспечения безопасности*, которые есть у протокола TCP. К сожалению, TCP уязвим со стороны атак, имеющих цель нарушить обслуживание, при которых рабочие характеристики TCP-сеанса могут серьезно пострадать в результате несанкционированного доступа к сети, что отрицательно повлияет на работу протокола CR-LDP. Адресатом сообщений Path протокола RSVP является выходной LSR, а не промежуточные LSR. Это значит, что *IPSec* (серия разработанных комитетом IETF проектов стандартов обеспечения аутентификации и защиты передаваемых по протоколу IP пакетов путем шифрования) не может использоваться, поскольку промежуточные LSR будут не в состоянии получить доступ к информации, содержащейся в сообщениях Path. Но RSVP имеет свои механизмы аутентификации и авторизации пользователей, позволяющие проверять полномочия каждого отправителя сообщений и не допускать несанкционированного или злонамеренного резервирования ресурсов. Аналогичные возможности могли бы быть специфицированы и для протокола CR-LDP, но ориентированный на соединение характер TCP-сеанса делает это требование менее актуальным, т.к. TCP может использовать какой-нибудь стандартный криптографический алгоритм.

IP-трафик *«точка — группа точек»* не поддерживается протоколами CR-LDP и RSVP-TE. Но базовый протокол RSVP первоначально разрабатывался с учетом возможности резервировать ресурсы для

деревьев многоадресной рассылки по протоколу IP, так что его проще расширить для поддержки многоадресного трафика. Поддержка многоадресной рассылки в настоящее время не определена ни для одного из существующих протоколов распределения меток, но 28 января 2004 года в рабочей группе MPLS IETF был предложен *draft* документ, который, если он будет принят, должен определить требования к расширению протокола RSVP-TE для поддержки режима «точка — группа точек». Так что у протокола RSVP-TE может в ближайшем будущем появиться функция многоадресной рассылки.

Еще во время спецификации протокола RSVP возникли сомнения относительно возможности его применения в крупных сетях из-за недостаточной *масштабируемости*. Эти сомнения были вызваны тем фактом, что RSVP резервирует ресурсы для индивидуальных «микропотоков», т.е. для потоков таких данных, которые соответствуют, как правило, одному пользовательскому приложению, функционирующему на паре рабочих станций. Число «микропотоков», проходящих через один маршрутизатор в крупной IP-сети, измеряется миллионами, что может предъявить серьезные требования к объему памяти и к производительности маршрутизаторов. Это обстоятельство иногда приводит к утверждению о слабой масштабируемости RSVP. Но в действительности мы говорим об отсутствии хорошего масштабирования RSVP только тогда, когда базовый RSVP используется, чтобы резервировать ресурсы для «микропотоков» индивидуальных пользовательских приложений, а когда протокол RSVP-TE используется для создания явно заданных трактов LSP, такое «микрорезервирование» не делается. Доминирующим фактором, определяющим масштабируемость протокола создания LSP, является число создаваемых трактов LSP, которое, разумеется, от протокола не зависит.

Все ориентированные на соединение протоколы требуют *хранения данных* о состояниях соединений, как на оконечных LSR, так и на промежуточных. При использовании протокола RSVP-TE требования во многом схожи во всей сети, потому что информация о состоянии должна храниться и периодически обновляться в каждом LSR. В эту информацию включаются параметры трафика, данные о резервированных ресурсах и об явно заданных маршрутах. Объем этой информации составляет порядка 500 байтов на один LSP.

Протокол CR-LDP требует, чтобы входной и выходной LSR хранили сходные объемы информации о состоянии, включая параметры трафика и данные об явно заданных маршрутах. Суммарный объем информации о состоянии, требуемый для функционирования протокола CR-LDP, составляет на оконечных узлах тоже порядка 500 байтов. В промежуточных LSR возможно снизить объем хранимой информации примерно до 200 байтов за счет отказа от функции модификации LSP, т.е. от возможности ремаршрутизации LSP или

от учета изменений ресурсов. Следует отметить, что буферы пересылки данных, необходимые для обеспечения гарантированного QoS при передаче по LSP, будут иметь намного больший размер, чем объем памяти, нужный для хранения данных о состоянии. Таким образом, различие между протоколами RSVP и CR-LDP в сети MPLS, не требующей поддержки модификации LSP, делается менее существенным.

Под *надежностью операторского класса* понимается коэффициент готовности в «пять девяток», т.е. 99,999%. Высокая готовность достигается за счет своевременного обнаружения и устранения неисправностей без какого-либо (или, в крайнем случае, только минимального) нарушения обслуживания. Вопросы обеспечения живучести LSP при возникновении программных или аппаратных отказов относятся к реализации оборудования, и эти вопросы обязан рассматривать и решать каждый поставщик сетевого оборудования MPLS.

В связи с тем, что протокол RSVP использует транспортный механизм без установления соединения, он хорошо приспособлен к системе, которая должна быть устойчивой к аппаратным отказам или сбоям программного обеспечения. Сведения о любых управляющих действиях, теряющиеся во время аварийного переключения на резервную систему, могут быть восстановлены посредством встроенного в протокол RSVP-TE механизма регенерации состояния.

С другой стороны, протокол CR-LDP предполагает надежную доставку управляющих сообщений и поэтому он более чувствителен к аварийным переключениям на резерв. Кроме этого, протокол TCP очень сложно сделать отказоустойчивым, и поэтому аварийное переключение на резервный стек протоколов TCP приводит к потере TCP-соединений. Такая ситуация интерпретируется протоколом CR-LDP как отказ всех соответствующих LSP, которые надо создавать заново от входного LSR.

Таким образом, изначально протокол RSVP может обеспечивать лучшие решения для сетей MPLS с высоким уровнем готовности. Ситуацию с CR-LDP исправляет документ RFC 3479, специфицирующий механизмы отказоустойчивости для этого протокола. После внедрения этих расширений CR-LDP приблизится по характеристикам обеспечения высокой готовности к протоколу RSVP.

С надежностью непосредственно связано *обнаружение отказов в звеньях и в маршрутизаторах.* Если два LSR непосредственно связаны двухточечным каналом, например, каналом ATM, неисправность в LSP можно, как правило, обнаружить путем мониторинга состояния интерфейсов LSP. Например, если в канале ATM пропадает сигнал, то и протокол CR-LDP, и протокол RSVP могут использовать уведомление об отказе в интерфейсе для обнару-

жения отказа LSP. Если два LSR соединены друг с другом через совместно используемую среду передачи, например, Ethernet, или соединены друг с другом не непосредственно, а через облако WAN, то они не обязательно получают уведомление об отказе звена из канального оборудования. В таких случаях задача обнаружения отказов LSP перекладывается на средства, имеющиеся в протоколах сигнализации. Протокол CR-LDP использует обмен сообщениями Hello и Keepalive для того, чтобы удостовериться, что смежный LSR и звено продолжают оставаться активными. Несмотря на то, что протокол TCP имеет встроенную систему поддержания соединения, она, как правило, слишком медленно, с точки зрения нужд LSP сети MPLS, реагирует на отказы в звене и в маршрутизаторе. В протоколе RSVP периодически передаваемые для регенерации состояния сообщения Path и Resv образуют фоновый трафик, который указывает на то, что звено продолжает оставаться работоспособным. Однако, для сведения к минимуму этого трафика в относительно стабильной сети, для таймера регенерации может быть установлено достаточно большое значение. В протоколе RSVP-TE для подтверждения активности и работоспособности звена и смежного LSR может использоваться обмен сообщениями Hello. Таким образом, методика и быстрота обнаружения отказов в обоих сравниваемых протоколах схожи.

Организация лямбда-сетей связана со значительным количеством проблем, возникающих при реализации технологии MPLS в оптической сети. Всеми преимуществами спектральной коммутации можно воспользоваться только в том случае, если коммутация LSP выполняется аппаратными средствами без помощи ПО. Число волн, однако, слишком мало по сравнению с вероятным числом LSP. Кроме того, возможности одной волны значительно превышают обычные требования одного LSP, поэтому организация взаимно-однозначного соответствия между ними была бы непозволительно расточительной тратой сетевых ресурсов. Рабочей группой IETF по вопросам MPLS выпущены документы RFC, содержащие необходимые дополнения как для протокола RSVP-TE, так и для CR-LDP, обеспечивающие их работоспособность в лямбда-сетях в рамках концепции GMPLS, о чем будет сказано в следующей главе.

Важно отметить, что в контексте *управления трафиком* протоколы CR-LDP и RSVP-TE выполняют резервирование ресурсов на разных стадиях процесса создания LSP.

Протокол CR-LDP переносит полную информацию о параметрах трафика в сообщении запроса метки Label Request. Это позволяет каждому маршрутизатору сети MPLS управлять трафиком при создании LSP. Параметры трафика могут согласовываться по мере продвижения процесса создания LSP, а окончательные значения

параметров передаются в обратном направлении в сообщении назначения метки Label Mapping, что обеспечивает контроль доступа и резервирование ресурсов в каждом LSR в реальном времени. Этот подход позволяет не создавать LSP по маршруту, который не имеет в данный момент времени достаточных ресурсов.

Протокол RSVP-TE переносит в сообщении Path набор параметров трафика в виде спецификации потока данных отправителя Tspec. Эта спецификация описывает данные, которые будут передаваться по LSP. Транзитные LSR могут анализировать эту информацию и на ее основе принимать решения о маршрутизации. Однако только тогда, когда сообщение Path дойдет до выходного LSR, спецификация Tspec будет преобразована в спецификацию Flowspec, которая переносится в обратном направлении сообщением Resv и в которой даются детали резервирования ресурсов, требующихся для LSP. Это означает, что резервирование не происходит до тех пор, пока сообщение Resv не пройдет через сеть, а результатом может стать то, что на выбранном маршруте создать LSP не удастся из-за нехватки ресурсов.

Протокол RSVP-TE содержит необязательную функцию Adspec, посредством которой можно сообщить о доступных ресурсах в сообщении Path. Это позволяет выходному LSR узнать, какие ресурсы доступны, и, в соответствии с этим, модифицировать спецификацию Flowspec, переносимую в сообщении Resv. К сожалению, эта функция не только требует, чтобы ее поддерживали все LSR на маршруте, но также имеет тот очевидный недостаток, что ресурсы, сведения о которых несет сообщение Path, к тому времени, когда будет получено сообщение Resv, уже могут оказаться занятыми другим LSP.

Частичное решение этой проблемы для LSR, использующих протокол RSVP, лежит в области реализации: можно обеспечить предварительное резервирование ресурсов при обработке сообщения Path. Это резервирование будет лишь приблизительным, поскольку оно опирается на спецификацию потока данных отправителя Tspec, а не на спецификацию Flowspec, но, тем не менее, оно может значительно облегчить положение.

Протокол CR-LDP предлагает несколько более жесткий подход к управлению трафиком, особенно в сетях, испытывающих высокую нагрузку.

Протокол RSVP-TE позволяет переносить в сообщениях Path и Resv объект *авторизации пользователей* с непрозрачным содержимым. Эта информация используется при обработке сообщений для контроля доступа в соответствии с установленными правилами. Это позволяет тесно привязать протокол RSVP-TE, поддерживающий распределение меток, к протоколам авторизации и надзора, например, в соответствии с RFC 2749, к COPS (Common Open Policy Service).

В отличие от этого, протокол CR-LDP позволяет переносить в неявном виде только информацию авторизации в форме адресов получателя и класса административного ресурса в параметрах трафика.

Различие между протоколами существует и в *индикации протоколов уровня 3.* Хотя LSP может переносить любые данные, бывают случаи, когда транзитному или выходному маршрутизатору сети MPLS может потребоваться информация о протоколе уровня 3. Если транзитный LSR не в состоянии доставить пакет (например, из-за отказа ресурса), он может передать в обратном направлении специфическое для протокола уровня 3 сообщение об ошибке, уведомляющее отправителя данных о возникшей проблеме. Чтобы этот механизм работал, LSR, который выявляет ошибку, должен знать, какой именно протокол уровня 3 используется. Информация о протоколе уровня 3 может также помочь выходному маршрутизатору пересылать пакеты данных.

RSVP-TE идентифицирует один единственный протокол передачи полезной нагрузки на этапе создания LSP, а CR-LDP и этого делать не в состоянии. Но даже RSVP не в состоянии помочь, когда в один LSP направляется трафик более чем одного протокола.

Трактами LSP, созданными с выбором оптимального маршрута в сети, можно управлять на их входе и их можно контролировать на выходе с помощью *административной базы данных MIB* сети MPLS. Эта MIB в настоящее время разрабатывается, и в рабочей группе на эту тему существует ряд проектов *draft*. В связи с весьма пессимистическими планами работы над CR-LDP эти проекты, скорее всего, будут ориентированы на поддержку RSVP-TE, хотя на ранней стадии разработки проектов MIB, предпочтение отдавалось как раз CR-LDP.

9.6.3. Служебный трафик в RSVP-TE и CR-LDP

Оба протокола создают потоки служебных сообщений для распределения меток, передачи сквозного запроса и сквозного ответа на запрос. Протокол RSVP ориентируется на «нежесткое состояние» маршрутов. Это значит, что протокол должен периодически обновлять состояние каждого LSP между смежными узлами, что позволяет автоматически учитывать изменения в дереве маршрутизации. В качестве транспортного средства протокол RSVP использует IP-дейтаграммы, так что управляющие сообщения могут теряться, и смежный узел, не получив соответствующего уведомления, может прекратить обслуживание. Регенерация состояния LSP гарантирует, что оно надлежащим образом синхронизировано между смежными узлами. Нагрузка на сеть от таких периодических обновлений зависит от чувствительности к отказам, которая регулируется выбором значения таймера регенерации (обновления). Сообщение

Path протокола RSVP будет иметь длину порядка 128 байтов, увеличиваясь на 16 байтов в каждом маршрутизаторе, если используется фиксированный маршрут. Сообщение Resv будет иметь длину порядка 100 байтов. При 10 000 LSP в межузловом звене и периоде регенерации 30 секунд на регенерацию будет потребляться свыше 600 Кбит/с пропускной способности звена. Много это или нет — зависит от характеристик звена и от того, какую это составляет долю от передаваемого по нему трафика.

Протокол CR-LDP не требует от LSR периодической регенерации каждого LSP после его создания. Это достигается благодаря тому, что в качестве транспортного средства для передачи сигнальных сообщений протокола CR-LDP используется протокол TCP. Протокол CR-LDP может обеспечивать надежную доставку сообщений Label Request и Label Mapping. Использование транспортного протокола TCP в тракте сигнализации никакой служебной информации в этот тракт не вносит, а только добавляет 20 байтов к длине каждого сигнального сообщения. Для поддержания связности со смежными узлами протокол CR-LDP использует сообщения Hello, с помощью которых он удостоверяется, что смежные узлы продолжают оставаться активными, и сообщения KeepAlive для мониторинга TCP-соединений. Периодический обмен этими относительно короткими сообщениями ведется для всего звена, а не для каждого из многочисленных LSP, по нему проходящих, и потому эти сообщения практически не оказывают влияния на пропускную способность звена. Таким образом, протокол CR-LDP, в принципе, вносит меньшую нагрузку в сеть, чем протокол RSVP. Но IETF специфицировал документ RFC 2961, в который вошли идеи уменьшения числа сообщений регенерации, требуемых протоколом RSVP. Рассмотрим подробнее решение этой проблемы, предложенное и детализированное в рабочих группах MPLS и RSVP в составе комитета IETF.

С учетом взаимосвязи между надежной доставкой информации и непроизводительными затратами на процедуры регенерации, одним из шагов для снижения объема обрабатываемой информации регенерации могло бы стать добавление в протокол RSVP механизма надежной доставки сообщений. Это было сделано путем определения пары объектов протокола RSVP, а именно объектов MESSAGE_ID и MESSAGE_ID_ACK. Эти объекты обеспечивают надежную доставку сообщений следующим образом. Предположим, что узел LSR1 имеет сообщение протокола RSVP, представляющее новое состояние, например, новое сообщение PATH, которое он должен передать к соседнему узлу LSR2. Узел LSR1 создает локально уникальный идентификатор нового состояния, помещает этот идентификатор в объект MESSAGE_ID и добавляет его к сообщению протокола RSVP, установив в объекте флаг «требуется подтверждение». Получив сообщение, LSR2 подтверждает его получение, передавая

к LSR1 сообщение, которое содержит объект MESSAGE_ID_ACK с тем же идентификатором. Если LSR2 уже имеет какое-либо сообщение, подлежащее передаче к LSR1, он просто включает объект MESSAGE_ID_ACK в это сообщение; в противном случае ему надо будет передать специально созданное сообщение подтверждения ACK. Этот простой механизм может следующим образом обеспечить решение проблемы, связанной с большими объемами информации регенерации. Узел, поддерживающий RSVP, может использовать очень короткий таймер регенерации для передаваемых им сообщений, приводящих к установлению нового состояния. При получении от соседнего узла подтверждения того, что новое состояние установлено, он может переключиться на очень длинный таймер регенерации. Таким образом, объем информации регенерации снижается без ущерба для своевременности резервирования.

При всем этом такой механизм отличается от «жесткого состояния», поскольку сообщения регенерации все равно должны периодически передаваться. Следовательно, сохраняются присущие не жесткому состоянию свойства «самовосстановления», включая интерактивную обработку отказов. Кроме того, этот механизм обладает обратной совместимостью, т.е. не создает никаких проблем, если в одних узлах он реализован, а в других — нет.

Дальнейшее снижение объема информации регенерации достигается с помощью механизма, который называется *сокращенной* или *ускоренной регенерацией (summary refresh).* Суть его в том, что после того как идентификатор сообщения уже привязан к сообщению протокола RSVP, для регенерации состояния нет необходимости передавать снова и снова полное сообщение, как это обычно делается. Узел может просто передавать сам идентификатор сообщения. Кроме того, возможна упаковка довольно большого числа идентификаторов сообщений в одно сообщение, так что одно сообщение протокола RSVP может обновлять целый ряд состояний. Такая возможность полезна, поскольку часто требуются значительные вычислительные затраты только на получение сообщения в процессор маршрутизатора, вне зависимости от того, как велико это сообщение или от того, каковы затраты на обработку его содержимого.

Для поддержки механизма сокращенной регенерации было определено новое сообщение протокола RSVP — сообщение SREFRESH. При адресации к определенному устройству это сообщение содержит просто перечень ранее переданных идентификаторов сообщений. Узел, получающий такое сообщение, ведет себя так, как если бы он получил набор сообщений регенерации, каждое из которых идентично сообщению, в котором был первоначально передан объект MESSAGE_ID, то есть он сбрасывает таймер регенерации.

Одна из возможных проблем — получение идентификатора MESSAGE_ID, который не распознается. Это может произойти, если следующий участок маршрута изменился, а передающий узел это изменение не обнаружил. Для выхода из такой ситуации принимающий узел, не распознавший идентификатор сообщения, может передать отрицательное подтверждение MESSAGE_ID_NAK. Передающий узел, принявший такое подтверждение, должен передать полное сообщение протокола RSVP, к которому относился MESSAGE_ID.

В результате всех этих усовершенствований были решены проблемы, связанные с масштабируемостью RSVP при его использовании для MPLS. Появилась возможность многократно производить резервирование ресурсов и трактов LSP при незначительном объеме трафика регенерации.

Проблема снижения объема передаваемой служебной информации в RSVP-TE была рассмотрена нами столь подробно потому, что обычно одним из аргументов противников RSVP-TE был именно трафик обновления состояний. Теперь же видно, что после реализации в протоколе RSVP механизма, сокращающего объем информации регенерации, различие между RSVP-TE и CR-LDP по этому признаку становится несущественным.

9.6.4. Сравнение ремаршрутизации в RSVP-TE и CR-LDP

Рассмотрим вопрос изменения маршрута для LSP после получения уведомления об отказе или при изменении топологии сети. Предварительное программирование резервных (альтернативных) маршрутов для тракта LSP называется *защитой LSP* и рассматривается в этом подразделе ниже. Явно заданный LSP может быть ремаршрутизирован только входным LSR — отправителем данных. Следовательно, об отказе в некоторой точке LSP должен быть информирован входной LSR, и при этом постепенно разрушается весь LSP. Однако не строго специфицированный участок LSP с явным маршрутом и любая часть LSP, маршрут для которого задавался по участкам, могут быть ремаршрутизированы, если обнаружен отказ звена или смежного маршрутизатора (*локальное восстановление*), или если стал доступен лучший маршрут, или если ресурсы LSP требуются для создания нового LSP с более высоким приоритетом (*приоритетное вытеснение*). Выше уже рассматривался случай, когда сеть поддерживает функцию Fast ReRoute.

Ремаршрутизацию, управляемую входным узлом, поддерживает как протокол CR-LDP, так и протокол RSVP-TE, хотя имеются незначительные различия в том, как это делается.

Маршрутизатор LSR, использующий протокол RSVP-TE, может определить новый маршрут, как только станет доступным и/или

потребуется альтернативный маршрут, просто путем обновления LSP, в результате чего выбирается другой следующий маршрутизатор. Старый тракт при этом не используется и разрушается после срабатывания таймера, поскольку сообщения регенерации по нему больше не передаются. Ясно, что при таком способе непроизводительно тратятся ресурсы старого LSP.

Этого можно избежать, передав по явно заданному маршруту сообщение *PathTear* или *ResvTear*. При этом активизируется *механизм включения нового тракта до разрыва старого (make-before-break)*, т.е. механизм, при котором старый тракт продолжает использоваться (и обновляться) все время, пока создается новый тракт, а после его создания LSR, производящий ремаршрутизацию, выполняет переключение на новый тракт и разрушает старый тракт. Эта методика позволяет избежать двойного резервирования ресурсов как в протоколе CR-LDP (использованием значения *modify* флага действия в сообщении Label Request), так и в протоколе RSVP-TE (использованием фильтров при стиле резервирования Shared-Explicit).

В нестабильных сетях интенсивный служебный трафик, обеспечивающий ремаршрутизацию нестрого специфицированных участков LSP на промежуточных LSR при появлении лучших маршрутов, может приводить к возникновению перегрузок. Для того чтобы это предотвратить, нестрого специфицированный участок маршрута может закрепляться следующим образом.

В протоколе CR-LDP это делается просто путем пометки нестрогого участка явно заданного маршрута как закрепленного. Это означает, что как только маршрут будет определен, к нему будут относиться, как к строго специфицированному маршруту, который изменяться не может.

В протоколе RSVP закрепление требует некоторой дополнительной обработки. Пусть первоначальный маршрут специфицирован с нестрогим участком. Чтобы информировать входной LSR о выбранном маршруте, в сообщениях Path и Resv используется объект Record Route (RRO). Входной LSR может затем использовать эту информацию для повторной передачи сообщения Path, которое будет содержать строго специфицированный явный маршрут.

И RSVP-TE, и CR-LDP используют гибкий подход к ремаршрутизации LSP и к применению механизма с включением нового LSP до разрыва старого. Протокол CR-LDP опирается на внесенное в его спецификацию добавление, позволяющее поддерживать включение нового до разрыва старого, а протокол RSVP-TE требует дополнительного обмена сообщениями для закрепления маршрута.

Стоит отметить один из недостатков протокола CR-LDP, связанный с использованием протокола TCP для LDP-сеансов. При работе CR-LDP имеют место дополнительные затраты времени на опреде-

ление отношения смежности между двумя LSR. Перед тем как маршрутизаторы смогут инициировать LDP-сеанс, они должны пройти процедуру вхождения в связь по протоколу TCP. Это дает преимущество протоколу RSVP-TE, который не требует установления соединения перед началом процедуры распределения меток. Можно сказать, что протокол RSVP имеет «облегченные отношения смежности», которые позволяют определять новые взаимосвязи между соседними маршрутизаторами быстро, по мере необходимости. И это важно для выполнения быстрой ремаршрутизации.

Модификация LSP, например, при изменении параметров трафика в LSP, является операцией, эквивалентной ремаршрутизации, хотя при модификации LSP изменение маршрута не является обязательным. Следовательно, эта функция всегда присутствует в протоколе RSVP-TE и будет присутствовать в протоколе CR-LDP при условии, что реализация протокола поддерживает значение modify флага действия в сообщениях Label Request (дело в том, что ранние реализации протокола модификацию LSP не поддерживают).

Защита LSP заключается в программировании резервных путей для тракта LSP с автоматическим переключением на резервный тракт при отказе основного тракта. Несмотря на то, что с концептуальной точки зрения эта функция тоже аналогична ремаршрутизации, защита LSP обычно рассматривается как намного более важная операция, целью которой является оперативное переключение на новый тракт с минимально возможным прерыванием передачи данных по LSP (как правило, расчетная длительность прерывания должна быть менее 50 мс). В обоих сравниваемых протоколах могут поддерживаться несколько уровней защиты LSP.

Простейшим видом защиты LSP является попытка входного или транзитного LSR выполнить ремаршрутизацию LSP немедленно по получении уведомления об отказе. Такая возможность существует в обоих протоколах, однако, из-за необходимости передавать разнообразные сигнальные сообщения, аварийное переключение на новый маршрут происходит относительно медленно (обычно это занимает, как минимум, несколько секунд). Такая невысокая скорость переключения неприемлема для IP-телефонии и некоторых других приложений реального времени.

Намного более быстрой защиты LSP можно достичь, если звено между двумя LSR защищено схемой защиты на уровне 2. Такая защита прозрачна для LSP и может применяться с любым из двух сравниваемых протоколов. Впрочем, реализация защиты на уровне 2 может оказаться дорогим делом, и сама защита ограничена участком LSP между двумя соседними маршрутизаторами. К тому же, защита звена не обеспечивает защиты от отказа отдельных LSR.

9.6.5. Сравнение протоколов по их внедрению

Единственным фактором, влияющим на практическое внедрение как RSVP-TE, так и CR-LDP, является предпочтение производителей соответствующего сетевого оборудования. После приостановки IETF работы над протоколом CR-LDP, производители были вынуждены сконцентрировать свои усилия на RSVP-TE, но т.к. многие сети уже используют CR-LDP, и Операторы не спешат от него отказываться, производители продолжают поддерживать и решения на его основе.

Несмотря на то, что стандартный протокол RSVP уже в течение ряда лет предоставляется многими поставщиками оборудования и является распространенным сетевым протоколом, те изменения, которые требуется внести в стандартный RSVP для поддержки распределения меток, настолько нетривиальны, что протокол RSVP-TE становится во многих отношениях новым протоколом.

Протокол CR-LDP базируется на идеях, которые были реализованы в ведомственных сетях уже более десяти лет назад, но в качестве стандартизованного протокола IETF он является относительно молодым. Как уже отмечалось выше, производители оборудования в настоящее время либо поддерживают оба протокола, либо только RSVP-TE, хотя первоначально такие компании, как Nortel Networks и Nokia активно продвигали решения на базе протокола CR-LDP, а Cisco Systems и Juniper Networks занимались протоколом RSVP-TE.

Крупные компании-поставщики телекоммуникационного оборудования обычно помещают в Интернет обновления и новые версии своего программного обеспечения, которые либо доступны свободно, либо предназначены только для авторизированных клиентов компании. В новых версиях учитываются изменения, внесенные в тот или иной протокол международными стандартизирующими организациями и исследовательскими центрами самих компаний. Но обычно это — уже готовое программное обеспечение, предназначенное для работы с оборудованием определенного типа. Компания Nortel Networks, продолжая поддержку протокола CR-LDP, поместила на своем Web-сайте свободно доступный исходный код своей программной реализации CR-LDP. Таким образом, Операторам, нуждающимся в реализации функций MPLS-TE в уже установленном сетевом оборудовании, несколько проще делать это с протоколом CR-LDP, чем с RSVP-TE. При выборе же нового оборудования, предпочтение, скорее всего, будет отдаваться RSVP-TE, как развивающемуся и совершенствующемуся протоколу.

Другим вопросом, который встает перед Оператором, внедряющим в своей сети новый протокол, является функциональная совместимость версий уже выбранного протокола. Всегда существуют

два требующих ответа вопроса о совместимости: поддерживают ли две реализации протокола одинаковые опции и интерпретируют ли они спецификации одинаково. Протокол RSVP-TE имеет больше вариантов реализации, чем протокол CR-LDP, и поэтому, на первый взгляд, более высок риск того, что реализации RSVP-TE окажутся функционально несовместимыми. Единственным способом удостовериться, что две реализации протокола корректно взаимодействуют друг с другом, является, разумеется, тестирование их функциональной совместимости, о чем будет сказано в главе 11. Участие в совместном тестировании функциональной совместимости является обязательным для всех поставщиков сетевого оборудования MPLS.

Поскольку в сети MPLS-TE обычно функционирует и традиционный для MPLS механизм распределения меток, то есть протокол LDP, мы рассмотрим также вопрос о его стыковке с анализируемыми здесь протоколами. В связи с тем, что протокол CR-LDP построен как расширение протокола LDP, реализациям CR-LDP легче поддерживать и функции LDP. С другой стороны, протокол RSVP-TE является полностью самостоятельным протоколом, и хотя RSVP-TE тоже поддерживает последовательное создание LSP по участкам, для поддержки функций протокола LDP в нем должен быть реализован дополнительный стек протоколов. Этот факт очевиден, но совсем не очевидно, что лучше: иметь больше протоколов или меньше. Например, IP, ICMP и ARP являются тремя самостоятельными протоколами, которые требуются для передачи IP-пакетов по Интернет, и кажется маловероятным, что IP-сети функционировали бы лучше, если бы эти три протокола вдруг каким-то образом слились в один. Можно сказать только то, что это различие протоколов RSVP-TE и CR-LDP не говорит о существенном преимуществе последнего.

Приведенный анализ двух протоколов сигнализации в сетях MPLS-TE позволяет оценить особенности применения каждого из них. В целом оба протокола обладают сегодня практически одинаковыми возможностями и техническими характеристиками. Но это сегодня. После решения рабочей группы IETF по вопросам MPLS прекратить работы по CR-LDP, этот протокол со временем начнет уступать позиции все совершенствующемуся протоколу RSVP-TE. И в будущем, судя по количеству относящихся к протоколу RSVP-TE документов RFC, находящихся на рассмотрении в IETF, этот разрыв будет лишь увеличиваться.

Глава 10
Эволюция к GMPLS

10.1. MPLambS и GMPLS

Принципы MPLS, описанию которых посвящены предыдущие 9 глав, оказались столь удачными (в чем читатель, добравшийся до этого места книги, уже имел возможность убедиться), что дальнейшая эволюция MPLS и появление ее расширенной версии были вполне предсказуемы. Набор утвержденных спецификаций *Generalized MPLS* появился в январе 2003 года, однако всплеска публикаций и обсуждений не последовало. Но и нельзя сказать, что появление GMPLS прошло незамеченным и не оказало влияния на развитие концепции многопротокольной коммутации по меткам в целом.

Наибольшее внимание привлек частый случай GMPLS, получивший название *MPLambdaS — многопротокольная лямбда (т.е. световой волны) коммутация*, иногда именуемая в публикациях *MPλS,*. Здесь лямбды (lambdas) и являются метками, что, по мнению авторов, сыграет огромную роль, как в качестве технологии оптического управления, так и при использовании в технологии коммутации пакетов физического пути поверх оптического. Столь оптимистический прогноз роли MPLambdaS в построении оптических сетей связан с уменьшением количества функциональных уровней, необходимых для выполнения тех же самых задач, поскольку благодаря использованию MPLS непосредственно поверх уровня DWDM устраняется необходимость в таких уровнях как ATM и SDH.

Идея о возможности распространить парадигму замены меток на новые оптические технологии была впервые представлена как спецификация управления световыми путями, или оптическим следами. Каждый *узел OXC (Optical Cross-Connect)* взаимодействует, подобно MPLS LSR, с плоскостью управления. Между соседними OXC создается отдельный канал управления IP. Световые пути становятся трактами LSP, а выбор лямбд и портов кросс-соединения оказывается процессом, который аналогичен процессу назначения

меток и привязки их к FEC, обсуждавшемуся в первых двух главах. Соответственно, протоколы сигнализации MPLS, рассмотренные в главах 3 и 4 — CR-LDR и RSVP-TE,— видоизменяются для формирования световых путей.

Кстати, при обсуждении CR-LDP и RSVP упоминалось, что IETF приняла решение сконцентрировать свои усилия на RSVP-TE и прекратить развитие CR-LDP. Это решение как раз и было связано с GMPLS. Дело в том, что в феврале 2002 года IETF опубликовала документ RFC 3468, речь в котором шла о расширении возможностей сигнализации MPLS, в частности, для поддержки QoS. Там были предложены два практически равноценных варианта: либо усовершенствовать протокол LDP, добавив в него возможности поддержки QoS, либо в протокол RSVP, уже зарекомендовавший себя как средство обеспечении QoS, внедрить функции распределения меток, причем в документе были приняты оба варианта, и их развитие шло параллельно. А в июне 2002 года был опубликован обзор первых реализаций сетей GMPLS, показавший, что из 22 решений 21 использовало в качестве протокола сигнализации RSVP-TE и лишь одно — CR-LDP. Встал вопрос о целесообразности поддержки двух функционально схожих стандартов, и было принято решение оставить CR-LDP в качестве *«proposed standard»*, но дальнейшей работы по его доработке не вести, а все усилия сфокусировать на RSVP-TE. Такое решение рабочей группы по MPLS определило основное направление ее работ, что отразилось и на содержании этой книги.

В этой главе рассматривая форматы для GMPLS, мы будем использовать не стандартные *Generalized* формата, а сообщения и метки для работы с RSVP-TE.

Описанные в главах 5 и 6 протоколы внутренней IGP-маршрутизации — OSPF и IS–IS — тоже оснащаются специальными расширениями, учитывающими оптические особенности топологии и планирования ресурсов. Создание светового LSP от входного OXC LSR к выходному OXC LSR требует конфигурировать кроссировочное оборудование в каждом транзитном OXC LSR таким образом, чтобы входной порт был связан с соответствующим выходным портом.

GMPLS можно рассматривать как обобщение MPLambdaS и как более универсальное расширение технологий ядра MPLS, которые используют парадигму замены меток и протоколы плоскости управления для различных коммутационных технологий, в первую очередь (но не только), оптических. Эти технологии включают в себя коммутацию с временным разделением (встречающуюся, например, в мультиплексорах SDH, где временные интервалы и являются метками), коммутацию по длинам волн (встречается в OXC, где метками являются частоты) и пространственную коммутацию

(встречается в сетевых устройствах, которые направляют трафик от входного порта или волокна к соответствующему выходному порту или волокну, где метками являются номера портов).

Таким образом, MPLS эволюционирует к GMPLS путем распространения понятия *метка* на временное мультиплексирование TDM, частотное мультиплексирование FDM и пространственное мультиплексирование SDM. Метки больше не являются исключительно дополнительными полями в заголовке пакета сетевого уровня, как это описано в главе 2, они могут быть также оптическими лямбдами и т.п.

Существуют четыре класса трактов, которые можно создавать с помощью сигнализации GMPLS:

- статистически-мультиплексированные тракты — переносят обычные пакеты MPLS, использующие промежуточный заголовок,
- тракты TDM — каждый временной канал является меткой,
- тракты FDM — каждая электромагнитная частота (длина световой волны) является меткой,
- тракты SDM — меткой является позиция, например, местоположение волокна в пучке.

Технология GMPLS специфицирована в документе RFC 3471, описывающем саму технологию, а также в RFC 3472, посвященном сигнализации CR-LDP, и в RFC 3473, описывающем протокол RSVP-TE. Все три документа имеют схожую структуру, но если в RFC 3471 специфицируются только информационные поля сообщений, то в двух других документах эти поля вдобавок обрамляются служебной информацией того или другого протокола.

Это допускает еще одну трактовку слова *Generalized* в названии технологии. Речь идет о том, что GMPLS расширяет и число типов оборудования, входящего в MPLS-сеть. Универсальность новой архитектуры заключается в том, что она может включать в себя LSR, не способные анализировать заголовки пакетов, а производить маршрутизацию, основываясь на временных интервалах, длинах волн или физических портах. Таким образом, звенья между всеми LSR могут быть разделены на следующие классы:

- звенья Packet-Switch Capable (PSC), т.е. звенья между теми LSR, которые способны различать границы пакетов и ячеек и выполнять маршрутизацию, основываясь на содержании их заголовков, например, звенья между обычными MPLS-LSR или ATM-коммутаторами;
- звенья Time-Division Multiplex Capable (TDM), через которые данные маршрутизируются на основе временных интервалов, например, звенья SDH или звенья между цифровыми АТС;

- звенья Lambda Switch Capable (LSC), маршрутизация через которые ведется на основе длины волны, т.е. звенья между оптическими лямбда-коммутаторами;
- звенья Fiber-Switch Capable (FSC), через которые данные маршрутизируются на основе физической среды их переноса, например, звенья между оптическими коммутаторами, работающими с несколькими физическими волокнами.

Поскольку PSC-звенья ничем не отличаются от звеньев, используемых в традиционной MPLS-сети, ниже мы будем различать *PSC-звенья* и *не-PSC-звенья*, так как именно наличие последних обусловило появление большинства опций GMPLS.

Мы видим теперь, что можно формировать иерархию маршрутизации, как это показано на рис. 10.1. Наверху иерархии расположатся FSC-звенья, за ними — LSC, потом TDM и, наконец, PSC. Таким образом, LSP, который начинается и заканчивается PSC-звеном, может (вместе с несколькими другими LSP) быть помещен в LSP, начинающийся и заканчивающийся TDM-звеном. LSP уровня TDM и других уровней также могут быть вложены в LSP, иерархически расположенные выше.

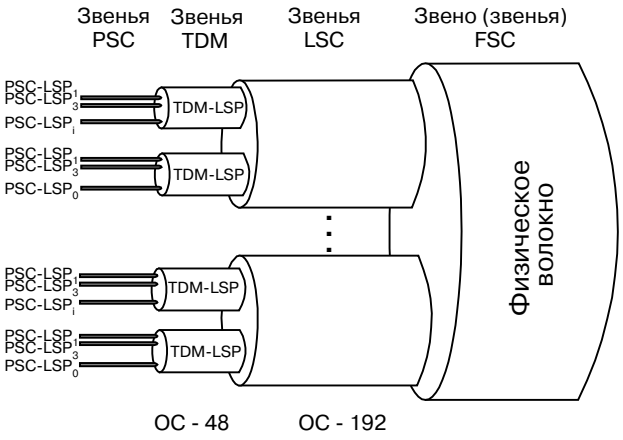


Рис. 10.1. Структура LSP в GMPLS

Для реализации этих нововведений в базовый вариант технологии MPLS введен ряд изменений, коснувшихся основных свойств LSP, процесса распределения меток, однонаправленной природы LSP, обработки ошибок и синхронизации входного и выходного устройства LSP. Рассматривавшиеся в предыдущей главе TE-соединения традиционной MPLS, вдоль которых проходит LSP, могут представлять собой набор магистралей с различным кодированием меток. Таким может быть, например, LSP, состоящий из звеньев между маршрутизаторами, между маршрутизаторами и ATM-коммутаторами и между ATM-коммутаторами. В GMPLS та-

кие возможности получены за счет поддержки трактов, в которых метка кодируется не только как в традиционной MPLS, но и как временной интервал, как длина волны или как физический порт. Правило, существующее в MPLS-TE и определяющее, что LSP должен начинаться и заканчиваться в маршрутизаторах, в GMPLS приняло вид требования, чтобы LSP начинался и заканчивался в однотипных LSR.

Еще одним различием между традиционными LSP и LSP в GMPLS (кроме PSC-типа) является то, что изменение их пропускной способности имеет дискретный характер. Очевидно также, что на участках со звеньями не-PSC требуется меньше меток, чем на участках со звеньями PSC.

В GMPLS распределение меток тоже производится нижним LSR по запросу верхнего, но, в отличие от традиционной MPLS, верхний LSR может сам предложить метку, что оказывается полезным при организации LSP, проходящего через оптическое оборудование. В GMPLS расширены также возможности ограничения диапазона меток, которые может назначать нижний LSR: любой верхний LSR может запретить использование определенных меток на каком-то участке или на всем LSP. Эта опция появилась из-за особенностей оптический сетей, где иногда требуется использовать на маршруте меньший, чем прежде, диапазон длин волн или мультиплексировать все волны в одну волну.

Из-за использования в GMPLS ресурсов, отличных от PSC, появилась необходимость создания двунаправленных LSP. В частности, они используются для предотвращения конфликтной ситуации захвата ресурсов, возникающей при создании различных LSP в отдельных сигнальных сеансах; а также для упрощения процедур восстановления после сбоев в не-PSC оборудовании. Кроме того, при создании двунаправленных LSP меньше время задержки и меньше количество необходимых для этой операции сообщений.

В технологии GMPLS возможно разделение каналов сигнализации и каналов данных, что важно для поддержки тех технологий, где сигнальный трафик не может передаваться вместе с пользовательской информацией. GMPLS предусматривает также возможность расширения протоколов сигнализации специфическими параметрами для поддержки определенных технологий.

GMPLS определяет еще несколько усовершенствований, необходимых для работы MPLS в оптических сетях, в число которых входят связывание каналов, ненумерованные каналы и новый, базирующийся на IP, *протокол LMP (Link-Management Protocol)*. Связывание каналов представляет собой агрегирование атрибутов более чем одного параллельного канала в набор атрибутов, единый для пучка каналов. Выигрыш от такого связывания состоит в уменьшении базы данных о состоянии каналов и в улучшении некоторых

важных характеристик масштабируемости. Ненумерованные — это такие каналы, которые не снабжены IP-адресами. Использование альтернативной идентификации каналов упрощает многие задачи управления ими. Вновь предложенный ненумерованный тег канала является кортежем «идентификатор маршрутизатора/номер канала». LMP — дополнительный протокол управления, который необходим в связи с особыми оптическими требованиями мониторинга и управления между двумя соседними оптическими узлами. LMP обеспечивает верификацию связности каналов, корреляцию свойств каналов, управление управляющими каналами и локализацию неисправностей.

Все эти усовершенствования и рассматриваются в следующих параграфах главы.

10.2. Метки в GMPLS

Для работы с оборудованием оптических и традиционных TDM сетей разработчикам пришлось создать несколько новых форм меток, которые называются *универсальными метками (generalized label).* Введено также несколько дополнительных форм — *предлагаемые метки* и *набор меток*. Структура универсальной метки будет описана в разделе 10.2.2, но сначала опишем процедуру запроса метки в GMPLS.

10.2.1. Запрос универсальной метки

Запрос универсальной метки (рис. 10.2) содержит сведения о характеристиках, необходимых для создания LSP. Эти характеристики определяют кодирование LSP и тип передаваемой по нему полезной нагрузки. Запрос универсальной метки переносит параметр *LSP Encoding Type* — тип кодирования LSP (например SDH, Gigabit Ethernet и т.п.). Параметр определяет природу LSP, а не звеньев, по которым он проходит. Звено может поддерживать несколько форматов кодирования, причем под поддержкой подразумевается способность переносить и коммутировать информацию, представленную в одном или нескольких указанных форматах, в зависимости от ресурсов звена и его пропускной способности. Запрос универсальной метки определяет также тип коммутации для звена; обычно это поле остается постоянным на всем протяжении LSP.

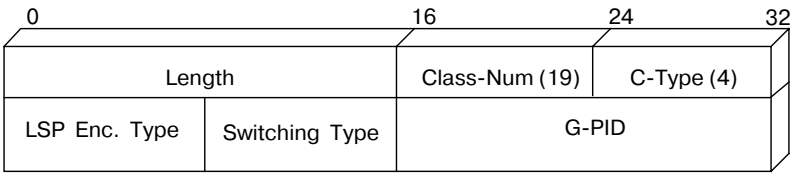


Рис. 10.2. Формат запроса универсальной метки

В изображенном на рис. 10.2 формате запроса универсальной метки представлены следующие параметры.

LSP Encoding Type — 8-битовое поле, определяющее требуемый для LSP тип кодирования. Возможные значения LSP Encoding Type представлены в таблице 10.1.

Switching Type — 8-битовое поле, определяющее тип коммутации для звена. Это поле необходимо для звеньев, которые допускают коммутацию более одного типа. Возможные значения LSP Switching Type представлены в табл. 10.2.

Generalized PID (G-PID) — 16-битовое поле, которое определяет полезную нагрузку, переносимую LSP. Например, это может быть идентификатор пользовательского уровня. G-PID используется конечными узлами LSP или предпоследним узлом LSP. При этом для пакетных и Ethernet LSP используются стандартные значения *Ethertype*. Остальные значения определены в табл. 10.3.

Таблица 10.1. Значения LSP Encoding Type

Поле *LSP Encoding Type*	Десятичное значение	Тип кодирования
00000001	1	Пакетные данные
00000010	2	Ethernet
00000011	3	ANSI/ETSI PDH
00000100	4	Резервное значение
00000101	5	SDH ITU-T G.707 SONET ANSI T1.105
00000110	6	Резервное значение
00000111	7	Digital Wrapper
00001000	8	Лямбда-коммутация
00001001	9	Волокно (физическое)
00001010	10	Резервное значение
00001011	11	Оптоволоконный канал

Таблица 10.2. Значения LSP Switching Type

Поле *Switching Type*	Десятичное значение	Тип коммутации
00000001	1	Packet-Switch Capable-1 (PSC-1) — пакетная коммутация
00000010	2	Packet-Switch Capable-2 (PSC-2)
00000011	3	Packet-Switch Capable-3 (PSC-3)
00000100	4	Packet-Switch Capable-4 (PSC-4)
00110011	51	Layer-2 Switch Capable (L2SC) — коммутация на 2-м уровне
01100100	100	Time-Division-Multiplex Capable (TDM) — временная коммутация
10010110	150	Lambda-Switch Capable (LSC) — лямбда-коммутация
11001000	200	Fiber-Switch Capable (FSC) — коммутация оптических портов

Таблица 10.3. Значения поля G-PID

Поле G-PID	Десятичное значение	Тип	Технология
0000000000000000	0	Неизвестен	Все
0000000000000001	1	Резервное значение	
0000000000000010	2	Резервное значение	
0000000000000011	3	Резервное значение	
0000000000000100	4	Резервное значение	
0000000000000101	5	Asynchronous mapping of E4	SDH
0000000000000110	6	Asynchronous mapping of DS3/T3	SDH
0000000000000111	7	Asynchronous mapping of E3	SDH
0000000000001000	8	Bit synchronous mapping of E3	SDH
0000000000001001	9	Byte synchronous mapping of E3	SDH
0000000000001010	10	Asynchronous mapping of DS2/T2	SDH
0000000000001011	11	Bit synchronous mapping of DS2/T2	SDH
0000000000001100	12	Резервное значение	SDH
0000000000001101	13	Asynchronous mapping of E1	SDH
0000000000001110	14	Byte synchronous mapping of E1	SDH
0000000000001111	15	Byte synchronous mapping of 31 * DS0	SDH
0000000000010000	16	Asynchronous mapping of DS1/T1	SDH
0000000000010001	17	Bit synchronous mapping of DS1/T1	SDH
0000000000010010	18	Byte synchronous mapping of DS1/T1	SDH
0000000000010011	19	VC-11 в VC-12	SDH
0000000000010100	20	Резервное значение	
0000000000010101	21	Резервное значение	
0000000000010110	22	DS1 SF Asynchronous	SONET
0000000000010111	23	DS1 ESF Asynchronous	SONET
0000000000011000	24	DS3 M23 Asynchronous	SONET
0000000000011001	25	DS3 C-Bit Parity Asynchronous	SONET
0000000000011010	26	VT/LOVC	SDH
0000000000011011	27	STS SPE/HOVC	SDH
0000000000011100	28	POS — без скремблирования, 16 бит CRC	SDH
0000000000011101	29	POS — без скремблирования, 32 бит CRC	SDH
0000000000011110	30	POS — со скремблированием, 16 бит CRC	SDH
0000000000011111	31	POS — со скремблированием, 32 бит CRC	SDH
0000000000100000	32	ATM mapping	SDH
0000000000100001	33	Ethernet	SDH, Lambda, оптоволокно
0000000000100010	34	SONET/SDH	Lambda, оптоволокно
0000000000100011	35	Резервное значение	Lambda, оптоволокно
0000000000100100	36	Digital Wrapper	Lambda, оптоволокно
0000000000100101	37	Lambda	Оптоволокно
0000000000100110	38	ANSI/ETSI PDH	SDH
0000000000100111	39	Резервное значение	SDH
0000000000101000	40	LAP SDH (LAPS — X.85 и X.86)	SDH
0000000000101001	41	FDDI	SDH, Lambda, оптоволокно
0000000000101010	42	DQDB (ETSI ETS 300 216)	SDH
0000000000101011	43	Оптоволоконный канал	Оптоволоконный канал
0000000000101100	44	HDLC	SDH
0000000000101101	45	Ethernet V2/DIX	SDH, Lambda, оптоволокно
0000000000101110	46	Ethernet 802.3	SDH, Lambda, оптоволокно

Информация о полосе пропускания переносится в виде 32-битового числа в формате IEEE с плавающей точкой в объектах SENDER_TSPEC и FLOWSPEC. Для не пакетных LSP полезно определить дискретные значения полосы пропускания. Некоторые характерные величины приведены в табл. 10.4.

Таблица 10.4. Кодирование полосы пропускания

Тип сигнала	Скорость (Мбит/с)	Число в формате IEEE Floating point
DS0	0.064	0x45FA0000
DS1	1.544	0x483C7A00
E1	2.048	0x487A0000
DS2	6.312	0x4940A080
E2	8.448	0x4980E800
Ethernet	10.00	0x49989680
E3	34.368	0x4A831A80
DS3	44.736	0x4AAAA780
STS-1	51.84	0x4AC5C100
Fast Ethernet	100.00	0x4B3EBC20
E4	139.264	0x4B84D000
FC-0 133M		0x4B7DAD68
OC-3/STM-1	155.264	0x4B9450C0
FC-0 266M		0x4BFDAD68
FC-0 531M		0x4C7D3356
OC-12/STM-4	622.08	0x4C9450C0
Gigabit Ethernet	1000.00	0x4CEE6B28
FC-0 1062M		0x4CFD3356
OC-48/STM-16	2488.32	0x4D9450C0
OC-192/STM-64	9953.28	0x4E9450C0
10GigE-LAN	10000.00	0x4E9502F9
OC-768/STM-256	39813.12	0x4F9450C0

10.2.2. Универсальная метка

Как уже говорилось выше, универсальная метка, в отличие от традиционной, может не только присваиваться пакетам, но и идентифицировать, например, волокно в пучке, частотный диапазон в волокне, определенную длину волны в частотном диапазоне или в волокне (в случае MPλS), или указывать временные интервалы, переносимые некоторой волной. Универсальная метка может также служить традиционной меткой MPLS, Frame Relay и ATM, причем в универсальной метке нет необходимости определять «класс», к которому она относится, т.к. он подразумевается по известным свойствам звена. Формат универсальной метки представлен на рис. 10.3.

Универсальная метка не является иерархической: когда требуются многоуровневые метки, каждый LSP должен создаваться отдельно. Если метка определяет порт или длину волны, ее поле Label содержит 32-битовое число — значение метки. Эти значения могут конфигурироваться или назначаться динамически при помощи специального протокола.

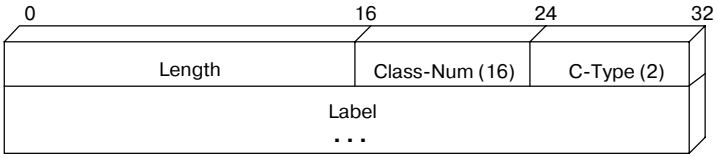


Рис. 10.3. Формат универсальной метки

10.2.3. Коммутация диапазонов волн

Коммутация диапазонов волн заслуживает отдельного параграфа, т.к. является особым случаем лямбда-коммутации. В этом случае группа волн, соседних по длине, переносится в другой диапазон волн. Для оптимизации решения этой задачи оптический коммутатор должен иметь возможность оперировать несколькими длинами волн как единым блоком. *Метка диапазона волн (waveband label)* предназначена именно для таких случаев. Между меткой диапазона волн и меткой длины волны нет существенных различий, за исключением того, что семантически диапазон волн может быть разделен на отдельные волны.

Формат метки диапазона волн представлен на рис. 10.4.

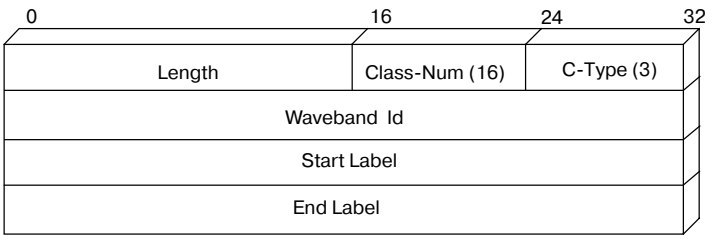


Рис. 10.4. Формат метки диапазона волн

Waveband Id — идентификатор диапазона волн. Его значение выбирается отправителем и используется во всех сообщениях, связанных с этим диапазоном.

Start Label — определяет идентификатор канала в нижней волне диапазона с точки зрения отправителя TLV.

End Label — определяет идентификатор канала в верхней волне диапазона. Идентификаторы каналов могут либо конфигурироваться, либо динамически назначаться при помощи специального протокола.

10.2.4. Предлагаемая метка

Предлагаемая метка используется верхним LSR, чтобы информировать нижний LSR о желательном значении метки, и может переноситься в сообщении *Path*. Это позволяет верхнему узлу начать конфигурацию оборудования до получения метки от нижнего узла. Такая предварительная настройка полезна для устройств, конфигурация которых требует значительного времени. Она снижает время задержки перед установлением соединения, а также при создании альтернативных LSP после возникновения сбоев в сети.

Отметим, что предлагаемая метка — это всего лишь вспомогательное средство оптимизации работы сети, поэтому если нижний узел навяжет верхнему метку, отличную от предложенной, верхний LSR будет вынужден произвести реконфигурацию в соответствии с полученной меткой. Очевидно, что верхний LSR не имеет права использовать предложенную им метку для передачи данных, пока не получит подтверждение снизу. Предлагаемая метка имеет Class-Num = 129.

10.2.5. Набор меток

Набор меток (Label Set) используется для ограничения пространства значений, из которого нижний LSR может выбрать метку для определенного LSP. Эти ограничения носят локальный характер и определяются индивидуально для каждого звена. IETF предложено использование Label Set в оптическом домене в следующих четырех случаях:

- оконечное оборудование способно работать с незначительным количеством волн или диапазонов длин волн,
- существует ряд интерфейсов, не поддерживающих преобразование длин волн и требующих использовать одну и ту же волну в некоторой последовательности узлов, или даже на всем маршруте,
- желательно ограничить количество преобразований длин волн для уменьшения искажения оптического сигнала,
- оконечные устройства звена поддерживают различные наборы длин волн.

Label Set может действовать в пределах нескольких пересылок. При этом каждый узел генерирует собственный набор меток, основываясь, например, на аппаратных возможностях оборудования, и пересылает его нижестоящему LSR.

Набор меток состоит из одного или нескольких *Label_Set_TLV*, каждый из которых содержит один или несколько элементов набора меток. Информация, переносимая в наборе меток, имеет вид, представленный на рис. 10.5.

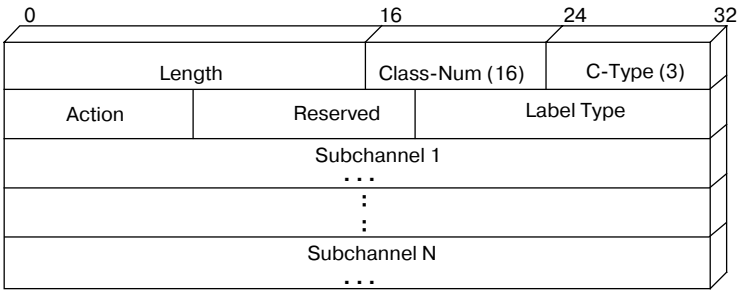


Рис. 10.5. Набор меток Label Set

Action — 8-битовое поле, определяющее содержание TLV, значения которого приведены в табл. 10.5.

Reserved — 10-битовое резервное поле, имеющее значение ноль при передаче и игнорируемое на приеме.

Label Type — тип метки. 14-битовое поле, которое определяет тип и формат метки, переносимой в TLV. Кодировка этого поля зависит от используемого протокола сигнализации. В протоколе RSVP-TE поле содержит данные о C-типе соответствующего объекта RSVP_LABEL, то есть используются только 8 младших битов поля.

Subchannel — субканал, предоставляющий метку (длина волны, волокно), пригодную для назначения. Это поле имеет формат универсальной метки.

Таблица 10.5. Значения поля Action

Поле Action	Значение
00000000	*Inclusive List* — TLV содержит один или более элементов, которые включаются в *Label Set*
00000001	*Exclusive List* — TLV содержит один или более элементов, которые исключаются из *Label Set*
00000010	*Inclusive Range* — TLV задает диапазон значений меток, которые должны быть включены в *Label Set*. В объекте содержится 2 элемента, один — определяющий начало диапазона, а другой — его конец. Нулевое значение одного из элементов означает, что соответствующая граница диапазона отсутствует
00000011	*Exclusive Range* — TLV задает диапазон значений меток, которые исключаются из *Label Set*. В объекте содержится 2 элемента, один — определяющий начало диапазона, а другой — его конец. Нулевое значение одного из элементов означает, что соответствующая граница диапазона отсутствует

10.3. Двунаправленные LSP

При рассмотрении двунаправленных LSP не всегда удобно оперировать терминами *верхний* и *нижний* LSR, поэтому в GMPLS используются другие термины: узел, начавший создание LSP, называется *инициатором*, а противоположный ему оконечный узел LSP — *терминатором.* Для одного двунаправленного LSP может существовать только один терминатор и один инициатор.

При создании двунаправленных LSP сразу выявляется ряд преимуществ используемых для этого процедур перед процедурами создания однонаправленных LSP:

- снижается время установления двусторонней связи между узлами, а также время ее восстановления,
- используется меньше служебных сообщений,
- упрощается выбор маршрута и повышается вероятность успешного установления двусторонней связи,
- предоставляется интерфейс для оборудования SDH, требующего двунаправленных связей hop-by-hop,
- удовлетворяется пожелание многих провайдеров услуг оптических сетей относительно возможности иметь двунаправленные оптические LSP.

Но, помимо преимуществ, появляются и дополнительные сложности. Для создания двунаправленного LSP необходимо назначить за один сеанс сразу две метки. Таким образом, создание двунаправленного LSP будет характеризоваться наличием в соответствующем сообщении *Path* дополнительного объекта *Upstream Label TLV*, имеющего формат универсальной метки, Class-Num = 35. Инициатор передает терминатору сообщение, в котором задает так называемую *Upstream Label*, то есть метку, которую терминатор должен использовать для адресации к инициатору, и предлагаемую метку, а терминатор должен подтвердить или отклонить использование предлагаемой метки, назначая *Downstream Label*.

Одна из главных проблем при создании двунаправленного LSP — это необходимость *разрешения конфликтов*. Например, конфликт возникает, когда узлы с обеих сторон назначили одинаковые метки практически одновременно, т.е. два запроса создать двусторонний LSP, следующие в противоположных направлениях, содержат одни и те же метки.

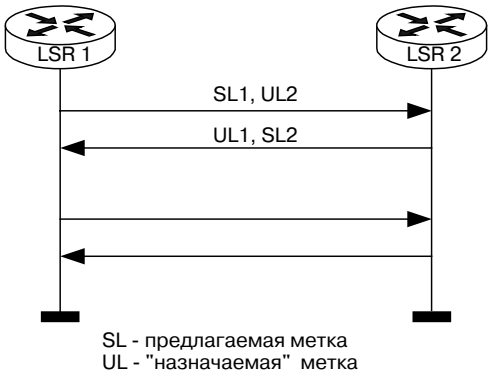


Рис. 10.6 Соперничество за метки

Если ограничений на используемые для двунаправленных LSP метки нет, и существует альтернативный ресурс меток, конфликтная ситуация разрешится простым назначением новых меток. Однако если, например, требуется, чтобы LSP физически были сгруппированы определенным образом, или если отсутствует резерв меток, то конфликт должен быть разрешен следующим способом.

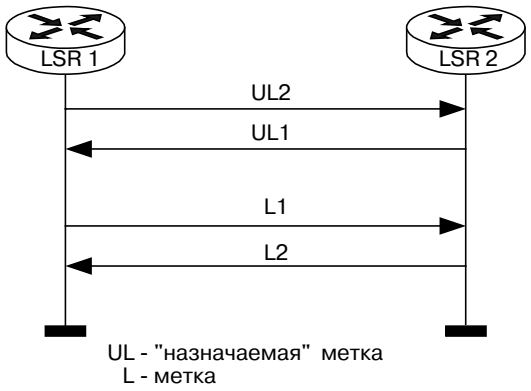


Рис. 10.7. Разрешение конфликта (ситуация без ограничений)

Узел, имеющий большее значение *идентификатора узла (node ID*), выигрывает состязание и должен передать уведомляющее сообщение *PathErr/NOTIFICATION*, содержащее информацию «проблема маршрутизации/сбой при назначении метки». По получении такого сообщения второй узел должен попробовать назначить другую метку. Если это невозможно, запускается стандартный механизм обработки ошибок.

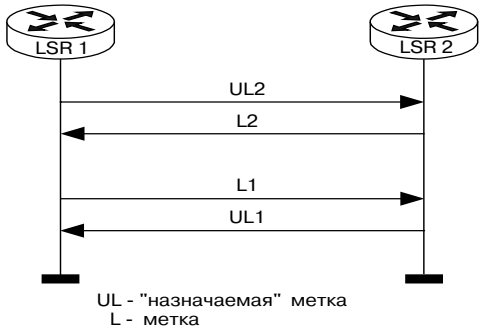


Рис. 10.8 Разрешение конфликта (ситуация наличия ограничений)

Чтобы уменьшить вероятность возникновения конфликтной ситуации, можно внедрить стратегию, требующую, чтобы узел с меньшим ID никогда сам не назначал предлагаемых меток, а всегда принимал их от узла с большим ID.

Здесь же следует упомянуть еще об одном нововведении GMPLS, связанном с обработкой ошибок, которые обусловлены невозможностью использовать метку. Как и в MPLS, в таком случае будет передано сообщение об ошибке — «неприемлемое значение метки»,— но в GMPLS вместе с ним будет отправлен объект *Acceptable Label Set* — набор приемлемых меток, имеющий тот же формат, что и *Label Set*, но Class-Num = 130. Этот объект может переноситься сообщениями *PathErr* и *ResvErr.*

Для RSVP-TE существует два дополнительных положения. Во-первых, за node ID принимается IP-адрес, указываемый в объекте RSVP_HOP. А во-вторых, при передаче инициирующего сообщения *Path* идентификатор node ID соседнего маршрутизатора может быть неизвестен, и в этом случае узлу предлагается случайная метка, выбранная из доступного пространства меток.

10.4. Уведомления и сообщения об ошибках

10.4.1. Запрос уведомления

В GMPLS уведомления могут запрашиваться как верхним, так и нижним маршрутизатором. Для запроса уведомлений служит объект *Notify Request*. Он может переноситься в сообщениях *Path* или *Resv* и имеет значение Class-Num = 195 и значение C-типа, зависящее от версии протокола IP. Объект *Notify Request* содержит адрес узла, которому следует передать уведомление в случае неисправности в LSP. Формат объекта IPv4 Notify Request, C-тип = 1 представлен на рис. 10.9, а формат объекта IPv6 Notify Request, C-тип = 2 — на рис. 10.10.

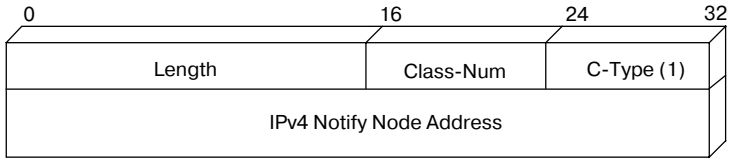


Рис. 10.9. Формат IPv4 Notify Request Object

IPv4 Notify Node Address — 32-битовый IPv4 адрес узла, которому необходимо передать уведомление.

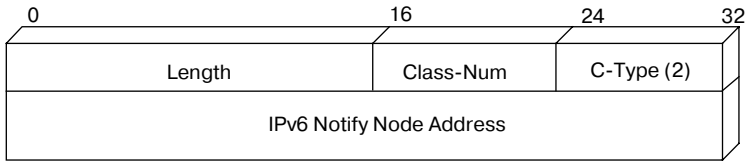


Рис. 10.10. Формат IPv6 Notify Request Object

IPv6 Notify Node Address — 128-битовый IPv6 адрес узла, которому необходимо передать уведомление.

Если сообщение содержит несколько объектов Notify_Request, только первый из них является значащим, а остальные могут игнорироваться.

10.4.2. Уведомляющее сообщение

Уведомляющее сообщение — это механизм, предназначенный для предоставления информации о состоянии LSP несмежным узлам. Отличием уведомляющего сообщения от уже существующих сообщений об ошибках (*PathErr, ResvErr*) является то, что оно «адресуемо» и может передаваться узлу, не связанному с отправителем непосредственно.

Одно уведомляющее сообщение может содержать информацию, относящуюся к нескольким сеансам, в которых отправителем может быть как верхний, так и как нижний маршрутизатор. Генерируя уведомляющее сообщение, узел должен постараться скомбинировать в одном сообщении всю информацию, предназначенную уведомляемому узлу и связанную с определенным значением ERROR_SPEC, как это показано на рис. 10.11. Объект ERROR_SPEC содержит код ошибки и IP-адрес узла, обнаружившего ошибку, или отказавшего звена.

Узел, получивший уведомляющее сообщение, должен ответить соответствующим подтверждающим сообщением. Уведомляющее сообщение имеет Message Type = 21.

<***Notify message***>	::= <Common Header> [<INTEGRITY>] [[<MESSAGE_ID_ACK> | <MESSAGE_ID_NACK>] ...] [<MESSAGE_ID>] <ERROR_SPEC> <notify session list>
<***notify session list***>	::= [<notify session list>] <upstream notify session> <downstream notify session>
<***upstream notify session***>	::= <SESSION> [<ADMIN_STATUS>] [<POLICY_DATA>...] <sender descriptor>
<***downstream notify session***>	::= <SESSION> [<POLICY_DATA>...] <flow descriptor list>

Рис. 10.11. Уведомляющее сообщение

10.4.3. Разрушение процесса пересылки сообщением об ошибке PathErr

Технологией MPLS в некоторых аварийных ситуациях предусматривается пересылка сообщения *PathErr* к отправителю сообщения *Path*. Промежуточные узлы, через которые проходит *PathErr*, могут

его анализировать, но не должны предпринимать никаких действий. В случае, когда маршрутизация в сети ведется на базе стандартных протоколов IGP, и маршрут может динамически изменяться, такое решение представляется действительно удачным. Но это выглядит несколько иначе, когда RSVP начинает работать с явно заданными маршрутами: часто ошибки могут быть исправлены только в узле-источнике или в другом верхнем маршрутизаторе. Для того чтобы освободить ресурсы, узел-источник должен получить сообщение *PathErr* и затем либо передать сообщение *PathTear*, либо ждать срабатывания таймера. Это приводит к тому, что ресурсы на маршруте остаются резервированными и не используемыми дольше, чем необходимо, а также к увеличению служебной нагрузки.

В GMPLS эта проблема решена — промежуточным узлам предоставлена возможность самостоятельно разрушать процесс пересылки при возникновении ошибок некоторых типов. Для этого определен новый флаг в объекте ERROR_SPEC, описанном в базовой версии RSVP (точнее, существует два объекта ERROR_SPEC — для IPv4 и для IPv6). К двум уже существующим флагам (0x01 — *InPlace* и 0x02 — *NotGuilty*), переносимым в однобайтовом поле *Flags*, добавлен флаг 0x04 *Path_State_Removed*. Этот флаг означает, что промежуточный узел, пересылающий сообщение об ошибке, удалил соответствующее *PathErr* состояние процесса пересылки.

По умолчанию флаг *Path_State_Removed* имеет нулевое значение, как при отправлении, так и при транзитной пересылке. Флаг может быть установлен узлом, обнаружившим ошибку, в результате которой процесс пересылки в нем был нарушен. Если узел не является для данного сеанса оконечным, ему следует создать и передать соответствующее сообщение *PathTear*. Узел, получивший сообщение *PathErr* с установленным в объекте ERROR_SPEC флагом *Path_State_Removed*, может удалить у себя сведения о состоянии процесса.

10.5. Метки для явно заданного маршрута

В традиционной MPLS с функциями инжиниринга трафика существует возможность задавать маршрут, по которому будет проходить LSP. Это делается при помощи объектов ERO в RSVP-TE или объектов ER-Hop в CR-LDP. Но существует ряд ситуаций, когда объект, задающий явный маршрут, не содержит достаточно информации для того, чтобы обеспечить управление созданием LSP. Такой ситуацией будет случай, когда инициатор хочет выбрать метку, используемую на маршруте, но ни ERO, ни ER-Hop не содержат подобъекта, задающего метку. В GMPLS вводится такой подобъект, формат которого будет зависеть от используемого протокола сигнализации, но всегда иметь одни и те же поля:

L — бит, имеющий значение 0,

U — бит индикации направления метки, который имеет нулевое значение для присвоения *Downstream Label*, а единичное значение используется только при создании двунаправленного LSP и означает присвоение *Upstream Label,*

Label — поле имеет переменную длину, и его формат соответствует формату универсальной метки.

Для RSVP это будет подобъект Label ERO, который имеет вид, приведенный на рис. 10.12.

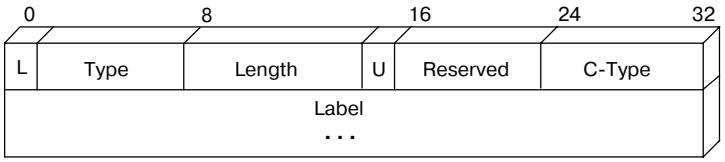


Рис. 10.12. Формат подобъекта Label ERO

В дополнение к вышеописанным, *Label ERO* содержит следующие поля.

Type=3 означает тип подобъекта — «метка».

Length — общая длина подобъекта в байтах, которая должна быть кратна 4.

C-Type — C-тип, копируемый из объекта LABEL.

10.6. Информация о защите

Важность такого аспекта, как защищенность, обсуждалась в главе 8, где речь шла о построении виртуальных частных сетей на основе MPLS-технологии. Все это справедливо и в отношении защиты в GMPLS.

Информация о защите переносится в GMPLS новым объектом *Protection* (рис. 10.13). Использование информации о защите является для каждого LSP опциональным. Информация о защите отражает требуемый в звене тип защиты создаваемого LSP. Если запрошен конкретный тип защиты, например «1+1» или «1:N», то запрос создания LSP обрабатывается только в случае, если защита такого типа может быть обеспечена. Характеристики защиты звена могут объявляться в процессе работы протоколов маршрутизации и использоваться при вычислении маршрута для LSP.

Информация о защите определяет также, является ли LSP основным или резервным. Ресурсы резервного LSP не используются до тех пор, пока основной LSP не выйдет из строя, и могут заниматься на это время другим LSP.

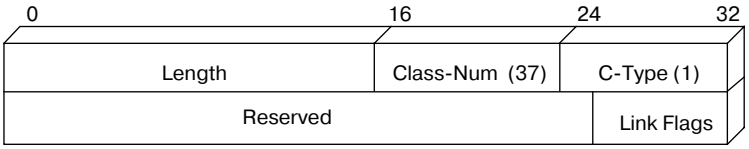


Рис. 10.13. Формат объекта Protection

Secondary (S) — значение этого бита, равное единице, указывает, что запрашиваемый LSP является вторичным.

Reserved — 25-битовое поле, имеющее значение ноль на передаче и игнорируемое на приеме.

Link Flags — 6 битов, определяющих желательный тип защиты звена; возможные значения типов защиты приведены в табл. 10.6.

Таблица 10.6. Типы защиты

Поле *Link Flags*	Тип защиты
000000	Может использоваться защита любого типа, или она может отсутствовать
000001	*Extra Traffic* — LSP должен использовать звенья, используемые для защиты других звеньев (по которым проходит первичный трафик). Эти звенья должны быть освобождены в случае выхода из строя защищаемого звена. Таким образом это — самый низкий уровень защиты звена
000010	*Unprotected* — не должно быть защиты звена
000100	*Shared* — требуется защита путем резервирования
001000	*Dedicated 1:1* — требуется защита типа 1:1
001010	*Dedicated N+1* — требуется защита типа N+1
010100	*Enchased* — требуется защита, более надежная, чем дает схема N+1

10.7. Информация об административном статусе

Информация об административном статусе (*Administrative Status Information*) в GMPLS тоже переносится в новом объекте Admin_Status (рис. 10.14). Эта информация служит для двух целей.

Во-первых, она отображает состояние LSP, то есть, находится ли он в рабочем, нерабочем или тестовом состоянии, или же происходит разрушение (удаление) LSP. Каждый узел может обрабатывать такого рода информацию по своему усмотрению, а точнее, в соответствии с настройкой, произведенной администратором сети.

Во-вторых, объект информации об административном статусе может содержать запрос установить определенный статус LSP. Другие варианты возможного применения зависят от используемого в сети протокола сигнализации.

Отметим, что использование этого объекта для каждого LSP является опциональным.

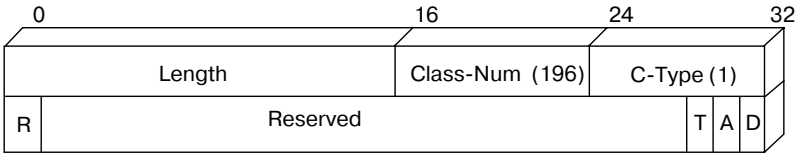


Рис. 10.14. Формат объекта Admin_Status

Reflect (R) — если значение этого бита равно 1, то пограничный узел не должен отсылать объект обратно в соответствующем сообщении. В запросах изменения статуса значение бита R всегда равно 0.

Reserved — 28 резервных битов; при передаче имеют значение ноль и игнорируются на приеме.

Testing (T) –когда запрашивается тестирование LSP, этому биту присваивается значение 1.

Administratively down (A) –когда запрашивается перевод LSP в *нерабочее* состояние, этому биту присваивается значение 1.

Deletion in progress (D) –этому биту присваивается значение 1, если запрашивается разрушение LSP.

10.8. Разделение общего тракта управления

Узлы оптической сети, на которую в первую очередь ориентирована технология GMPLS, могут соединяться сотнями волокон, каждое из которых переносит сотни и тысячи длин волн. Чтобы уменьшить базы данных и улучшить масштабируемость сети, в GMPLS предложено объединять отдельные *тракты, переносящие данные пользователей* (в дальнейшем, сокращенно,— *несущие тракты*), таким образом, чтобы затем представлять их протоколам маршрутизации как единый несущий тракт. Для такого объединенного несущего тракта понадобится только один общий служебный тракт управления.

В этом параграфе мы рассмотрим два момента: идентификацию несущих трактов, к которым относится служебная информация, переносимая по общему тракту управления, а также обработку ошибок в тракте управления, не влияющих на несущие тракты.

10.8.1. Идентификация интерфейсов

В отличие от традиционной MPLS, где существует однозначное соответствие между несущим трактом и служебным трактом, в GMPLS приходится передавать по тракту управления, в дополнение к управляющей информации, еще и информацию, идентифицирующую тот несущий тракт, к которому та относится.

GMPLS поддерживает явную идентификацию несущих трактов, используя разные способы идентификации интерфейсов, такие как адреса IPv4 и IPv6, индексы интерфейсов объединенного несущего тракта и индексы интерфейсов отдельных несущих трактов, из которых составлен объединенный тракт. Эти отдельные тракты удобно называть компонентами объединенного тракта. Индексы назначаются либо конфигурацией, либо с использованием специального протокола. Интерфейс всегда выбирает отправитель сообщения *Path.*

Идентификатор интерфейсов (*Interface_ID*) представляет собой набор объектов TLV (рис. 10.15).

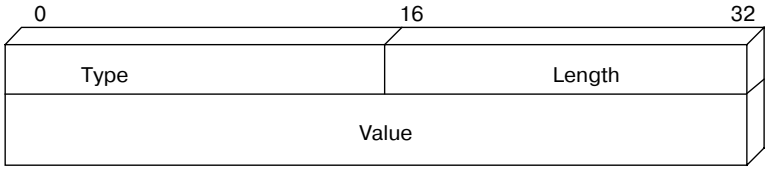


Рис. 10.15. Формат TLV-объекта в составе Interface_ID

Length — 16-битовое поле, отражающее общую длину TLV в байтах, которая обязательно должна быть кратна 4; в противном случае поле *Value* дополняется нулями.

Type — 16-битовое поле, определяющее тип идентифицируемого интерфейса.

Таблица 10.7. Типы интерфейсов, определенные IETF

Поле *Type*	Длина	Формат	Описание
1	8	IPv4	Адрес IPv4
2	20	IPv6	Адрес IPv6
3	12	См.ниже.	Индекс интерфейса объединенного тракта (IF_INDEX)
4	12	См.ниже.	Интерфейс компонента объединенного тракта (нижний) (COMPONENT_IF_DOWNSTREAM)
5	12	См.ниже.	Интерфейс компонента объединенного тракта (верхний) (COMPONENT_IF_UPSTREAM)

Для интерфейсов типов №3, 4 и 5 используется следующий формат поля *Value*:

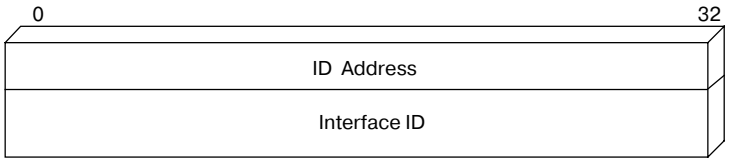


Рис. 10.16. Формат поля Value для интерфейсов типа 3, 4 и 5

IP Address — поле может содержать либо IP-адрес звена, либо IP-адрес узла, переносимый в адресном TLV.

Interface ID — для типа 3 в этом поле переносится идентификатор интерфейса. Для типов 4 и 5 *Interface ID* идентифицирует интерфейс компонента объединенного несущего тракта. Особое число 0xFFFFFFFF означает, что для всех компонентов объединенного тракта действует единая метка.

Для переноса вышеописанной информации используются новые подтипы объекта RSVP_HOP, формат которых зависит от версии действующего протокола IP.

Формат объекта IPv4 IF_ID RSVP_HOP представлен на рис. 10.17:

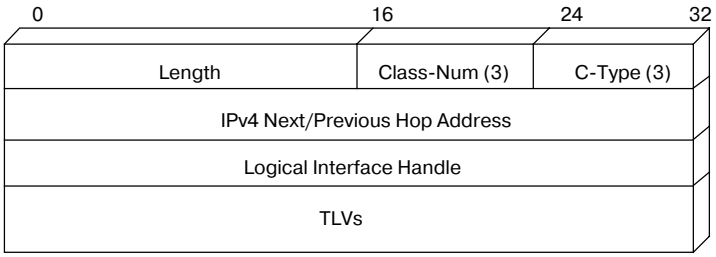


Рис. 10.17. Формат IPv4 IF_ID RSVP_HOP Object

Формат объекта IPv6 IF_ID RSVP_HOP показан на рис. 10.18.

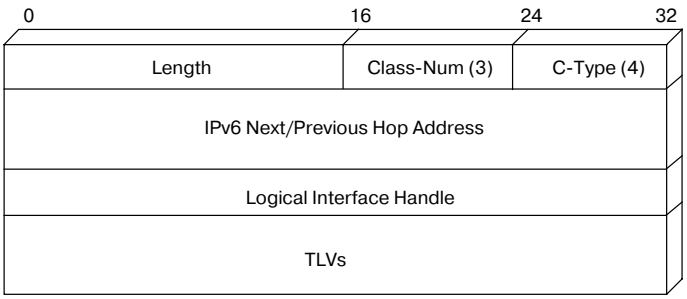


Рис. 10.18. Формат IPv6 IF_ID RSVP_HOP Object

Узлы, которым при передаче уведомлений об ошибках *PathErr* или *ResvErr* нужно указать интерфейс, с которым связана ошибка, могут использовать объект IF_ID ERROR_SPEC, соответствующий используемой версии протокола IP.

10.8.2. Обработка ошибок

В соответствии с концепцией GMPLS тракт управления стал независимым, что привело к возможности ошибок двух новых типов. Первый тип — это неисправность, не позволяющая двум соседним узлам обмениваться служебными сообщениями в течение неко-

торого времени. Как только связь будет восстановлена, протокол сигнализации должен оповестить об этом узлы. Он должен также убедиться, что процессы на узлах синхронизированы друг с другом. Такие ошибки называются ошибками тракта управления.

Второй тип –отказ управления и перезапуск узла с потерей большинства сведений о состоянии пересылки (при этом данные пересылки сохраняются). В данном случае верхний или нижний узел (в зависимости от того, где случилась неисправность) должен синхронизировать свое состояние с состоянием перезапущенного узла. Для того чтобы можно было произвести ресинхронизацию, узел, подвергающийся перезагрузке, должен сохранить привязку всех входящих и исходящих меток. Такие ошибки называются ошибками узла.

Механизмы обработки перечисленных отказов зависят от протокола. Для поддержки обработки ошибок обоих типов в RSVP добавлен объект *Restart_Cap* (рис. 10.19), переносимый в сообщениях Hello.

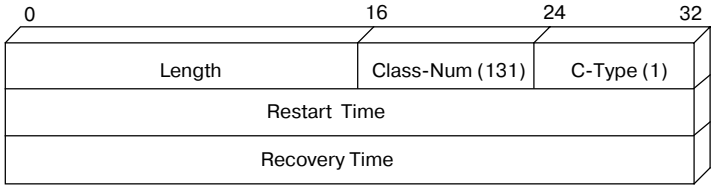


Рис. 10.19. Формат Restart_Cap Object

Restart Time — (32 бита) суммарное время в миллисекундах, которое требуется отправителю объекта *Restart_Cap* для перезапуска своих компонентов RSVP-TE и тракта, используемого сигнализацией RSVP. Значение 0xffffffff указывает, что для перезапуска управления требуется неопределенное время, и что неисправности управления не влияют на сами пересылки.

Recovery Time — (32 бита) рекомендуемый отправителем период времени в миллисекундах, в течение которого получателю нужно повторно синхронизировать с отправителем процессы MPLS и RSVP после восстановления. Нулевое значение поля указывает, что при перезагрузке текущие состояния пересылок не были сохранены. В случае ошибок тракта управления узел А должен обновить все состояния пересылок, связанные с соседним узлом В, причем обновление всех их должно быть произведено до истечения *Recovery Time*, объявленного узлом В.

Процедуры восстановления после ошибок узла потребовали специфицировать новый объект — *Recovery Label,* — который имеет формат универсальной метки, Class-Num = 34 и С-тип предлагаемой метки. Описание этих процедур приведено в RFC 3473.

10.9. Форматы сообщений

В этом параграфе приведены форматы сообщений, ранее уже специфицированных для RSVP, но претерпевших в технологии GMPLS некоторые изменения. Следует заметить, что появление двунаправленных LSP повлекло за собой и различие в форматах сообщений, относящихся к двунаправленным и к однонаправленным LSP.

Формат сообщения *Path*:

```
<Path Message> ::=       <Common Header> [ <INTEGRITY> ]
                                         [ [<MESSAGE_ID_ACK> | <MESSAGE_ID_NACK>] ... ]
                                         [ <MESSAGE_ID> ]
                                        <SESSION> <RSVP_HOP>
                                        <TIME_VALUES>
                                          [ <EXPLICIT_ROUTE> ]
                                        <LABEL_REQUEST>
                                          [ <PROTECTION> ]
                                          [ <LABEL_SET> ... ]
                                          [ <SESSION_ATTRIBUTE> ]
                                          [ <NOTIFY_REQUEST> ]
                                          [ <ADMIN_STATUS> ]
                                          [ <POLICY_DATA> ... ]
                                         <sender descriptor>
```

Формат <sender descriptor> для однонаправленных LSP:

```
<sender descriptor> ::=  <SENDER_TEMPLATE> <SENDER_TSPEC>
                                          [ <ADSPEC> ]
                                          [ <RECORD_ROUTE> ]
                                          [ <SUGGESTED_LABEL> ]
                                          [ <RECOVERY_LABEL> ]
```

Формат <sender descriptor> для двунаправленных LSP:

```
<sender descriptor> ::=  <SENDER_TEMPLATE> <SENDER_TSPEC>
                                          [ <ADSPEC> ]
                                          [ <RECORD_ROUTE> ]
                                          [ <SUGGESTED_LABEL> ]
                                          [ <RECOVERY_LABEL> ]
                                         <UPSTREAM_LABEL>
```

Формат сообщения *PathErr*:

```
<PathErr Message> ::=    <Common Header> [ <INTEGRITY> ]
                                          [ [<MESSAGE_ID_ACK> | <MESSAGE_ID_NACK>] ... ]
                                          [ <MESSAGE_ID> ]
                                         <SESSION> <ERROR_SPEC>
                                          [ <ACCEPTABLE_LABEL_SET> ... ]
                                          [ <POLICY_DATA> ... ]
                                         <sender descriptor>
```

Формат сообщения *Resv*:

```
<Resv Message> ::=      <Common Header> [ <INTEGRITY> ]
                                        [ [<MESSAGE_ID_ACK> | <MESSAGE_ID_NACK>] ... ]
                                        [ <MESSAGE_ID> ]
                                       <SESSION> <RSVP_HOP>
                                       <TIME_VALUES>
                                        [ <RESV_CONFIRM> ]  [ <SCOPE> ]
                                        [ <NOTIFY_REQUEST> ]
                                        [ <ADMIN_STATUS> ]
                                        [ <POLICY_DATA> ... ]
                                       <STYLE> <flow descriptor list>
```

Формат сообщения *ResvErr*:

```
<ResvErr Message> ::=   <Common Header> [ <INTEGRITY> ]
                                          [ [<MESSAGE_ID_ACK> | <MESSAGE_ID_NACK>] ... ]
                                          [ <MESSAGE_ID> ]
                                         <SESSION> <RSVP_HOP>
                                         <ERROR_SPEC> [ <SCOPE> ]
                                          [ <ACCEPTABLE_LABEL_SET> ... ]
                                          [ <POLICY_DATA> ... ]
                                         <STYLE> <error flow descriptor>
```

Формат сообщения *Hello*:

```
<Hello Message> ::= <Common Header> [ <INTEGRITY> ] <HELLO>
                                    [ <RESTART_CAP> ]
```

10.10. Что дальше?

Появление технологии Generalized MPLS уже отчасти дает ответ на поставленный в названии параграфа вопрос. Принципы GMPLS позволяют не задумываться о сегодняшних и будущих способах передачи информации и средах передачи. Остается лишь следить за обновлениями версий протоколов GMPLS.

Однако исследования рабочей группы IETF по MPLS не прекращаются. Более того, к ним добавились усилия двух посвященных построению оптических сетей форумов MPLS, материалы которых упоминались в этой главе,— Форума по оптическому межсетевому взаимодействию OIF и Форума по взаимосвязи услуг оптических доменов ODSI.

Главная функция OIF — специфицировать протоколы и другие технические условия, которые необходимы для реализации открытого и стандартизованного интерфейса O-UNI для оптических сетей. Форум ODSI занимается разработкой стандартного интерфейса пользователь-сеть и вопросами поддержки протоколов оптического оборудования. Он начал свою деятельность в январе 2000 года и разработал ряд спецификаций. Были закончены документы MIB — определения управляемых объектов для системы управления ODSI, а также спецификации управления доступом и учета стоимости. Форум ODSI также провел тестирование своих протоколов на предмет взаимодействия различных IP-устройств и оптических устройств.

В составе *Форума ATM (ATM Forum)* тоже имеются две рабочие группы, которые выполняют работы, включающие в себя разные аспекты MPLS:

- Рабочая группа по управлению трафиком (Traffic Management WG);
- Рабочая группа по совместной работе ATM и IP (ATM-IP Collaboration WG).

Кроме того, в состав *Сектора стандартизации электросвязи Международного союза электросвязи ITU-T* в настоящее время входят три исследовательские комиссии (Study Groups), занимающиеся вопросами MPLS:

- Исследовательская комиссия 11 (SG11): MPLS Signaling (Сигнализация в сети MPLS),
- Исследовательская комиссия 13 (SG13): MPLS Network Architecture (Архитектура сети MPLS),
- Исследовательская комиссия 15 (SG15): MPLS and IP Equipment Requirements (требования к оборудованию MPLS и IP).

Так что вопрос, озаглавивший этот параграф, пока остается открытым. Что ждет MPLS? Какие новые перспективы могут открыться благодаря этой технологии? Над чем сейчас трудятся разработчики сетевого оборудования и средств тестирования MPLS?

На эти вопросы мы постараемся ответить в следующей, 11-й главе, которая так и называется — «Камо Грядеши?».

Глава 11
Камо Грядеши?

11.1. Внедрение и перспективы MPLS

Разнообразие рассматриваемых в заключительной главе вопросов обусловлено характером этой развивающейся, динамичной и популярной технологии. MPLS создавалась, главным образом, как технология для ядра сети, как средство, позволяющее сетевым устройствам, которым приходится справляться с огромными потоками информации, обрабатывать эти потоки более эффективно. Но в последнее время наблюдается некоторый конфликт между уже утвержденными и зафиксированными аспектами технологии MPLS, обсуждавшимися в предыдущих главах книги, и разнообразными предлагаемыми направлениями ее совершенствования, развития и расширения.

Обсуждается вопрос о том, удастся ли MPLS проникнуть в локальные сети и в клиентское и серверное программное обеспечение TCP/IP. Такого рода расширение MPLS не представляет сложности с технической точки зрения, так как весьма вероятно, что программное обеспечение IP для компьютеров будет поддерживать протокол RSVP, который MPLS может использует для создания LSP.

В отношении глобальных сетей вопрос заключается в том, обеспечит ли универсальная способность MPLS к транспортировке информации с любыми разумными требованиями к QoS и в любых форматах протоколов вытеснение ею других сетевых технологий коммутации, включая ATM. Все эти вопросы затрагиваются в заключительной главе книги.

В начале главы кратко описывается проблематика эксплуатационного управления, баз управляющей информации MIB, тестирования, принципов и вариантов реализации MPLS. Этим ключевым для сегодняшнего внедрения и будущего развития MPLS вопросам посвящены три следующих параграфа главы.

Далее в главе рассматриваются аспекты MPLS, относящиеся как к принципиально новым ее перспективам, так и к дальнейшим расширениям уже описанных составляющих MPLS. Это и DiffServ-aware

для более эффективного инжиниринга трафика, и оценка возможностей непосредственной передачи речи поверх MPLS, и будущая тотальная передача «всего» поверх MPLS. Последнему вопросу — обсуждению концепции AToM (Any Transport over MPLS), т.е. любого транспорта поверх MPLS, — посвящен отдельный и весьма важный параграф главы.

В изложении этих материалов основной упор сделан на разрабатываемые документы и концепции перспективных направлений технологии MPLS,. На момент написания книги в рабочей группе MPLS комитета IETF находилось на рассмотрении порядка 20 *draft*-проектов документов RFC. Кроме того, связанные с MPLS интересные документы выпускает рабочая группа по Traffic Engineering.

Еще одной ярко выраженной тенденцией развития современных телекоммуникаций является увеличение доли оптических сетей. В предыдущей главе книги уже описано, каким образом технология MPLS отреагировала на это явление — MPLamdaS (многопротокольная лямбда-коммутация) и практически включившая ее в себя технология GMPLS.

На современном рынке телекоммуникаций как новые, так и существующие услуги требуют, как правило, наличия точно оговоренных соглашений об уровнях обслуживания (SLA). Необходимость соблюдать SLA заставляет уделять самое пристальное внимание вопросам обеспечения гарантированного QoS. Не стала исключением и технология MPLS. Проблематика SLA и QoS рассматривается в заключительных параграфах главы.

11.2. MIB и MPLS

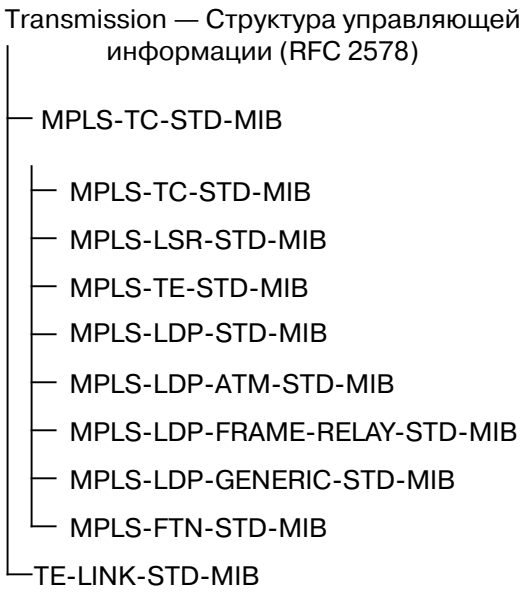
То, что MPLS является сегодня наиболее перспективной транспортной технологией в конвергентных сетях и, следовательно, вынуждена работать с разнотипным трафиком, различными видами сетевой политики и сложными сетевыми топологиями, побуждает разработчиков сосредоточить усилия на повышении ее управляемости. Прежде всего, это делается путем спецификации баз управляющей информации MIB и различных управляемых объектов.

Управление MPLS — это мониторинг и управление приложениями, услугами и соответствующими ресурсами в домене MPLS. Применительно к MPLS широко используется стандартизованная система управления *SNMP (Simple Network Management Protocol)*. Ключевой аспект SNMP состоит во взаимодействии с управляемыми объектами через виртуальную базу данных, которая называется MIB.

База управляющей информации MIB (Management Information Base) содержит изменяемые характеристики управляемых объектов, доступные для SNMP. В виду относительной простоты их использования и высокой эффективности управления, которое они обеспечивают, базы MIB весьма оперативно внедрились в технологию MPLS. На момент написания книги некоторые базы MIB были уже специфицированы комитетом IETF, некоторые же находились в состоянии *draft*-проектов.

Логически MIB можно изобразить в виде абстрактного дерева, листьями которого являются отдельные информационные элементы. Идентификаторы (ID) объектов имеют иерархическую структуру и уникальным образом идентифицируют объекты MIB этого дерева. Идентификаторы объектов высшего уровня MIB назначаются Международной Электротехнической Комиссией ISO (ISO IEC). Идентификаторы объектов низшего уровня назначаются относящимися к ним организациями. В случае MPLS идентификаторы определяются IANA.

Структура MIB для MPLS имеет следующий вид:



Модуль *MPLS-TC-STD-MIB* определяет текстовые условные обозначения, которые могут быть общими для родственных MIB-модулей. Эти условные обозначения позволяют нескольким MIB-модулям использовать единый синтаксис и формат для дублирующейся информации. Например, метки — центральное понятие MPLS — присутствуют во многих MIB-модулях. Поэтому условные обозначения для представления MPLS-меток определены в модуле MPLS-TC-STD-MIB, а все остальные MIB-модули импортируют из него нужные им сведения.

Модуль *MPLS-LSR-STD-MIB* описывает управляемые объекты для моделирования MPLS LSR. Этот MIB-модуль используется для моделирования и управления базовым поведением коммутации по меткам. Он представляет *базу LFIB (Label Forwarding Information Base)* и предоставляет сведения обо всех LSP, проходящих через данный LSR. Поскольку коммутация по меткам является общей для всех MPLS-приложений, то к этому модулю обращаются другие MPLS MIB-модули. В общем, MPLS-LSR-STD-MIB предоставляет модель привязки входных меток к выходным меткам посредством концептуального объекта *MPLS cross-connect.* Компоненты MPLS cross-connect и их характеристики содержатся в MPLS-LSR-STD-MIB и обычно используются другими MIB-модулями для обращения к LSP. Например, MPLS-TE-STD-MIB моделирует туннели, обсуждавшиеся в главе 8.

Модуль *MPLS-LDP-STD-MIB* описывает управляемые объекты, используемые для того, чтобы моделировать и управлять протоколом LDP. Этот модуль содержит объекты, общие для всех реализаций LDP. Для реализации LDP, поддерживающей стандартные MIB, модуль предоставляет основной набор объектов, которые будут использоваться вместе с одним или несколькими дополнительными LDP MIB-модулями.

Модуль *MPLS-LDP-GENERIC-STD-MIB* предоставляет объекты для управления пространством меток на платформной основе и обычно реализуется вместе с MPLS-LDP-STD-MIB. Этот модуль содержит таблицы для конфигурации диапазонов общих меток. Хотя спецификации LDP не определяют возможности конфигурировать диапазоны для общих меток, MIB-модуль может резервировать диапазоны меток. По крайней мере, рабочей группе, придумавшей этот MIB-модуль, такая опция показалась полезной.

Модуль *MPLS-LDP-ATM-STD-MIB* используется совместно с MPLS-LDP-STD-MIB в реализациях, использующих в качестве транспорта ATM. Соответственно, таблицы в этом модуле позволяют конфигурировать LDP для работы с ATM.

Модуль *MPLS-LDP-FRAME-RELAY-STD-MIB* аналогичен предыдущему, только в качестве транспорта в нем выступает Frame Relay.

Модуль *MPLS-TE-STD-MIB* описывает управляемые объекты для моделирования и управления TE-туннелями. Этот модуль MIB основывается на таблице, содержащий данные о TE-туннелях, которые либо начинаются в рассматриваемом LSR, либо проходят через этот LSR, либо заканчиваются в нем. MIB-модуль содержит объекты конфигурации и статистики, необходимые для TE-туннелей. Напомним, *что LSP-туннелям* был посвящен параграф 8.2 главы о VPN, а организуемые методами инжиниринга трафика *TE-туннели* обсуждались в главе 9.

Модуль *MPLS-FTN-STD-MIB* описывает управляемые объекты, предназначенные для моделирования и управления каждой привязкой FEC-to-NHLFE, являющейся элементом таблиц коммутации по меткам в пограничных маршрутизаторах LSR.

Модуль *TE-LINK-STD-MIB* описывает управляемые объекты, используемые для моделирования и управления TE-соединениями, включая объединенные звенья. Функция *TE-соединений (TE-link)* создана для агрегации одного или более схожих потоков трафика или TE-звеньев между парой LSR.

Помимо MIB, специфицированных MPLS WG, существуют MIB других рабочих групп. К ним относится модуль *PPVPN-MPLS-VPN-STD-MIB*, который описывает управляемые объекты, используемые для моделирования и управления MPLS-VPN. Модуль содержит таблицы, моделирующие компоненты VRF и соответствующие им интерфейсы.

TE-MIB разработан TE WG и содержит объекты для мониторинга TE-туннелей на их входе. Во многом этот объект дублирует информацию, содержащуюся в MPLS-TE-STD-MIB, но TE-MIB более прост и не так многофункционален.

Рабочая группа CCAMP разрабатывает MIB-модули, призванные обеспечить поддержку GMPLS. Они будут непосредственно взаимодействовать с MPLS MIB модулями. Все вышеупомянутые MIB-модули содержат информационные таблицы, определяют несколько переменных и имеют набор сообщений, которые они могут генерировать.

11.3. Оборудование MPLS

Как отмечалось в первой главе книги, LSR фактически представляет собой IP-маршрутизатор с поддержкой MPLS. Эта поддержка MPLS, превращающая обыкновенный маршрутизатор в LSR, предусматривает умение читать как метки, так и заголовки протокола сетевого уровня, отличать на входе пакеты, снабженные меткой, от обычных IP-пакетов, выполнять с такими пакетами с меткой процедуру *Label Swapping* и отправлять их на соответствующий порт согласно имеющейся в LSR таблице коммутации по меткам.

Все эти функциональные возможности современных LSR поддерживаются установленным на них программным обеспечением. Примером может служить программное обеспечение Cisco IOS Software, имеющее несколько версий. Каждая версия имеет несколько наборов функций *Feature Set*, которые используются в зависимости от применения маршрутизатора. Позже, благодаря модульности программного обеспечения, к тому или иному *Feature Set* можно добавлять дополнительные функции. Вообще говоря, Cisco IOS Soft-

ware предназначено для установки на обычные IP-маршрутизаторы Cisco, а начиная с версии 12.0, эти программные пакеты Cisco IOS Software включают в себя поддержку MPLS. В оборудовании Cisco функции MPLS, в той или иной степени, сегодня поддерживают маршрутизаторы старших серий, начиная с 3600, а также маршрутизатор 2691 из серии 2600 и некоторые коммутаторы Cisco Catalyst.

Наряду с программной поддержкой функций MPLS, часть производителей оборудования выпускает специализированные устройства с аппаратной поддержкой MPLS. Иллюстрацией может служить оборудование компании Juniper Networks, которая имеет свой собственный программный пакет JUNOS, аналогичный Cisco IOS Software, но также реализует в своем оборудовании и аппаратную поддержку MPLS.

Для такой аппаратной поддержки применяются различные решения, например, использование *программируемых логических матриц FPGA (Field-Programmable Gate Arrays)*, как это делает компания Marvell, или *специализированных интегральных схем ASIC (Application Specific Integrated Circuit)*, нашедших свое место в оборудовании Riverstone Networks.

Аппаратная поддержка имеет своих сторонников и противников, положительным моментом будет упрощение и ускорение аппаратно реализуемых функций, однако в дальнейшем оператору может потребоваться аппаратная модернизация, которая, несомненно, обойдется ему заметно дороже, чем обновление программного обеспечения.

Среди операторских компаний первой объявила о создании VPN на базе MPLS компания AT&T в январе 1999 г. Вскоре появились и решения от других провайдеров, предлагающие конечным пользователям гибкость IP и аналог виртуальных каналов.

В России одной из крупнейших IP/MPLS сетей обладает компания ТрансТелеКом. Ядром сети являются высокопроизводительные коммутирующие маршрутизаторы производства компании Cisco Systems. Пограничный слой состоит из маршрутизаторов Cisco Systems, обеспечивающих агрегирование клиентского трафика абонентов IP-сети и коммутаторов Fast Ethernet для объединения инфраструктуры узла и для подключения оборудования пользователей. В состав IP-сети входит система управления устройствами и услугами и комплекс серверов, обеспечивающих традиционные услуги Интернет, такие как DNS, SMTP, WWW.

Первым же услуги IP VPN на базе MPLS стал предоставлять Раском, а чуть позже — Эквант. MPLS поддерживается во всех узлах магистральной сети Раскома. Его сеть тоже полностью строится на оборудовании Cisco. Основу ее составляют мощные маршрути-

заторы Cisco GSR 12000, устанавливаемые на московской и питерской площадках Раском. Каждый из них пропускает по сети трафик до 40 Гбит/с. Клиентский трафик агрегируется маршрутизаторами Cisco 7000, соединенными с опорной сетью гигабитовыми оптическими интерфейсами.

Поэтапный переход к MPLS осуществляет в настоящее время крупнейший российский Интернет-провайдер РТКомм.Ру. Основа сети компании — магистральная сеть «Ростелекома», взятая в аренду, и на 78 из 132 узлов уже используется технология MPLS.

Среди семи межрегиональных операторских компаний «Связьинвеста» первые мультисервисные IP/MPLS сети были построены компанией Уралсвязьинформом и Южной телекоммуникационной компанией, чья MPLS-сеть работает в Краснодарском крае с 2002 года.

Оборудование IP/MPLS используется в принадлежащей компании МГТС сети передачи данных общего пользования, доступ к которой обеспечивается по существующей кабельной распределительной сети МГТС с использованием технологий xDSL. Основа сети — магистральное ядро, которое объединяет 10 мощных узлов пакетной коммутации, соединенных по волоконно-оптическим линиям каналами STM 16 (технология IP/MPLS) и STM 4 (технология ATM). Первое кольцо — это основная рабочая магистраль, а второе служит «горячим» резервом для первого, и выполняет также служебные и технологические функции для ряда систем МГТС. В магистральных узлах сети установлены маршрутизаторы Cisco GSR 12012 и концентраторы Cisco 6400.

Другой московский MPLS-проект реализован компанией Комстар, теперь объединившейся с Телмос и МТУ-информ под общим названием «Комстар-объединенные системы». Но строительство MPLS сети она начала еще в одиночку в начале 2003 года. Сеть строилась на оборудовании производителя Riverstone Networks, совместимом с коммутаторами и мультиплексорами Alcatel, на которых построена сеть ATM/Frame Relay компании МТУ-Информ, что облегчило их объединение. В MPLS-сети Комстар внедрены приложения Traffic Engineering и Fast ReRoute, и предоставляются услуги VPN, как на уровне 3, так и на уровне 2 (VPLS).

В октябре 2003 года компания Голден Телеком объявила о заключении соглашения с компанией AT&T о предоставлении доступа к защищенной IP сети SIPN (SWIFT Secure IP Network) на территории России и других стран СНГ. В соответствии с соглашением, заключенным со SWIFT, доступ к сети будет производиться через сеть IP VPN, построенную на базе технологии MPLS. В декабре 2003 года Голден Телеком успешно подключила к SWIFT SIPN первого клиента.

Даже этот, не претендующий на полноту обзор MPLS-решений в России демонстрирует, что относительно молодая технология MPLS уже прочно закрепилась на отечественном рынке. На ее базе реализуется ряд масштабных национальных и международных проектов.

11.4. Тестирование MPLS

Платформы тестирования рассмотренного в предыдущем параграфе оборудования MPLS реализуются как в виде сетевых программно-аппаратных систем мониторинга, так и в виде специализированных протокол-тестеров. На рис. 11.1 показаны варианты подключения тестовых устройств к рассматривавшемуся в предыдущих главах домену сети MPLS.

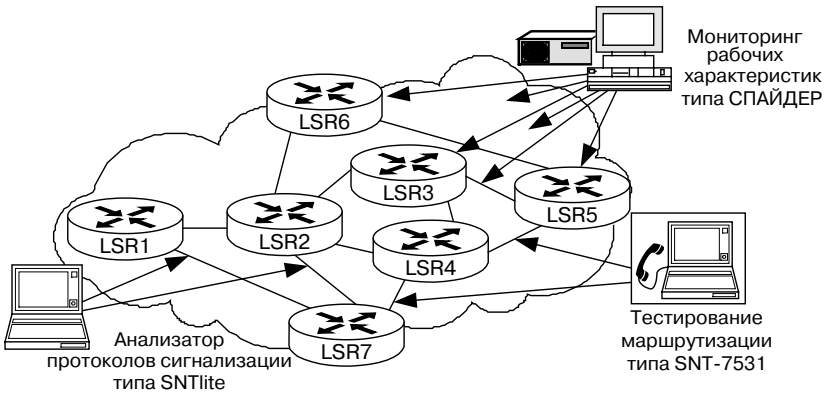


Рис. 11.1 Варианты тестирования MPLS

Как правило, речь идет не о специализированных протокол-тестерах и системах мониторинга, а о соответствующих программно-аппаратных модулях MPLS, вводимых к в существующее тестовое оборудование ведущих мировых компаний.

Так, компания *Agilent Technologies* выпускает два типа тестового оборудования для MPLS: *QA-Robot*, являющийся системой тестирования соответствия, и *RouterTester* — эмулятор MPLS и тестер маршрутизации, способные тестировать функциональные возможности MPLS совместно с протоколами сигнализации LDP и RSVP-TE. Тестовое оборудование позволяет генерировать запросы меток, эмулировать процедуры распределения меток, выполнять маршрутизацию по протоколам BGP-4, OSPF и IS-IS с соответствующими расширениями MPLS-TE. Тестер QA-Robot позволяет эмулировать перегрузки в одном LSP, а RouterTester расширяет эти функции возможностями анализа производительности сети MPLS при обслуживании разнородного трафика (речи, данных и видео). Функциональные возможности тестового оборудования *Agilent Technologies* иллюстрирует табл. 11.1.

Таблица 11.1. Функциональные возможности тестирующего оборудования Agilent

MPLS (LDP/CR-LDP) Эмуляция			
	OA Robot	RouterTester	
LDP/CR-LDP параметры	Hold Timer & KeepAlive Таймер для организации сеансов и техобслуживания		
Поддерживаемые сообщения	Hello		
	KeepAlive, Address, Address Withdraw		
	Initialization		
	Label Request (LDP & CR-LDP)		
	Label Mapping (LDP & CR-LDP)		
	Label Release (LDP & CR-LDP)		
	Label Withdraw (LDP & CR-LDP)		
	Notification (Fatal & Advisory)		
Label Request поле сообщений			
	OA Robot	RouterTester	
Общий заголовок	Автоматически из описания интерфейса		
FEC	Значения, задаваемые пользователем, или взятые из IGP/BGP/SSM		
LSPID	Значения, задаваемые пользователем, или взятые из IGP/BGP/SSM		
Hop Count	Значения, задаваемые пользователем, или взятые из IGP/BGP/SSM		
Path Vector	Значения, задаваемые пользователем, или взятые из IGP/BGP/SSM		
Explicit Route	Предусмотрен список Virtual LSRs		
Traffic TLV	Задается пользователем		
Resource Class TLV	Задается пользователем		
Preemption TLV	Задается пользователем		
Pinning TLV	Задается пользователем		
Label Mapping поле сообщений			
	OA Robot	RouterTester	
Критерий	Label Mapping автоматически создается и передается в ответ на Label Request		
FEC	Задается пользователем или берется из IGP/BGP/SSM		
LSPID	Задается пользователем или берется из IGP/BGP/SSM		
Hop Count	Задается пользователем		
Label Request Msg ID TLV	Переносится из принимаемого Label Request		
Traffic TLV	Переносится из принимаемого Label Request, если не задано в дескрипторе		
Label TLV	Берется из Label Manager или 0		
Path Vector	Задается пользователем		
User defined TLV	Задается пользователем		
Global Statistics глобальная статистика			
	OA Robot	RouterTester	
Действующие LSP ('s)	Общее число действующих LSP		
Inbound/Outbound Statistics внутренняя и внешняя статистика			
	OA Robot	RouterTester	
Label Request Messages (суммарное число и скорость)	Число принятых Label Request		
Label Mapping Messages (суммарное число и скорость)	Число принятых Label mapping		
Label Release messages (суммарное число и скорость)	Число принятых Label Release		
Label Withdraw messages (суммарное число и скорость)	Число принятых Label Withdraw		
Скорость организации LSP	Значение скорости организации LSP		
MPLS производительность			
	OA Robot	RouterTester	
Действующие LSP ('s)	700 MHz, 256 MB	GbE 128MB	OC3/12/48 32 MB
Скорость создания LSP	1600/s	Собираемые данные	

Другая компания, *Ixia,* разработала разных типов генераторы трафика и анализаторы рабочих характеристик, которые можно использовать для тестирования MPLS. Специальное программное обеспечение Ixia эмулирует RSVP-TE и может использоваться для создания тысяч туннелей MPLS.

Еще одна компания *Spirent Communications* производит анализатор, генератор и эмулятор ADTECH для MPLS.

Отечественная компания *ГК Экран* реализовала весьма схожий с использованным Agilent Technologies подход в протокол-тестерах *SNT (Signaling Network Testing).* В перечень тестируемых протоколов входят LDP, CR-LDP, RSVP, RSVP-TE. Отдельный программный комплекс обеспечивает анализ и тестирование маршрутизации, как внутридоменовой (с помощью протоколов OSPF и IS-IS), так и междоменовой (на базе BGP-4). Аналогично представленному в таблице 11.1, эти функции масштабируются как «вверх», до уровня *платформы сетевого мониторинга СПАЙДЕР*, первоначально созданной для стека протоколов ОКС7, но имеющей специальные опции для мониторинга сети MPLS, так и «вниз», до уровня младшей модели протокол-тестера *SNTlite.*

Масштабируемость протокол-тестеров MPLS компании *Anritsu* типа *MD 1231A* близка к вышеупомянутым отечественным системам и иллюстрируется табл. 11.2.

Таблица 11.2. Протокол-тестеры компании Anritsu

Функция		MU120101A 10/100 Mb Ethernet модуль	MU120102A Gigabit Ethernet модуль	MU120111A 10/100 Mb Ethernet модуль	MU120112A Gigabit Ethernet модуль
MD1231A-07 — опция протокола OSPF				x	x
MD1231A-08 MPLS — опция протокола LDP/CR-LDP				x	x
MD1231A-09 MPLS — Опция протокола RSVP				x	x
MD1231A-10 RFC2889 — Опция теста Benchmarking			x	x	x
Протокольные расширения и дополнения	BGP-4 расширенные функции эмуляции			x	x
	IS-IS функции декодирования	x	x	x	x
	OSPF функции декодирования	x	x	x	x
	MPLS (LDP/CR-LDP) функции декодирования	x	x	x	x
	MPLS (RSVP) функции декодирования	x	x	x	x
1000BASE-T поддержка		N/A		N/A	x
Поддержка режима мониторинга		x	x	x	x
Измерения BER	Не фрагментированный BER-тест		x	x	x
	Опция BER-теста пакетов		x	x	x

Компания *Randcom Inc.* выпускает протокол-анализатор PrismLite и анализатор Packet over SONET (PoS) типа WireSpeed 622 PoS.

Компания *NetTest* выпускает протокол-тестер, весьма схожий с изображенными на рис. 11.1 тестерами SNT. Впрочем, сходство только внешнее. Функции тестирования и оценки производительности сети MPLS в оборудовании NetTest выполняются совместно системами *interWATCH* и *interEMULATOR*. Объектами этих совместных усилий являются распределение меток MPLS (LDP/CR-LDP/RSVP/RSVP-TE) и протоколы маршрутизации BGP-4/OSPF(TE)/IS-IS(TE).

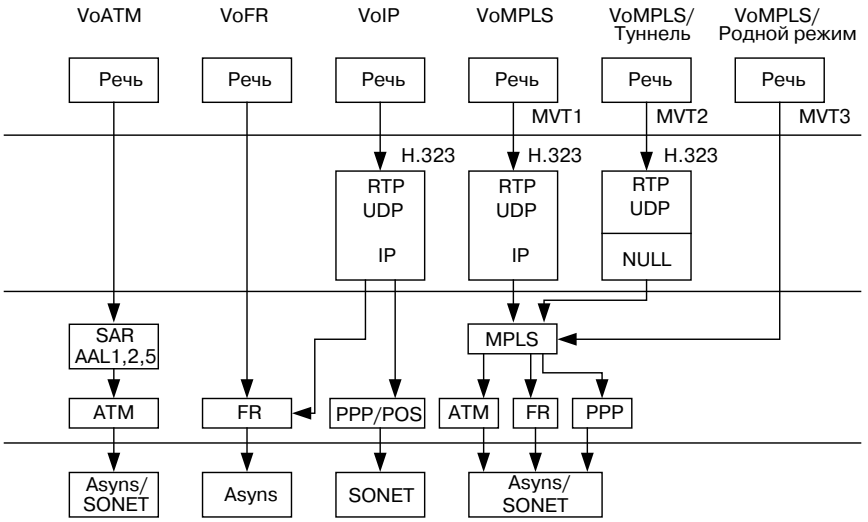
11.5. VoMPLS

Предложение по MPLS-телефонии (VoMPLS) было внесено в IETF и отвергнуто в 2000 г. Аргументация такого категорического отказа обусловлена все возрастающей ролью IP в целом и IP-телефонией (VoIP) в частности, а следовательно, перспективностью технологии VoIPoMPLS, которая и рассматривается в этой книге. Действительно, при кажущемся упрощении, изъятие IP из стека протоколов ограничивает область коммуникаций облаком MPLS и делает невозможной доставку речевой информации на рабочее место пользователя. При этом, благодаря механизму сжатия/подавления заголовков, передаваемые по VoIPoMPLS данные выглядят аналогично тому, что предлагает VoMPLS, обеспечивая ту же экономичность и глобальный охват.

Тем не менее, VoMPLS, являясь новой технологией, уже имеет ряд сторонников; ей посвящена отдельная книга [34], в которой представлен следующий стек протоколов для передачи речи по пакетным сетям (VoIP) в общем и для VoMPLS в частности (рис. 11.2).

Три представленных на рис. 11.2 режима VoMPLS имеют следующие свойства.

В *режиме MPLS Voice Type 1 (MVT1)* используется H.323-инкапсуляция протоколов более высокого уровня, включая IP, после чего речевые пакеты передаются по сети MPLS. Преимущество этого режима состоит в том, что он позволяет разработчикам использовать весь аппарат H.323, включая шлюз, привратник, а также технологию Softswitch. Недостаток этого режима в большом объеме служебной информации, вносимой последовательностью заголовков H.323, даже с учетом их сжатия. Существуют также концептуальные сомнения по поводу смешения режима протокола UDP, передающего дейтаграммы пользователя, и потокового режима MPLS.



RTP = Протокол транспортировки в реальном времени
UDP = Протокол передачи дейтаграмм пользователя
PPP = Протокол двухточечной связи
MVTx = MPLS Voice Type x (x=1, 2, 3)

Рис. 11.2. Стек протоколов для Voice over Packet

В *режиме MPLS Voice Type 2 (MVT2)* тоже используется H.323-инкапсуляция, но уже без IP, а для создания соединений на основе LSP может использоваться механизм туннелирования.

Режим MPLS Voice Type 3 (MVT3) — «родной» режим передачи речи, который во многом схож с режимом VoATM передачи речи по сети ATM. В этом режиме речевые биты переносятся непосредственно в MPLS-пакетах, однако, требуют решения проблемы адресации и синхронизации, которые в ATM решены.

Какой бы из трех режимов VoMPLS ни использовался, в качестве протоколов сигнализации MPLS могут применяться как протокол CR-LDP, так и протокол RSVP.

Таким образом, можно констатировать, что сегодня существуют два направления в решения задач передачи речевого трафика в сети MPLS. Одно направление — *речь-через-IP-через-MPLS (VoIPoMPLS),* которое использует IP-инкапсуляцию речевого пакета и протокол транспортировки в реальном времени (RTP) в качестве протокола прикладного уровня. Заметим, что RTP передается далее поверх UDP для транспортировки IP через MPLS поверх таких технологий низкого уровня, как FR, ATM PP или Ethernet. Другое направление — *VoMPLS*, которое использует размещение пакетизированных речевых выборок непосредственно в пакетах MPLS.

В дополнение к VoMPLS начинается также работа над передачей мультимедийного трафика через сети IP (MMoIP), при которой будет использоваться MPLS. Этот трафик обладает своим набором требований, и как раз сейчас начинается исследование структуры для этой технологии.

11.6. «Все» через MPLS

Важным направлением развития MPLS является использование этой технологии для пересылки трафика уровня 2 разных типов через специальным образом конфигурированные LSP-туннели, уже рассматривавшиеся на страницах этой книги. Здесь речь идет о переносе над уровнем MPLS трафика уровня звена данных. Таким образом, MPLS будет выполнять роль универсального механизма доступа и такого же универсального трубопровода для разнотипного трафика в единой сети NGN. Эта техника может использоваться для создания сетей VPN уровня 2, а а также для пересылки Ethernet-трафика виртуальной сети LAN (VLAN) через домены MPLS. и Ethernet. В число других транспортных технологий низкого уровня, для которых рассматривается применение этого механизма, входят ATM AAL5, ATM Cell Relay, Frame Relay, PPP, HDLC и SDH.

ATM AAL5 через MPLS инкапсулирует сервисные блоки данных SDU (Service Data Units) в MPLS-пакеты и пересылает их через MPLS сеть, как это показано на рис. 11.3. Каждый AAL5 SDU переносится в отдельном пакете. Для этого входной LSR, получив AAL5 SDU, удаляет его заголовок. Затем LSR копирует элементы управляющего слова (control word) из заголовка в соответствующие поля SDU (в частности, биты EFCI и CLP), добавляет метки виртуального канала VC и метку LSP-туннеля, и дальше пакет стандартным образом пересылается по MPLS-сети. Приняв такой пакет, выходной LSR копирует элементы управляющего слова в заголовок, удаляет VS- и LSP-метки (если последняя еще осталась), добавляет AAL5 заголовок и отправляет пакет соответствующий интерфейс.

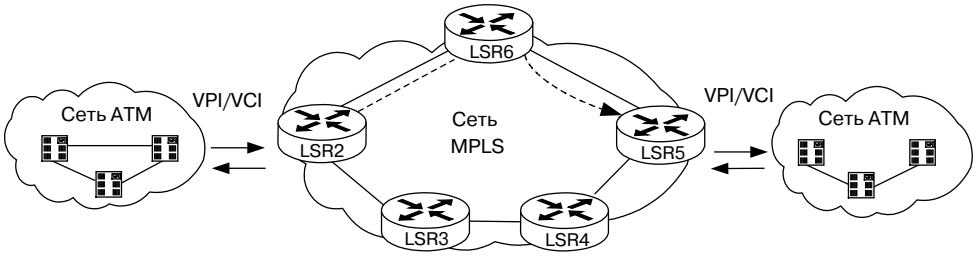


Рис. 11.3. Модель ATM через MPLS

ATM Cell Relay через MPLS переносит одиночные конверты ATM через сеть MPLS. Поддерживается только режим PVC. Здесь входной LSR получает конверт ATM и удаляет заголовок, за исключением управляющего слова (EFCI и CLP), а также VPI и VCI. Затем LSR добавляет VC-метку и метку LSP, и пакет пересылается по сети MPLS стандартным образом. Выходной LSR получает пакет, удаляет LSP- и VC-метки и управляющее слово, затем вставляет ATM-заголовок и отсылает пакет в соответствующий интерфейс.

Ethernet через MPLS переносит Ethernet PDU в MPLS-пакетах. Входные маршрутизаторы LSR транслируют адреса Ethernet MAC и поддерживают метки MAC для отображения входного и выходного портов. Каждый PDU переносится в отдельном пакете. При получении PDU входной LSR удаляет преамбулу, разделитель начала кадра — start of frame delimiter (SFD) — и проверочную комбинацию FCS. Остальная часть заголовка сохраняется прежней. LSR копирует контрольное слово из заголовка, даже если оно не используется, добавляет метки VC и LSP, и пакет пересылается по сети MPLS стандартным образом. Выходной LSR удаляет метки LSP и VC и управляющее слово. Затем он обновляет заголовок и отсылает пакет в соответствующий интерфейс. LSR может также дополнительно поддерживать мэппинг VLAN. Для инкапсуляции этого типа весь кадр Ethernet уровня 2 (без преамбулы и FCS) пересылается в поле данных пакета MPLS. Транспортируется также тег VLAN. Входной и выходной маршрутизаторы LSR могут оценивать поле приоритета в заголовке VLAN с целью мэппинга в поле EXP заголовка MPLS.

Frame Relay через MPLS инкапсулирует PDU Frame Relay в пакеты MPLS и пересылает их по сети MPLS. По аналогии с рис. 11.3, входные LSR транслируют DLCI согласно таблицам отображения меток виртуальных каналов во входные и выходные интерфейсы и отображает бит права на отбрасывание DE в соответствующее значение EXP в заголовке MPLS. В выходном LSR могут генерироваться ошибки Q.933 и Q.922, а также аварийные сигналы. Следует учитывать зависимость процесса транспортировки PDU от выбранного режима соединения: DLCI-to-DLCI или Port-to-Port. При режиме *DLCI-to-DLCI* входной LSR получает Frame Relay PDU и удаляет FR-заголовок и проверочную комбинацию (FCS), затем копирует элементы управляющего слова из заголовка FR в соответствующие поля FR PDU, в частности, Backward Explicit Congestion Notification (BECN), Forward Explicit Congestion Notification (FECN), Discard Eligibility (DE), вставляет метки VC и LSP, и пакет пересылается по сети MPLS стандартным образом. Выходной LSR, получив пакет, копирует элементы управляющего слова в заголовок FR, удаляет метки VC и LSP, добавляет FR-заголовок и пересылает пакет в соответствующий интерфейс. При соединении *Port-to-Port* для транспортировки инкапсулированных FR-пакетов используется режим HDLC.

В этом режиме HDLC-пакет транспортируется целиком, за исключением флагов HDLC и FCS. Содержимое пакета не используется и не изменяется, включая биты FECN, BECN и DE.

HDLC через MPLS инкапсулирует HDLC PDU в MPLS пакеты и пересылает их через сеть MPLS. При этом LSR не участвуют ни в каких протокольных процессах согласования или аутентификации. Получив HDLC PDU, входной LSR удаляет флаги и FCS. Затем LSR копирует управляющее поле в PDU, даже если оно не используется, добавляет метки VC и LSP, и пакет пересылается по сети MPLS стандартным образом. Выходной LSR получает пакет, удаляет метки VC и LSP, добавляет флаги и FCS и пересылает пакет в соответствующий интерфейс.

PPP через MPLS инкапсулирует PPP PDU в MPLS-пакеты и пересылает их по сети MPLS. При этом LSR не участвуют в протокольных процессах согласования и аутентификации. Входной LSR, получив PPP PDU, удаляет флаги, адрес, управляющее поле, FCS и добавляет метки VC и LSP, после чего пакет пересылается по сети MPLS стандартным образом. Выходной LSR удаляет метки VC и LSP и, добавив флаги, адрес, управляющее поле и FCS, отправляет пакет в соответствующий интерфейс.

11.7. Многоадресность

Организация с помощью MPLS рассылки «точка — группа точек», более известной по аббревиатуре P2MP (Point-to-Multipoint), давно привлекала разработчиков, но наталкивалась на множество сложностей. Дело в том, что MPLS изначально создавалась для передачи *одноадресного (unicast) трафика*. Режим P2MP требует *многоадресности (multicast*), т.е. возможности передавать один пакет получателям многоадресной группы. Таким образом, передается не один пакет, а много его копий.

Для создания LSP по принципу P2MP потребуется довольно интенсивный служебных трафик, особенно если учесть непостоянство и динамичность групп многоадресной рассылки. В многоадресном режиме в MPLS очень сложно будет организовать агрегацию как одноадресного трафика с многоадресным, так и многоадресного трафика разных групп. В MPLS распределением меток руководит нижний LSR, и верхний LSR не может гарантировать, что все нижние LSR назначат одинаковые метки, в случае же назначения меток сверху, верхний LSR мог бы назначить всем LSR в группе многоадресной рассылки единую метку.

Несмотря на эти препятствия, IETF и некоторые форумы исследуют возможность реализовать многоадресность в MPLS. Одной из областей, в которых проводятся исследования, являются приложения BGP/MPLS VPN. Один проект, находящийся в настоя-

щее время на грани перехода из состояния *draft* к более высокой степени стандартизации, и озаглавленный как «Многоадресность в сетях BGP/MPLS VPN», дополняет основные спецификации BGP/MPLS VPN описанием протоколов и процедур, которые провайдер услуг может реализовать для обслуживания многоадресного VPN-трафика с помощью протокола многоадресной маршрутизации PIM.

Сегодня ни CR-LDP, ни RSVP-TE многоадресность не поддерживают, но в MPLS WG разрабатывается находится *draft*-проект, сначала имевший название «Требования к P2MP-расширению протокола RSVP-TE», а сейчас озаглавленный «Требования к MPLS TE-LSP». Он призван специфицировать требования для функционирования P2MP-приложений поверх инфраструктуры MPLS-TE, причем не только в традиционной MPLS, но также и в сетях GMPLS.

Хотя работа по стандартизации P2MP еще далека от завершения, существует ряд реализованных проектов, одним из которых является совместный проект Nippon Telegraph and Telephone Corp. (NTT) и Motorola, реализованный осенью 2003 года и обеспечивающий поддержку QoS и инжиниринга трафика при многоадресной маршрутизации.

11.8. DiffServ-aware MPLS-TE

11.8.1. Объединение технологий

В этом пункте рассматривается весьма многообещающая концепция DiffServ-aware-MPLS-TE, также называемая MPLS DiffServ-TE. По сути, это еще один подход к инжинирингу трафика в MPLS, и отличается он от других подходов не сигнализацией или иными сетевыми механизмами, а ориентацией целевой функции не на оптимизационные задачи, а на гарантирование качества предоставления услуг.

В соглашениях об уровне обслуживания SLA задаются требуемые показатели качества обслуживания проходящего по сети трафика — задержка, джиттер, гарантированная полоса пропускания, устойчивость к отказам и время простоя. Формулируемые в SLA требования обуславливают как наличие определенных механизмов распределения ресурсов, организации очередей и отбрасывания трафика в зависимости от типа приложения, так и наличие гарантий предоставления полосы пропускания для каждого приложения.

До настоящего времени поставщики услуг использовали для обеспечения QoS только модель DiffServ, и механизм распределения ресурсов базировался на назначении для приложений разных классов обслуживания и на соответствующей маркировке трафика. Однако для обеспечения SLA недостаточно маркировать трафик.

Если он следует по маршруту, не имеющему адекватных ресурсов для выполнения требований к качеству функционирования (например, к задержке или к джиттеру), то соглашения SLA физически не смогут быть выполнены.

MPLS-TE позволяет создавать коммутируемые по меткам тракты через звенья, имеющие надлежащие ресурсы, тем самым гарантируя, что для обслуживаемого потока всегда будет иметься достаточная полоса пропускания, и что перегрузка будет предотвращена как в стабильном режиме работы, так и в случаях сетевых отказов. Недостаток простого совмещения DiffServ и MPLS-TE состоит в том, что MPLS-TE «забывает» о разделении потоков по классам обслуживания (CoS) и функционирует в доступной полосе пропускания обобщенно (одинаково для всех классов).

11.8.2. DiffServ в MPLS

Подход DiffServ к проблеме обеспечения QoS заключается в разделении всего трафика на небольшое число классов и на выделении сетевых ресурсов отдельно для каждого такого класса, а не для каждого информационного потока. Чтобы устранить необходимость в протоколе сигнализации, класс маркируется непосредственно в поле DiffServ Code Point (DSCP) пакета. Это поле длиной 6 битов является частью первоначально введенного в заголовках IP-пакета байта ToS (тип обслуживания). Дело в том, что IETF переопределил значение редко используемого поля ToS, разбив его на 6-битовое поле DSCP и 2-битовое поле уведомления о явной перегрузке ECN (Explicit Congestion Notification), как показано на рис. 11.4.

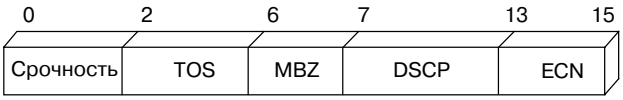


Рис. 11.4. Поля ToS и DSCP + ECN

Поле DSCP определяет уровень обслуживания пакета в данном сетевом узле. Для этого уровня обслуживания используется термин *режим пересылки PHB (Per-Hop Behavior)*, который выражает порядок обработки пакета в узле в плане очередности его диспетчеризации и отбрасывания. С точки зрения реализации PHB определяет очередность пересылки пакетов, вероятность отбрасывания пакетов в том случае, когда очередь становится длиннее заданного порога, ресурсы (буферную емкость и полосу пропускания), выделяемые каждой очереди, а также частоту, с которой обслуживается очередь.

IETF определил набор из 14 стандартных классов обслуживания трафика. В их число входят *класс негарантированного обслуживания BE (Best Effort)*, при в котором трафик не получает никакой

специальной обработкии и *класс срочной пересылки пакетов EF (Expedited Forwarding)*, при в котором трафик встречает минимальную задержку и низкую вероятность потерь. С практической точки зрения это означает очередь, выделенную трафику класса EF, у которой частота поступления в нее пакетов меньше, чем скорость обслуживания пакетов, так что задержки, джиттер и потери, вызванные перегрузкой, маловероятны. Типичными примерами трафика, которому присваивается класс EF, являются потоки речевой и видеоинформации: они имеют постоянные скорости передачи и требуют минимальных задержек и потерь.

Остальные 12 классов составляют *классы гарантированной пересылки пакетов AF (Assured Forwarding)*. Каждый такой класс определяется номером очереди и очередностью отбрасывания пакетов. IETF рекомендует использовать четыре разные очереди, каждая из которых имеет три уровня очередности отбрасывания, что и дает в итоге двенадцать классов AF. Для этих классов принята система наименований AF*xy*, где *x*- номер очереди, а *y* — уровень очередности отбрасывания. Таким образом, все пакеты класса AF*xy* будут помещаться в одну и ту же очередь *x* для их дальнейшей пересылки, и это гарантирует, что пакеты одного приложения не будут переупорядочены, если они различаются только очередностью отбрасывания. Классы AF применимы к трафику, который требует гарантий скорости передачи, но не требует гарантированных предельных значений задержки или джиттера.

Хотя IETF определила рекомендуемые значения кода DSCP для каждого из стандартных классов трафика, поставщики позволяют операторам сетей переопределять соответствие между DSCP и PHB, а также определять нестандартные режимы PHB. Важно иметь в виду, что как только пакет маркируется каким-то значением DSCP, тем самым определяется его обработка в плане уровня обслуживания в каждом из сетевых узлов, через которые он проходит. Следовательно, для обеспечения согласованной QoS-обработки важно поддерживать согласованное соответствие DSCP-PHB.

Это обстоятельство и обуславливает понятие *домен DiffServ*, который представляет собой совокупность поддерживающих DiffServ узлов с одинаковым набором заданных PHB, одинаковым соответствием кодов DSCP режима PHB и единой стратегией обеспечения услуг (рис. 11.5). Обычно домен DiffServ функционирует под управлением одного администратора. На границе домена DiffServ трафик маркируется кодами DSCP, которые задают желательный PHB (т.е. уровень обслуживания в каждом из сетевых узлов) и, в конечном итоге,— желательное качество обслуживания. Таким образом, система DiffServ обеспечивает различную обработку трафика в узлах, обеспечивая тем самым выполнение разных требований QoS для разных потоков. Этот подход является масштабируемым и не требующим сигнализации для каждого потока трафика, но он не может

гарантировать QoS, если тракт, по которому идет трафик, не обеспечен адекватными ресурсами.

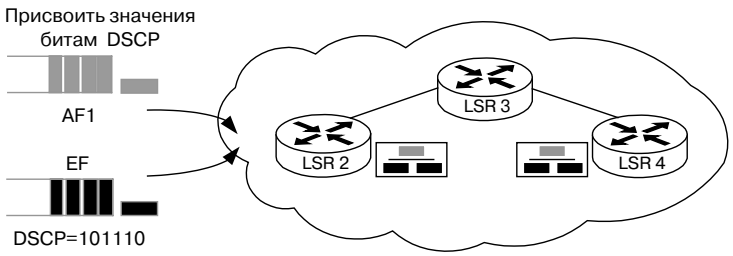


Рис. 11.5. Домен DiffServ

Механизмы поддержки DiffServ в сети MPLS описывает документ RFC 3270. Первая проблема, связанная с поддержкой технологии DiffServ в сети MPLS, состоит в том, что LSR принимают решения о пересылке пакетов на основании только информации, содержащейся в метке, так что PHB должен указываться в ней, а именно — в трех экспериментальных битах EXP метки. Этот способ решает исходную проблему передачи в метке MPLS информации о желательном режиме PHB, но порождает новую: каким образом отобразить значения кода DSCP, выраженные в 6-битовом поле (т.е. до 64 разных значений), на 3-битовое поле EXP, которое может переносить максимум восемь разных значений? Существуют два возможных решения этой проблемы.

Первое решение применяется к сетям, которые поддерживают менее восьми режимов PHB. Здесь с определением соответствия между кодами DSCP и режимами PHB все просто: каждый код DSCP эквивалентен определенной комбинации битов EXP и соответствует определенному PHB. При пересылке пакетов значение метки, содержащейся в пакете, определяет, куда его пересылать, а биты EXP определяют PHB для этого пакета. Комбинация битов EXP может задаваться на основании значений битов DSCP пакетов IP, переносимых по LSP, или назначаться администратором сети. Тракты LSP, для которых режимы PHB логически выводятся из значений битов EXP, обозначаются *E-LSP,* и по ним могут передаваться пакеты, имеющие до восьми разных PHB в одном LSP.

Второе решение применяется к сетям, которые поддерживают более восьми классов трафика, для чего битов EXP недостаточно. Единственным другим полем в MPLS-заголовке пакета, которое можно использоваться для этой цели, является сама метка. При пересылке пакетов, значение метки определяет, куда пересылать пакет и какое обслуживание ему предоставить в плане диспетчеризации, а биты EXP переносят только информацию, относящуюся к приоритету отбрасывания, который присвоен данному пакету. Таким образом, режим PHB определяется на основании как метки, так

и битов EXP. Поскольку метка в неявном виде привязана к PHB, эта информация должна сообщаться при создании тракта LSP. Тракты LSP, которые используют метку для передачи информации о желательном PHB, обозначаются *L-LSP*. По трактам L-LSP могут передаваться пакеты, относящиеся к трафику одного PHB или нескольких PHB, которые имеют одинаковый режим диспетчеризации, но различаются приоритетами отбрасывания пакетов, например, классов AF*xy*, где *x* — величина постоянная, а значения *y* разные. Различия между трактами E-LSP и L-LSP сведены в табл. 11.3.

Таблица 11.3. Сравнение трактов E-LSP и L-LSP

E-LSP	L-LSP
PHB задается битами EXP	PHB задается меткой или же сочетанием метки и битов EXP
Один LSP может передавать трафик до восьми разных PHB	Один LSP поддерживает один PHB или несколько PHB с одинаковым режимом диспетчеризации и с разными приоритетами отбрасывания (отсева)
Консервативное (обычное) использование меток, поскольку метка используется только для передачи информации о тракте	Расширенное использование меток и информации о состоянии, поскольку метка передает информацию как о тракте, так и о режиме диспетчеризации
Не требуется никакой сигнализации для передачи информации о PHB	Информация о PHB должна сообщаться при создании тракта LSP
Когда в сети используются только тракты E-LSP, сеть может поддерживать до восьми PHB. Когда требуется поддержка большего числа PHB, могут использоваться тракты E-LSP в сочетании с трактами L-LSP	В сети может поддерживаться любое число PHB

11.8.3. Class of Type — CT

Напомним уже затронутую выше проблему, заключающуюся в том, что технология MPLS-TE функционирует на совокупном, агрегатном уровне по всем классам обслуживания DiffServ и, как результат, не в состоянии дать гарантированную полосу пропускания по каждому классу. Базовое же требование DiffServ-TE — быть в состоянии отдельно резервировать полосу пропускания для трафика каждого класса. Это подразумевает необходимость отслеживать во всех маршрутизаторах сети того, какая полоса пропускания доступна для трафика каждого класса в любой момент времени.

Для этой цели в RFC 3564 вводится понятие *класс типа CT* (*Class of Type)*, которое определяется как совокупность ограничений по полосе пропускания звена данных. С помощью CT производится маршрутизация с учетом ограничений полосы пропускания звена и управление доступом. Предусматривается до восьми CT, обозначаемых CT0 ...CT7. По соглашению, трафику негарантированного обслуживания соответствует CT0. Тракты LSP, которые рассчитываются так, чтобы гарантировать полосу пропускания для опреде-

ленного CT, обозначаются *DiffServ-TE LSP*. Согласно разработанной IETF модели DiffServ-TE, LSP может передавать трафик только одного и того же CT и использовать при этом одинаковые или разные приоритеты вытеснения трактов потоков. Поскольку все тракты, предшествующие DiffServ-TE LSP, считаются трактами негарантированного обслуживания, им соответствует CT0.

11.8.4. Вычисление пути

Маршруты в MPLS-TE рассчитываются по алгоритму CSPF (кратчайший путь выбирается первым с учетом ограничений) в соответствии с задаваемыми оператором ограничивающими условиями по полосам пропускания звеньев. Технология DiffServ-TE добавляет доступную каждому из восьми CT полосу пропускания в качестве ограничивающего условия, которое может применяться к создаваемому LSP. Следовательно, происходит модернизация уже рассматривавшегося в главе 5 алгоритма Дийкстры, позволяющая принимать во внимание полосу пропускания, которая относится к определенному CT, в качестве ограничивающего условия при расчете тракта. Для того чтобы расчет оказался успешным, в каждом звене должна быть известна полоса пропускания, доступная для каждого CT на всех уровнях приоритета.

Это означает, что протоколы состояния звена (IGP) должны уведомлять о доступной в каждом звене полосе пропускания для каждого CT на каждом уровне приоритета. Вспомним, что имеется восемь CT и восемь уровней приоритета, что дает 64 значения, которые должны передаваться протоколами состояния звена. Однако IETF принял решение ограничить число значений восемью.

Для этой цели определен так называемый *TE-класс,* определяющий комбинацию <CT, приоритет>. DiffServ-TE поддерживает максимум восемь TE-классов, от TE0 до TE7 включительно, которые могут выбираться из 64 возможных комбинаций <CT, приоритет> посредством конфигурирующих процедур, как это показано на рис. 11.6. На этом рисунке представлены 64 комбинации CT и приоритета, из которых выбрано восемь TE-классов.

Протоколы IGP переносят уведомление о полосе пропускания, доступной для каждого TE-класса. Существующие проекты спецификаций IETF обязывают, чтобы это уведомление делалось с использованием *Unrestricted Bandwidth TLV*, которое ранее служило для распределения нерезервированной полосы пропускания при инжиниринге трафика. Следовательно, информация, полученная алгоритмом CSPF с помощью протоколов IGP, относится только к комбинациям CT и приоритета, которые формируют действующие TE-классы. Таким образом, для того чтобы CSPF мог выполнить достоверный расчет, CT и уровни приоритета, выбранные для LSP, должны соответствовать одному из выбранных TE-классов.

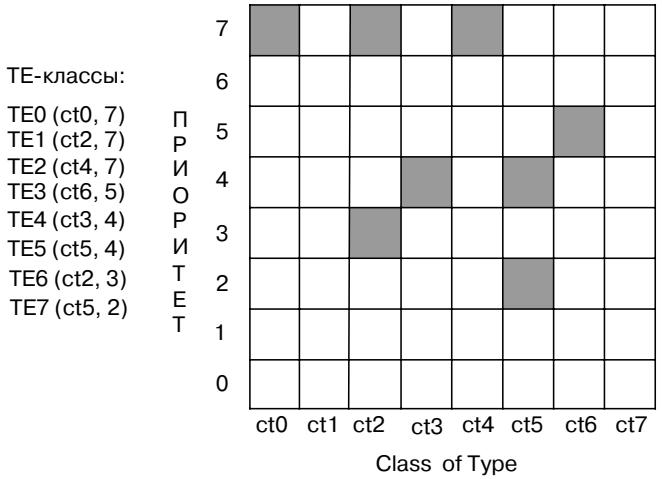


Рис. 11.6. Выбор восьми TE-классов из 64 возможных комбинаций

11.8.5. Сигнализация тракта

Результаты расчета тракта сообщаются всем маршрутизаторам, через которые проходят используемые этим трактом маршруты, а затем в каждом маршрутизаторе выполняется контроль доступа и учет полосы пропускания. Существующие проекты спецификаций IETF определяют необходимые расширения протокола RSVP-TE, которые позволяют ему создавать тракты с резервированием полосы пропускания для каждого CT. Информация о CT для LSP передается в новом объекте *Class of Type (CT)* в составе сообщения Path протокола RSVP. Возможность постепенного развертывания DiffServ-TE в сети обеспечивается тем, что:

- объект CT присутствует только в сообщениях для трактов LSP от CT1 до CT7 включительно (если объект CT отсутствует, то предполагается CT0),
- LSP, который получает сообщение Path с объектом CT и не распознает его, дает отказ в организации тракта.

Выполнение этих двух правил гарантирует, что LSP с резервированием для каждого CT можно организовать только через LSP, которые поддерживают DiffServ-TE, в то время как «старые» тракты (до введения DiffServ-TE), которые считаются принадлежащими CT0, могут проходить как через старые, так и через новые LSP.

Информация о CT, передаваемая в сообщении Path, специфицирует CT, согласно которому выполняется контроль доступа в каждом LSP тракта. Если узел определяет, что имеется достаточно ресурсов, и что новый LSP приемлем, он подсчитывает новую полосу пропускания для CT и назначает уровень приоритета. Эта информация затем передается обратно по протоколу IGP.

11.8.6. Модели назначения полосы пропускания

Одним из самых важных аспектов расчета доступной полосы пропускания является назначение полосы пропускания для разных CT. Доля пропускной способности звена, которую может занимать данный CT (или группа CT), называется *ограничением по полосе пропускания BC (bandwidth constraint)*. В документе RFC 3564 определено понятие *модели ограничений по полосе пропускания*, отражающее взаимосвязь между классами CT и ограничениями BC.

Наиболее понятная интуитивно модель ограничений по полосе пропускания ставит в соответствие одному BC один CT. Она называется *моделью максимального назначения MAM (Maximum Allocation Model)*. Согласно модели MAM пропускания способность звена просто распределяется между разными CT без возможности распределения неиспользуемой полосы пропускания, так что эта полоса может непроизводительно простаивать вместо того, чтобы использоваться для других CT. Рассмотрим сеть, показанную на рис. 11.7. Все звенья имеют пропускную способность 10 Мбит/с, а назначение полос пропускания: для CT1 — 1 Мбит/с и для CT0 — 8 Мбит/с. Разберем этот рисунок более подробно.

Пусть на входе имеется поток данных 10 Мбит/с. Входной поток содержит только данные и имеет класс, соответствующий CT0. На рис. 11.7. есть два маршрута от LSR1 до LSR3 с пропускной способностью 10 Мбит/с каждый. Из этих 10 Мбит/с в соответствии с моделью и рассматриваемым на рисунке сегментом сети по 1 Мбит/с резервировано для потоков с приоритетом CT1 (например речь). Таким образом, на маршруте 1 для входного потока есть свободная полоса в 9 Мбит/с. Оставшиеся 1 Мбит/с вынуждены пойти по маршруту 2. Следовательно, на первом маршруте остается 1 Мбит/с для потока класса CT1 и 0 Мбит/с для потоков класса CT0, а на втором маршруте — 1 Мбит/с для CT1 и 8 Мбит/с (т.к. 1 Мбит/с уже занят) для CT0.

Даже при отсутствии речевого трафика обслуживающему его первому LSP полоса пропускания доступна на всех звеньях, составляющих кратчайший маршрут, но эта полоса пропускания не может использоваться другим LSP для трафика данных. Поэтому второй LSP трафика данных вынужден идти по неоптимальному маршруту, даже несмотря на то, что на кратчайшем маршруте необходимая полоса пропускания имеется. С другой стороны, достоинством модели MAM является то, что она обеспечивает полную развязку трактов LSP, переносящих трафик от разных CT, в связи с чем нет необходимости задавать приоритеты. Если в сети, показанной на рис. 11.7, администратор захочет создать LSP речевого трафика, то наличие необходимых для этого ресурсов будет гарантировано, и не понадобится вытеснять с помощью приоритетов LSP трафика данных, забирая у них ресурсы.

Доступная полоса пропускания для модели MAM рассчитывается аналогично тому, как это делается для TE, за исключением того, что расчет выполняется для каждого класса отдельно. Для расчета полосы пропускания, доступной *CTn* с приоритетом *m*, необходимо вычесть из полосы пропускания, назначенной для CT*n*, сумму полос, назначенных для трактов LSP с CT*n* для всех уровней приоритета, которые выше или равны *m.*

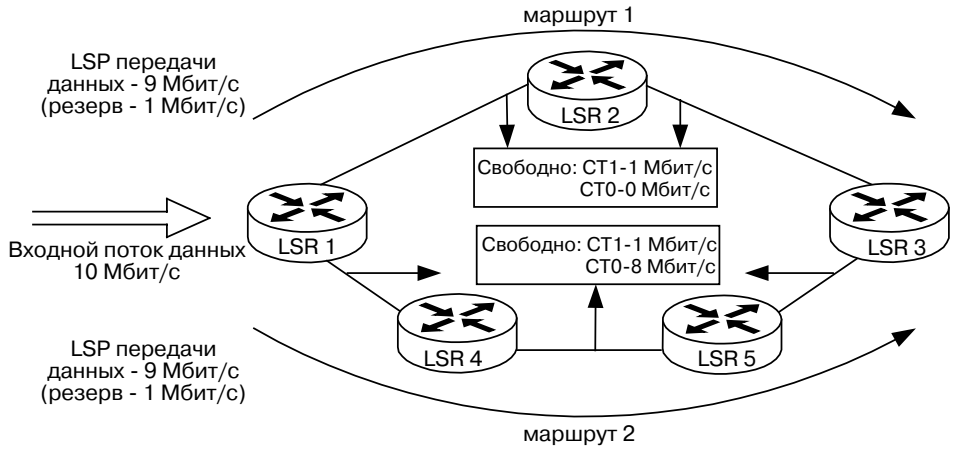


Рис. 11.7. Пример работы модели MAM

Модель назначения полосы пропускания, называемая *моделью матрешек RDM (Russian Dolls Model)*, улучшает эффективность использования пропускной способности звеньев по сравнению с моделью MAM благодаря тому, что позволяет классам CT совместно использовать полосу пропускания. В этой модели по-прежнему CT7 — это трафик с самыми жесткими требованиями к QoS, а CT0 — это трафик негарантированного обслуживания. Степень совместного использования пропускной способности варьируется в широком спектре, ограниченном двумя крайними случаями. На одном конце спектра мы имеем BC7 — фиксированная доля пропускной способности звена, которая резервируется только для трафика CT7. На другом конце спектра BC0 предоставляет полную пропускную способность звена, которая совместно используется всеми CT. В промежутке между этими двумя крайними значениями существуют варианты разной степени совместного использования: BC6 размещает трафик CT7 и CT6, BC5 — CT7, CT6 и CT5, и т.д. Эта модель напоминает принцип известных во всем мире русских матрешек, когда самая большая матрешка (BC0) содержит внутри себя матрешку меньшего размера (BC1), внутри которой находится еще меньшая матрешка (BC2) и т.д.

Преимущество модели RDM перед моделью MAM состоит в том, что она обеспечивает эффективное использование пропускной способности звеньев благодаря механизму совместного использования ресурсов. Рассмотрим все ту же сеть, показанную на рис. 11.7, которая передает речевой трафик и трафик данных. Полная доступная пропускная способность каждого звена по-прежнему составляет 10 Мбит/с. Для BC1 выделяется 1 Мбит/с, а для BC0 — 10 Мбит/с. Это означает, что каждое звено может передавать от 0 до 1 Мбит/с речевого трафика и использовать остальную часть своей пропускной способности для передачи данных. Если предположить, что по маршруту LSR1- LSR2- LSR3 уже организован LSP передачи данных, то в отсутствие речевого трафика может быть организован второй LSP передачи данных, которому будет предоставлена не используемая полоса пропускания, назначавшаяся для речевого трафика. Другим ценным свойством этой модели является дешевое выделение избыточных ресурсов для трафика реального времени. Поскольку «дополнительная» полоса пропускания может использоваться трафиком других типов, выделение ее для трафика реального времени не повлияет на общую пропускную способность сети.

Недостатком же модели RDM по сравнению с моделью MAM является то, что отсутствует развязка разных CT, и поэтому должен использоваться механизм приоритетного вытеснения для того, чтобы соответствующему CT в любом случае гарантировалась его доля пропускной способности. Если в той же сети, показанной на рис. 11.7, будут организованы два LSP для трафика данных, после чего администратор захочет создать LSP для трафика речи по тому же кратчайшему маршруту LSR1- LSR2- LSR3, он обнаружит, что для речевого трафика нет ресурсов. Необходимо, чтобы один из двух LSP с трафиком данных был вытеснен «речевым» LSP, т.к. в противном случае для речевого трафика полоса пропускания не сможет быть гарантирована. Трафик данных вытесненного LSP пойдет по более длинному маршруту LSR1 — LSR4 — LSR5 — LSR3. Для этого трактам LSP с трафиком речи и LSP с трафиком данных должны назначаться разные приоритеты, поскольку они используют ресурсы полосы пропускания совместно.

Расчет доступной полосы пропускания для модели RDM немного сложнее, чем для модели MAM, так как здесь надо принимать во внимание тракты LSP с несколькими уровнями приоритета и от всех CT, которые совместно используют одно и тоже BC. Например, доступная полоса пропускания для LSP от CT0 с приоритетом *p* равна BC0 минус сумма полосы пропускания, назначенных для всех LSP от всех CT с приоритетом не ниже *p*.

Различия между моделями MAM и RDM сведены в табл. 11.4.

Таблица 11.4. Сравнение моделей MAM и RDM

MAM	RDM
Одному BC ставится в соответствие один CT; модель проста для понимания и управления	Одному BC ставится в соответствие один или несколько CT, модель менее понятна на интуитивном уровне
Обеспечивает развязку разных CT и гарантирует полосу пропускания разным CT без необходимости использовать механизм приоритетного вытеснения	Отсутствует развязка разных CT; для того чтобы гарантировать полосы пропускания разным CT, требуется использовать механизм приоритетного вытеснения
Пропускная способность может использоваться неэффективно	Эффективное использование пропускной способности
Полезна в тех сетях, где исключено приоритетное вытеснение	Не рекомендуется в тех сетях, где исключено приоритетное вытеснение

Ясно, что модель BC играет решающую роль в определении полосы пропускания, которая доступна в звене каждому из TE-классов. Уведомления об используемой модели BC и о полосах пропускания, назначенных для каждого BC, передаются с помощью протоколов IGP в поле Bandwidth Constraints sub-TLV. Комитет IETF не обязывает использовать одну и ту же модель BC во всех звеньях сети. Однако такую сеть, в которой используется единая модель ограничений по полосе пропускания, легче конфигурировать, обслуживать и эксплуатировать.

11.9. MPLS и QoS

Вопросы обеспечения телекоммуникационными технологиями качества обслуживания восходят еще к работе [117], а сегодня они стали ключевыми и именно поэтому неоднократно обсуждались в предыдущих главах. Обзору перспектив развития работ в этой области посвящен данный параграф, завершающий главу 11 и всю книгу в целом.

Рассмотренная выше модель DiffServ тоже является одним из механизмов обеспечения качества обслуживания, но т.к. проблемы инжиниринга трафика и качества обслуживания в MPLS принципиально разделены, выше она рассматривалась именно в контексте MPLS TE. А сейчас мы сосредоточим внимание на MPLS QoS.

Приложение MPLS QoS родилось из слияния концепций Diff-Serv и MPLS. Заголовок метки MPLS содержит трехбитовое поле, которое первоначально даже называлось полем CoS, но позже было переименовано в поле Exp (Experimental), что отразило экспериментальный характер вариантов его использования.

Исследовательская работа над MPLS QoS отражена в проекте документа «MPLS Support of Differentiated Services». В рамках этой работы предложена новая концепция — *агрегат поведения BA*, представляющий собой набор всех IP-пакетов, которые проходят по одному маршруту и требуют одного и того же поведения

Diff-Serv. Разработана спецификация, позволяющая отображать конфигурацию Diff-Serv-агрегатов BA на LSP и наилучшим образом выполнять задачи Diff-Serv в домене MPLS. В этой спецификации входной LSR классифицирует и маркирует DSCP, которой соответствует BA. В транзитных маршрутизаторах LSR кодовая точка DSCP используется для выбора PHB, конфигурированного для определенного трафика. Кроме того, спецификация способствует сохранению пространства меток и уменьшает общее количество операций назначения и уничтожения меток, необходимых для сигнализации, путем использования LSP для данного FEC. Спецификация поддерживает как формат IPv4, так и формат IPv6 для одноадресного трафика. В работе MPLS QoS определены два типа Diff-Serv LSP: тип E-LSP и тип L-LSP.

E-LSP (EXP-Inferred-PSC LSP) может транспортировать несколько *упорядоченных агрегатов OA*. Поле EXP в заголовке метки MPLS используется LSR для назначения PHB, применяемого к пакету. E-LSP может поддерживать до восьми агрегатов BA данного FEC, независимо от количества агрегатов OA, охватываемых этими агрегатами BA. Мэппинг из поля EXP в его значение PHB для данного E-LSP может явно сигнализироваться в момент назначения метки или может быть определен предварительно.

L-LSP (Label-Only-Inferred-PSC LSP) может транспортировать только один OA. LSR маршрутизирует пакет, исходя из значения метки MPLS, но приоритет для отбрасывания пакета может быть дополнительно обозначен битами EXP в заголовке метки. Когда промежуточный заголовок не используется, как в ATM и FR, сведения о приоритете отбрасывания для LSR передаются внутри заголовка уровня 2 с использованием специальных полей приоритета отбрасывания на уровне звена данных. Для ATM, например, используется поле CLP.

Еще одно определяющее QoS направление связано с восстановлением LSP при сетевых сбоях. MPLS предусматривает много возможных вариантов восстановления. В основном, это концепции со сложными архитектурными и реализационными подходами, которые в настоящее время горячо обсуждаются рабочей группой IETF. Рассматриваются три исходных сценария восстановления LSP: локальное восстановление, сквозное восстановление и быстрая ремаршрутизация. Работа над приложением восстановления тракта LSP энергично продолжается, и уже написаны и обсуждаются многочисленные информационные документы, связанные со сценариями неисправностей и восстановления. Один документ — «A Method for MPLS LSP Fast-Reroute Using RSVP Detours» — вводит механизм создания резервных туннелей RSVP-TE. Приложение для этого механизма особенно удачно работает в крупномасштабных сетях. Предложение включает в себя два новых объекта RSVP-TE, которые позволяют маршрутизаторам LSR с RSVP-TE при необходимости создавать маршруты с обходом вокруг определенных звеньев и узлов.

Эта функциональная возможность позволяет быстро и автоматически ремонтировать LSP, пока данные успешно передаются по предварительно рассчитанным и предварительно организованным обходным маршрутам. Главный принцип этого предложения — потери пакетов сводятся к минимуму.

Другой документ — «A Method for Setting an Alternative Label Switched Paths to Handle Fast Reroute» — предлагает метод создания альтернативных LSP для быстрой ремаршрутизации трафика, когда первичный LSP выходит из строя. При всех способах восстановления тракта предварительно установленные альтернативные тракты могут использоваться в случаях, когда потеря пакетов из-за отказа LSP нежелательна., Поскольку прогнозировать, где может возникнуть неисправность LSP, трудно, приложения восстановления LSP применяют сложные вычисления и дополнительную сигнализацию, чтобы создать альтернативные маршруты для защиты всего тракта. Кроме того, в рамках IETF проходят обсуждения многих других новых приложений MPLS. Одно из них включает в себя использование иерархического стека меток.

Еще одно направление в разработке приложений MPLS — обеспечение поддержки трафика между автономными системами (Inter-AS). Необходимы механизмы, с помощью которых провайдеры услуг могут эффективно пересылать трафик MPLS друг другу и решать такие вопросы, как начисление платы и поддержка соглашений SLA. Продолжается работа и в области процедур защиты. По мере развития архитектуры MPLS, несомненно, последуют новые приложения.

Скорость развития технологии MPLS существенно усложнила работу авторов. Когда они начинали эту книгу, казалось достаточным описать метки и протокол LDP. В процессе работы этот список пополнился резервированием ресурсов, расширениями протоколов маршрутизации, туннелями, VPN, инжинирингом трафика и пр. Заканчивая работу над последней главой, авторы осознали необходимость хотя бы упомянуть в ней то, что активно обсуждается в контексте MPLS *на момент написания,* хотя, возможно, многое упомянутое заслуживает отдельных глав. А возможно, что *на момент прочтения* многое из упомянутого в этой главе утратит актуальность или, наоборот, разовьется гораздо глубже, чем здесь изложено.

Скорость развития MPLS сегодня выше, чем темпы работы типографии. Тем не менее, эта книга даст возможность разобраться с основами MPLS, с ее протоколами сигнализации и маршрутизации, со многими тонкостями этой чрезвычайно перспективной технологии. И может быть, в соответствии с духом наступающей эпохи NGN, построенной на базе MPLS, окажется возможным сетевое издание этой книги, регулярно обновляемое на сайте www.niits.ru.

Авторы обещают подумать над этим.

Литература

1. Alwayn, Vivek. Advanced MPLS Design and Implementation. Indianapolis, IN: Cisco Press, 2001.
2. Armitage Grenville. MPLS: the magic behind the myths, IEEE Communications Magazine, vol. 38, no. 1, January 2000.
3. Armitage Grenville. Quality of Service in IP Networks. — Macmillan Technical Publishing, 2000.
4. Arvidsson Ake, Krzesinski Antony. The Design of Optimal Multi-Service MPLS Network // Telektronik 2/3.-2001.
5. Ash G.R. Dynamic Routing in Telecommunications Networks. McGraw Hill, 1998.
6. Awduche D. MPLS and Traffic Engineering in IP Networks. IEEE Communications Magazine, vol. 37, December 1999.
7. Барсков А.Г. VPN — старые принципы, новые технологии//Сети и системы связи, №6(112). — 2004.
8. Belloni A. Alcatel 5620 IP/MPLS Data Network Management. Alcatel Telecommunication Review — 3rd Quarter 2002.
9. Black, Ulyess. MPLS and Label Switching Networks. Upper Saddle River, NJ: Prentice Hall PTR, 2001.
10. Bouillet E., Mitra D., and Ramakrishnan K.G. The Structure and management of Service Level Agreements in Networks. IEEE JSAC Vol. 20, No. 4, May 2002.
11. Chen T.M., Oh T.H.. Reliable services in MPLS. IEEE Communications Magazine, December 1999.
12. Davidson J., Peters J. Voice Over IP Fundamentals. — Cisco Press, 2000.
13. Davie B., Rekhter Y. MPLS, Technology and Applications. Morgan Kaufmann Publishers, 2000.
14. Douskalis B. Putting VoIP to Work: Softswitch Network Design and Testing. Upper Saddle River, NJ: Prentice Hall PTR, 2002.
15. Fortz B. and Thorup M. Optimizing OSPF/IS-IS Weights in a Changing World// IEEE Journal on Selected Areas in Communications, Vol. 20, No. 4, May 2002.
16. Garcia J.M., Rachdi A., Brun O. Optimal LSP Placement with QoS Constraints in DiffServ/MPLS Networks/ITC 18 / Charzinski J., Lehnert R., and Tran-Gia P. (Editors), Elsevier Science B.V., 2003.
17. Ghanwani A., Jamoussi B., Fedyk D., Ashwood-Smith P., Li L., and Feldman N. Traffic Engineering Standards in IP Networks Using MPLS. IEEE Communications Magazine, vol. 37, December 1999.
18. Гольдштейн А.Б.. Проблемы перехода к мультисервисным сетям// Вестник связи.-2002- №12.
19. Goldstein A., Yanovsky G. Traffic Engineering in MPLS Tunnels//In International Conference on “NExt Generation Teletraffic and Wired/Wireless Advanced Networking (NEW2AN’04)”, February 02-06, 2004.
20. Гольдштейн А.Б. Механизм эффективного туннелирования в сети MPLS// Вестник связи.-2004- №2.
21. Гольдштейн Б.С., Пинчук А.В., Суховицкий А.Л. IP-телефония. М.: Радио и связь, 2001.
22. Гольдштейн Б.С. Протоколы сети доступа. Том 2. М.: Радио и связь, 1999.
23. Гольдштейн Б.С., Ехриель И.М., Рерле Р.Д. Стек ОКС7. Подсистема МТР. Справочник//М.: Радио и связь-2003.
24. Гольдштейн Б.С., Ехриель И.М., Кадыков В.Б., Рерле Р.Д. Протоколы V5.1 и V5.2. Справочник//СПб.: BHV-2003.

25. Гольдштейн Б.С., Ехриель И.М., Рерле Р.Д. Стек ОКС7. Подсистема ISUP. Справочник// СПб.: BHV-2003.
26. Гольдштейн Б.С., Сибирякова Н.Г., Соколов А.В. Сигнализация R1.5. Справочник// СПб.: BHV-2004.
27. Goralski Walter J., Kolon Matthew C. IP telephony/The McGraw-Hill Co. Inc., 2000.
28. Gray Eric. MPLS: Implementing the Technology. Boston: Addison-Wesley, 2001.
29. Hendling K., Franzl G., Statovci-Halimi B., Halimi A. Residual Network and Link Capacity Weighting for Efficient Traffic Engineering in MPLS Networks. ITC 18 / Charzinski J., Lehnert R., and Tran-Gia P. (Editors), Elsevier Science B.V., 2003.
30. Hersent O., Gurle D., Petit Jean-Pierre. IP Telephony: Packet-Based Multimedia Communications Systems.- Addison-Wesley Pub Co, 2000.
31. Hoebeke R., Aissaoui M., Nguyen T. MPLS: Adding Value to Networking. Alcatel Telecommunication Review — 3rd Quarter 2002.
32. Hoey G.Van, Van S. den Bosch, P. deLa Vallee-Poussin, Degrande N., and H. De Neve. An Integrated Approach to MPLS Traffic Engineering over Automatically Switched Transport Networks. Internet Traffic Engineering. Vol. 13, No.1, January-February 2002.
33. Kamei S., Kimura T. Evaluation of Routing Algorithms and Network Topologies for MPLS Traffic Engineering. Proceedings of GLOBECOM'01, vol. 1, November 2001.
34. Kar K., Kodialam M., Lakshman T.V. Minimum Interference Routing of Bandwidth Guaranteed Tunnels with MPLS Traffic Engineering Applications. IEEE Journal on Selected Areas in Communications, vol. 18, December 2000.
35. Kodialam M., Lakshman T. Minimum interference routing with applications to MPLS traffic engineering // In IEEE Infocom 00, Vol.2, IEEE, Tel Aviv, Israel, 2000.
36. Кох Р., Яновский Г. Эволюция и конвергенция в электросвязи. М.: Радио и связь, 2001.
37. Kohler Stefan, Binzenhofer Andreas. MPLS Traffic Engineering in OSPF Networks — A combined approach. ITC 18 / Charzinski J., Lehnert R., and Tran-Gia P. (Editors), 2003 Elsevier Science B.V.
38. Lawrence J. Design multiprotocol label switching networks. IEEE Communications Magazine, July. 2001.
39. Li Tony. MPLS and the Evolving Internet Architecture. IEEE Communications Magazine, December 1999. pp. 38-41.
40. Minoli D. Voice over MPLS. Planning and Designing Networks. McGraw-Hill, 2002.
41. Mizuhara B., Kazutaka O., Katsumata K., Yamada K. MPLS Technologies for IP Networking Solution. NEC. Res. & Develop., Vol.42, No.2, April 2001.
42. Moy J. OSPF — Anatomy of an Internet Routing Protocol. Addison-Wesley, 1998.
43. Мюнх Б., Скворцова С. Сигнализация в сетях IP-телефонии. — Части I, II / Сети и системы связи, №13(47), 14(48). — 1999.
44. Pepelnjak Ivan and Guichard Jim. MPLS and VPN Architectures: A Practical Guide to Understanding, Designing, and Deploying MPLS and MPLS-enabled VPNs (Cisco Networking Fundamentals). Indianapolis, IN: Cisco Press, 2001.
45. RFC 1058 Routing Information Protocol (RIP). C. Hedrick. June 1988
46. RFC 1131. The OSPF Specification — Open Shortest Path First. J. Moy. October 1989.
47. RFC 1142. OSI IS-IS Intra-domain Routing Protocol. D. Oran, Editor. February 1990.

48. RFC 1191. Path MTU Discovery. J. Mogul, S. Deering. November 1990.
49. RFC 1195. Use of OSI IS-IS for Routing in TCP/IP and Dual Environments. R.W. Callon. December 1990.
50. RFC 1246. Experience with the OSPF protocol. J. Moy, Editor. July 1991.
51. RFC 1247. OSPF Version 2. J. Moy. July 1991.
52. RFC 1248. OSPF Version 2 Management Information Base. F. Baker, R. Coltun. July 1991.
53. RFC 1252. OSPF Version 2 Management Information Base. F. Baker, R. Coltun. August 1991.
54. RFC 1253. OSPF Version 2 Management Information Base. F. Baker, R. Coltun. August 1991.
55. RFC 1254. Gateway Congestion Control Survey. A. Mankin, K. Ramakrishnan. August 1991.
56. RFC 1321. The MD5 Message-Digest Algorithm. R. Rivest. April 1992.
57. RFC 1364 BGP OSPF Interaction. K. Varadhan. September 1992.
58. RFC 1370. Applicability Statement for OSPF. Internet Architecture Board. Lyman Chapin. October 1992.
59. RFC 1403. BGP OSPF Interaction. K. Varadhan. January 1993.
60. RFC 1483. Multiprotocol Encapsulation over ATM Adaptation Layer 5. Juha Heinanen. July 1993.
61. RFC 1577. Classical IP and ARP over ATM. M. Laubach. January 1994. (Развит в RFC 2225).
62. RFC 1582. Extensions to RIP to Support Demand Circuits. G. Meyer. February 1994.
63. RFC 1583. OSPF Version 2. J. Moy. March 1994. (Развитие RFC 1247. Развит в RFC 2178).
64. RFC 1584. Multicast Extensions to OSPF. J. Moy. March 1994.
65. RFC 1586. Guidelines for Running OSPF Over Frame Relay Networks. O. deSouza, M. Rodrigues. March 1994.
66. RFC 1587. The OSPF NSSA Option. R. Coltun, V. Fuller. March 1994.
67. RFC 1654 A Border Gateway Protocol 4 (BGP-4). Y. Rekhter, T. Li. July 1994.
68. RFC 1745 BGP4/IDRP for IP — OSPF Interaction. K. Varadhan, S. Hares, Y. Rekhter. December 1994.
69. RFC 1765 OSPF Database Overflow. J. Moy. March 1995.
70. RFC 1771. A Border Gateway Protocol 4 (BGP-4). Y. Rekhter, T. Li. March 1995. (Развитие RFC 1654).
71. RFC 1773 Experience with the BGP-4 protocol. P. Traina. March 1995.
72. RFC 1774 BGP-4 Protocol Analysis. P. Traina, Editor. March 1995.
73. RFC 1793 Extending OSPF to Support Demand Circuits. J. Moy. April 1995.
74. RFC 1930 Guidelines for creation, selection, and registration of an Autonomous System (AS). J. Hawkinson, T. Bates. March 1996.
75. RFC 1953. Ipsilon Flow Management Protocol Specification for IPv4. Version 1.0. P. Newman, W. Edwards, R. Hinden, E. Hoffman, F. Ching Liaw, T. Lyon, G. Minshall. May 1996.
76. RFC 1966. BGP Route Reflection. An alternative to full mesh IBGP. T. Bates, R. Chandra. June 1996. (Развит в RFC 2796).
77. RFC 1987. Ipsilon's General Switch Management Protocol Specification. Version 1.1. P. Newman, W. Edwards, R. Hinden, E. Hoffman, F. Ching Liaw, T. Lyon, G. Minshall. August 1996. (Развит в RFC 2297).

78. RFC 1997. BGP Communities Attribute. R. Chandra, P. Traina, T. Li. August 1996. RFC 2113. IP Router Alert Option. D. Katz. February 1997.
79. RFC 2178 Protocol OSPFv2. J. Moy. July 1997.
80. RFC 2205. Resource ReSerVation Protocol (RSVP) — Version 1 Functional Specification. R. Braden, Ed., L. Zhang, S. Berson, S. Herzog, S. Jamin. September 1997. (Развит в RFC 2750).
81. RFC 2208. Resource ReSerVation Protocol (RSVP) Version 1 Applicability Statement. Some Guidelines on Deployment. A. Mankin, Ed., F. Baker, B. Braden, S. Bradner, M. O`Dell, A. Romanow, A. Weinrib, L. Zhang. September 1997.
82. RFC 2209. Resource ReSerVation Protocol (RSVP) — Version 1 Message Processing Rules. R. Braden, L. Zhang. September 1997.
83. RFC 2210. The Use of RSVP with IETF Integrated Services. J. Wroclawski. September 1997.
84. RFC 2283. Multiprotocol Extensions for BGP-4. T. Bates, R. Chandra, D. Katz, Y. Rekhter. February 1998. (Развит в RFC 2858).
85. RFC 2328. OSPF Version 2. J. Moy. April 1998.
86. RFC 2370. The OSPF Opaque LSA Option. R. Coltun. July 1998.
87. RFC 2385. Protection of BGP Sessions via the TCP MD5 Signature Option. A. Heffernan. August 1998.
88. RFC 2547. BGP/MPLS VPNs. E. Rosen, Y. Rekhter. March 1999.
89. RFC 2571. An Architecture for Describing SNMP Management Frameworks. D. Harrington, R. Presuhn, B. Wijnen. April 1999. April 1999.
90. FRC 2578 Structure of Management Information Version 2 (SMIv2). Editors of this version: K. McCloghrie, D. Perkins, J. Schoenwaelder. Authors of previous version: J. Case, K. McCloghrie, M. Rose, S. Waldbusser. April 1999.
91. RFC 2676 QoS Routing Mechanisms and OSPF Extensions. G. Apostolopoulos, S. Kama, D. Williams, R. Guerin, A. Orda, T. Przygienda. August 1999.
92. RFC 2684. Multiprotocol Encapsulation over ATM Adaptation Layer 5. D. Grossman, J. Heinanen. September 1999. (Развитие RFC 1483).
93. RFC 2702. Requirements for Traffic Engineering Over MPLS. Awduche D., Malcolm J., Agogbua J., O'Dell M., McManus J., September 1999.
94. RFC 2740. OSPF for IPv6. R. Coltun, D. Ferguson, J. Moy. December 1999.
95. RFC 2749. COPS usage for RSVP. S. Herzog, Ed., J. Boyle, R. Cohen, D. Durham, R. Rajan, A.Sastry. January 2000.
96. RFC 2751. Signaled Preemption Priority Policy Element. S. Herzog. January 2000.
97. RFC 2844. OSPF over ATM and Proxy-PAR. T. Przygienda, P. Droz, R. Haas. May 2000.
98. RFC 2858. Multiprotocol Extensions for BGP-4. T. Bates, Y. Rekhter, R. Chandra, D. Katz. June 2000. (Развитие RFC 2283).
99. RFC 2917. A Core MPLS IP VPN Architecture. K. Muthukrishnan, A. Malis. September 2000.
100. RFC 2961. RSVP Refresh Overhead Reduction Extensions. L. Berger, D. Gan, G. Swallow, P. Pan, F. Tommasi, S. Molendini. April 2001.
101. RFC 2983. Differentiated Services and Tunnels. D. Black. October 2000 .
102. RFC 3031. Multiprotocol Label Switching Architecture. E. Rosen, A. Viswanathan, R. Callon. January 2001.
103. RFC 3032. MPLS Label Stack Encoding. E. Rosen, D. Tappan, G. Fedorkow, Y. Rekhter, D. Farinacci, T. Li, A. Conta. January 2001.

104. RFC 3033. The Assignment of the Information Field and Protocol Identifier in the Q.2941 Generic Identifier and Q.2957 User-to-user Signaling for the Internet Protocol. M.Suzuki. January 2001.
105. RFC 3034. Use of Label Switching on Frame Relay Networks Specification. A. Conta, P. Doolan, A. Malis. January 2001.
106. RFC 3035. MPLS using LDP and ATM VC Switching. B. Davie, J. Lawrence, K. McCloghrie, E. Rosen, G. Swallow, Y. Rekhter, P. Doolan. January 2001.
107. RFC 3036. LDP Specification. L. Andersson, P. Doolan, N. Feldman, A. Fredette, B. Thomas. January 2001.
108. RFC 3037. LDP Applicability. B. Thomas, E. Gray. January 2001.
109. RFC 3038. VCID Notification over ATM link for LDP. K. Nagami, Y. Katsube, N. Demizu, H. Esaki, P. Doolan. January 2001.
110. RFC 3063. MPLS Loop Prevention Mechanism. Y. Ohba, Y. Katsube, E. Rosen, P. Doolan. February 2001.
111. RFC 3101. The OSPF Not-So-Stubby Area (NSSA) Option. P. Murphy. January 2003.
112. RFC 3107. Carrying Label Information in BGP-4. Y. Rekhter, E. Rosen. May 2001.
113. RFC 3209. RSVP-TE: Extensions to RSVP for LSP Tunnels. D. Awduche, L. Berger, D. Gan, T. Li, V. Srinivasan, G. Swallow. December 2001.
114. RFC 3232. Assigned Numbers: RFC1700 is Replaced by an On-line Database. Edit by J. Reynolds. January 2002.
115. RFC 3270 Multi-Protocol Label Switching (MPLS) Support of Differentiated Services. F. Le Faucheur, Editor, L. Wu, B. Davie, S. Davari, P. Vaananen, R. Krishnan, P. Cheval, J. Heinanen. May 2002.
116. RFC 3468. The Multiprotocol Label Switching (MPLS) Working Group decision on MPLS signaling protocols. L. Andersson, G. Swallow. February 2003.
117. RFC 3471. Generalized Multi-Protocol Label Switching (GMPLS) Signaling Functional Description. L. Berger, Editor. January 2003.
118. RFC 3472. Generalized Multi-Protocol Label Switching (GMPLS) Signaling Constraint-based Routed Label Distribution Protocol (CR-LDP) Extensions. P. Ashwood-Smith, Editor, L. Berger, Editor. January 2003.
119. RFC 3473. Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Extensions. L. Berger, Editor. January 2003.
120. RFC 3479. Fault Tolerance for the Label Distribution Protocol (LDP). A. Farrel, Ed. February 2003.
121. Rouskas George N., Jackson Laura E. Optimal Granularity of MPLS Tunnels, ITC 18 / Charzinski J., Lehnert R., and Tran-Gia P. (Editors), 2003 Elsevier Science B.V.
122. Swallow George. MPLS Advantages for Traffic Engineering. IEEE Communications Magazine, December 1999, pp. 54-57.
123. Vismanathan A., Feldman N., Wang Zh., Callon R. Evolution of Multiprotocol Label Switching. IEEE Communications Magazine, May 1998.
124. Wang Z. Internet QoS: Architectures and Mechanisms for Quality of Service. The Morgan Kaufmann Series in Networking, Morgan Kaufmann Publishers, March 2001.
125. Xiao X., Hannan A., Bailey B., Ni L. Traffic Engineering with MPLS in the Internet, IEEE Network, March/April 2000.

Глоссарий

AAL (ATM adaptation layer) — уровень ATM-адаптации, обеспечивающий согласование общих для всех услуг функций уровня ATM, расположенного под AAL, с функциями расположенного над AAL верхнего уровня.

ABR (Area Border Router) — пограничный маршрутизатор, подключенный к нескольким областям протокола OSPF.

AC (Admission Control) — механизма управления доступом.

Alias — псевдоним.

ARIS (Aggregate Route-based IP Switching) — IP-коммутация на базе объединенных маршрутов.

ARP (Address Resolution Protocol) — протокол преобразования адресов, использующийся для привязки сетевых адресов к адресам физического уровня.

AS (Autonomous System) — автономная система — группа маршрутизаторов с общим для них административным управлением.

ASBR (Autonomous System Border Router) — пограничный OSPF-маршрутизатор автономной системы, расположенный на границе двух доменов и преобразующий маршруты других протоколов маршрутизации в маршруты OSPF.

ASN.1 (Abstract Syntax Notation One) — абстрактная синтаксическая нотация версия 1. Язык модели OSI для описания типов данных независимо от структуры компьютеров и методов их представления. Определен стандартом ISO 8824.

AS-путь (AS path) — список автономных систем, пересекаемых BGP-маршрутом; как правило, используется для выявления петель маршрутизации.

ATM (Asynchronous Transfer Mode) — асинхронный режим переноса информации. Для поддержки этого режима служит протокол, содержащий два функциональных уровня: уровень ATM, реализующий функции переноса ATM-конвертов, общие для всех услуг, и уровень ATM-адаптации (уровень AAL), который предоставляет верхним уровням функции, зависящие от реализуемых этими верхними уровнями услуг.

ATM Forum (Форум ATM) — международная организация, созданная в 1991 г. Cisco Systems, NET/ADAPTIVE, Northern Telecom и Sprint для разработки и внедрения стандартов в области ATM-технологии.

BDR (Backup Designated Rouler) — резервный назначенный OSPF-маршрутизатор.

BGP (Border Gateway Protocol) — протокол пограничного шлюза, пришедший на смену EGP и регламентирующий обмен информацией с другими BGP-системами.

CBWFQ (Class Based WFQ) — технология WFQ, действие которой распространяется на несколько классов трафика с совместным доступом к ресурсам.

CE-маршрутизатор — периферийный маршрутизатор сети заказчика; связан с периферийным маршрутизатором сети провайдера (PE-маршрутизатором) и является для него одноранговым.

CIDR (Classless InterDomain Routing) — бесклассовая междоменовая маршрутизация; определяет адрес для следующей пересылки пакета на основании анализа длины префикса и адреса конечного пункта назначения.

CLNP (Connectionless Network Protocol) — сетевой протокол, не ориентированный на соединение.

CLNS (Connectionless Network Service) — сетевая услуга без создания соединения. Информация о маршрутах для протокола CLNS предоставляется протоколом маршрутизации IS-IS.

CoS (Class of Service) — класс обслуживания, характеристика, позволяющая определить, как протокол верхнего уровня использует протокол нижнего уровня для обработки его сообщений.

CSR (Cell Switch Router) — маршрутизатор коммутации конвертов ATM.

DiffServ (Differentiated Services) — модель дифференцированного обслуживания, при которой пользовательский трафик разделяется на классы с разной дисциплиной обслуживания. Одна из моделей обеспечения QoS.

DLCI (Data-Link Connection Identifier) — идентификатор соединения уровня 2, который определяет PVC или SVC в сетях Frame Relay. В базовой спецификации Frame Relay DLCI-идентификаторы являются локальными (чтобы указать одно и то же соединение, подключенные устройства могут использовать разные значения DLCI), а в расширенной спецификации LMI — глобальными (указывают отдельные конечные устройства).

DMZ (DeMilitarized Zone) — демилитаризованная зона — буферная зона между «надежной» и «ненадежной» частями сети.

DR (Designated Router) — назначенный OSPF-маршрутизатор, ответственный за лавинное распространение информации о маршрутах в вещательном сегменте сети и за извещение об этом сегменте устройств оставшейся части сети.

DWDM (Dense Wave Division Multiplexing) — компактное мультиплексирование с разделением по длине волны.

eBGP (Exterior BGP) — внешний протокол пограничного шлюза; по этому протоколу взаимодействуют два BGP-маршрутизатора, принадлежащих разным автономным системам.

EGP (Exterior Gateway Protocol) — протокол внешнего шлюза; предназначен для обмена между автономными системами большими объемами информации. Примерами протоколов маршрутизации внешнего шлюза являются EGP, BOP и IDRP.

ES-IS (End System-to-Intermediate System) — протокол обмена информацией о маршрутах между конечной и промежуточной системами, используемый протоколом CLNS для того, чтобы генерировать информацию, которой конечные системы обмениваются с маршрутизаторами.

FEC (Forwarding Equivalence Class) — класс эквивалентности пересылки — класс пакетов сетевого уровня, которые пересылаются в сети MPLS по одному и тому же LSP (или LSP-туннелю).

Flapping — перескакивание. Проблема маршрутизации, возникающая, когда маршрут между двумя узлами периодически изменяется (проходя то через один маршрутизатор, то через другой) вследствие некоторой сетевой проблемы, которая вызывает периодические сбои в работе интерфейса.

Flooding — лавинная рассылка. Способ рассылки пакетов, при котором полученный интерфейсом трафик пересылается всем другим интерфейсам этого устройства.

GMPLS (Generalized MPLS) — универсальная MPLS. Технология MPLS, доработанная для работы в гибридных или оптических сетях.

Hop-by-hop — способ транспортировки пакетов, при котором каждый сетевой узел самостоятельно принимает решение о дальнейшей переселки пакета.

HSRP (Hot Standby Router Protocol) — протокол маршрутизаторов Cisco, который предоставляет виртуальный IP-адрес, разделяемый между двумя маршрутизаторами. При выходе из строя одного маршрутизатора его место немедленно занимает другой маршрутизатор, который продолжает обработку трафика, поступающего на тот же виртуальный адрес.

iBGP (Interior BGP) — внутренний протокол пограничного шлюза. По этому протоколу взаимодействуют два BGP-маршрутизатора, принадлежащих одной и той же автономной системе.

ICMP (Internet Control Message Protocol) — протокол управляющих сообщений Internet.

IETF (Internet Engineering Task Force) — рабочая группа инженерных задач Internet. Организация, отвечающая за разработку протоколов сети Internet.

IFMP (Ipsilon Flow Management Protocol) — протокол, относящий трафик к тому или иному классу и помогающий создавать виртуальный канал.

IGMP (Internet Group Management Protocol) — межсетевой протокол управления группами.

IGP (Interior Gateway Protocol) — протокол маршрутизации внутреннего шлюза.

IntServ (Integrated Services) — модель интегрированного обслуживания, при которой пользовательское приложение может запрашивать резервирование сетевых ресурсов для передачи трафика. Одна из моделей обеспечения гарантированного качества обслуживания (QoS).

IPoATM — технология передачи IP-трафика поверх ATM-сети с целью обеспечения качества обслуживания.

IPNG (Internet Protocol Next Generation) —Internet-протокол следующего поколения.

IP Switching — одна из технологий-предшественниц MPLS, разработанная фирмой Ipsilon.

IS-IS (Intermediate System-to-Intermediate System) — протокол «промежуточная система—промежуточная система» маршрутизации внутри автономной системы.

ISP (Internet Service Provider) — провайдер услуг Интернет.

Label Stack — стек меток.

LER (Label Edge Router) — пограничный маршрутизатор домена.

LDP (Label Distribution Protocol) — протокол распределения меток, т.е. информации о привязке меток к FEC.

LIB — информационная база меток.

LSA (Link-State Advertisement) — извещение о состоянии каналов. Пакет, использующийся протоколом маршрутизации OSPF для распределения по сети информации о маршрутах.

LSP (Label Switching Path) — коммутируемый по меткам тракт.

LSP (Link-State Packet) — пакет с информацией о состоянии каналов. Пакет, использующийся протоколом маршрутизации IS-IS для распространения информации о состоянии каналов между маршрутизаторами сети.

LSR (Label Switching Router) — маршрутизатор сети MPLS.

MOSPF (Multicast Open Shortest Path First) — многоадресная рассылка с применением алгоритма первоочередного выбора кратчайших маршрутов.

MPLS (Multiprotocol Label Switching) — многопротокольная коммутация по меткам.

MTU (Maximum Transfer Unit) — максимально допустимый размер передаваемого пакета.

NHLFE (Next Hop Level Forwarding Entry) — указывает следующий участок, операцию, которая должна быть выполнена со стеком меток.

NHS (Next Hop Server) — сервер следующей пересылки.

NHRP (Next Hop Resolution Protocol) — протокол определения адреса следующей пересылки пакета, использующийся в сетях с поддержкой коммутируемых виртуальных каналов.

NSAP (Network Service Access Point) — точка доступа к сетевым услугам в CLNS-сети.

NSSA (Not-so-Stubby Area) — частично тупиковая область. Область протокола OSPF, в которую информация о внешних маршрутах не поступает, но может генерироваться внутри области.

OSI (Open System Interconnection) — взаимодействие открытых систем, международная программа стандартизации, которая создана ISO и ITU-T для разработки стандартов межсетевого обмена данными, способствующих функциональной совместимости оборудования разных производителей.

OSPF (Open Shortest Path First) — открытый протокол маршрутизации IGP по принципу «первым выбирается кратчайший путь», предложенный в качестве преемника RIP для Internet.

O-UNI (LDP Extensions for Optical User Network Interface (O-UNI) Signaling) — расширения протокола LDP для поддержки сигнализации в оптическом интерфейсе «пользователь-сеть».

QoS (Quality of Service) — качество обслуживания.

PE-маршрутизатор — периферийный маршрутизатор провайдера. Входит в сеть провайдера, связывается с CE-маршрутизатором и является для него одноранговым. PE-маршрутизаторы преобразуют адреса IPv4 в 12-байтовые адреса VPN-IPv4. PE-маршрутизаторы называются также Edge LSR.

PoP (Point of Presence) — точка входа в сеть.

PPP (Point-to-Point Protocol) — протокол «точка-точка».

P-маршрутизатор — маршрутизатор провайдера, т.е. магистральный маршрутизатор MPLS-VPN, выполняет коммутацию по меткам и является одноранговым для других P-маршрутизаторов и может непосредственно подключаться к PE-маршрутизатору.

RIP (Routing Information Protocol) — протокол маршрутизации внутреннего шлюза (IGP), поставляемый вместе с системами UNIX BSD.

RSVP (Resource reSerVation Protocol) — протокол резервирования ресурсов.

RTO (Round Trip Timeout) — лимит времени передачи подтверждения приема. Время, в течение которого EIGRP-маршрутизатор будет ожидать подтверждение доставки пакета, прежде чем предпринять какие-либо дальнейшие действия.

SIA (Stuck-in-Active) — постоянно активный маршрут. Маршрут протокола EIGRP, который находится в состоянии активного поиска более трех минут.

SHIM HEADER — заголовок-клин, вставляемый в пакет между заголовками уровня звена данных и сетевого уровня.

SNMP (Simple Network Management Protocol) — упрощенный протокол управления сетью.

SPF (Shortest Path First) — алгоритм выбора кратчайшего пути, используется протоколами маршрутизации IS-IS и OSPF для вычисления дерева кратчайших путей к каждой точке назначения в сети. (Алгоритм Дийкстры).

SVC (Switched Virtual Circuit) — коммутируемый виртуальный канал. Коммутируемый двухточечный канал передачи информации, наиболее часто применяемый в сетях ATM, однако используемый также в сетях Frame Relay и X.25.

TDP (Tag Distribution Protocol) — протокол распределения тегов.

TLV —тип-длина-значение. Схема кодирования большей части информации, переносимой в LDP-сообщениях.

TS (Tag Switching) — коммутация по тегам. Одна из технологий -предшествениц MPLS, разработанная фирмой Cisco.

TE (Traffic Engineering) — инжиниринг трафика. Управление трафиком с целью оптимизации загрузки сетевых ресурсов.

ToS (Type of Service) — тип обслуживания.

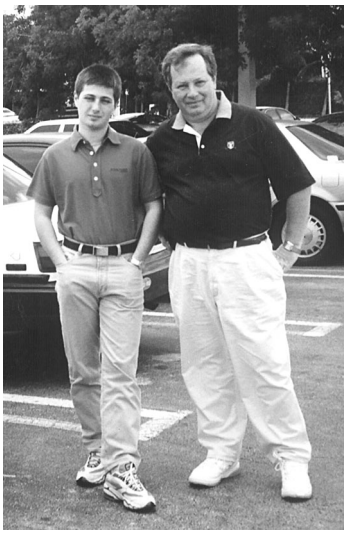
TTL (Time To Live) — «время жизни». Время или число пройденных маршрутизаторов, определяющее продолжительность существования пакета в сети; параметр TTL является средством предотвращения петель маршрутизации.

VCI (Virtual Circuit Identifier) — идентификатор виртуального канала. Пара VPI/VCI в заголовке ATM-конверта определяет соединение в сети ATM.

VPI (Virtual Path Identifier) — идентификатор виртуального тракта. Совместно с VCI определяет соединение в сети ATM.

WFQ (Weighted Fair Queing) — взвешенная недискриминационная организация очередей, способствующая предотвращению перегрузок.

Предметный указатель



Гольдштейн Борис Соломонович — доктор технических наук, профессор, заведующий кафедрой СПбГУТ им. проф. М. А. Бонч-Бруевича, заместитель директора ЛОНИИС, автор более 200 печатных работ в области инфокоммуникаций, в числе которых книги «Сигнализация в сетях связи», «Протоколы сети доступа», «Evolution of Telecommunication Protocols», «Системы коммутации» и др.

Гольдштейн Александр Борисович — кандидат технических наук, начальник сектора ЛОНИИС, родился в 1980 г., в 2001 г. окончил СПбГУТ им. проф. М. А. Бонч-Бруевича, в 2004 г. там же защитил кандидатскую диссертацию «Исследование механизма туннелирования мультимедийного трафика в сети MPLS», опубликовал более 20 печатных работ по перспективным направлениям современных инфокоммуникаций — MPLS, Softswitch и др.

Гольдштейн Александр Борисович
Гольдштейн Борис Соломонович

ТЕХНОЛОГИЯ И ПРОТОКОЛЫ MPLS

Компьютерная верстка *М.А. Фрост*

ИБ № 3004 ЛР № 065953 от 15.08.98

Подписано в печать 22.11.2004
Формат 70×100/16.
Бумага офсетная.
Гарнитура прагматика. Печать офсетная.
Объем 19 печ. л.
Тираж 4000 экз. Зак № 3405
Издательство «БХВ — Санкт-Петербург», 198005, Санкт-Петербург, Измайловский пр., д.29

Отпечатано с готовых диапозитивов
в ГУП «Типография «Наука»
199034, Санкт-Петербург, 9-я линия В.О., 12

А.Б. Гольдштейн
Б.С. Гольдштейн

ТЕХНОЛОГИЯ MPLS
ПРОТОКОЛЫ MPLS
MultiProtocol Label Switching

Многопротокольная коммутация по меткам MPLS – это именно тот инструмент, с помощью которого сегодняшний хаос неуправляемой передачи пакетов по IP-сети может быть превращен в стройный, эффективно функционирующий механизм для сети связи следующего поколения NGN. Эта технология - синтез всего самого лучшего из технологий уровня 2 типа ATM, Frame Relay и Ethernet и маршрутизации уровня 3 пакетных IP-сетей. Изучение рассмотренных в книге протоколов позволит читателю оценить, насколько MPLS позитивна и перспективна, представляет ли она действительно главное направление эволюции сетевых технологий при переходе к NGN.

В книге рассматриваются:

- Принципы MPLS, ее функционирование, сильные и слабые стороны
- Структура метки, таблицы маршрутизации
- Классы эквивалентности пересылки FEC
- Протокол LDP, принципы назначения меток
- Протоколы OSPF и IS-IS, типы сообщений
- Протокол BGP, спикеры, метрики и атрибуты
- Туннели и VPN
- Инжиниринг трафика
- Протоколы RSVP и RSVP-TE
- MPLambS и GMPLS,

а также перспективы технологии MPLS, включая «Все поверх MPLS», VoMPLS, DiffServ-aware, MPLS-TE, проблемы QoS и многое другое.

ЗАКАЗ КНИГ:
191186
Россия
Санкт-Петербург
Абонентский ящик 138

Факс: (812) 3892972
Телефон: (812) 3896897
E-mail: nio1@loniis.ru
Internet: www.niits.ru

ISBN 5-8206-0126-2
9 785820 601262