Санкт-Петербургский государственный университет
телекоммуникаций им. проф. М.А. Бонч-Бруевича

УТВЕРЖДАЮ:
Заведующий кафедрой сетей связи
проф., д.т.н. Г.Г.Яновский

_____

ДИПЛОМНАЯ РАБОТА

«Разработка функциональных моделей механизмов обеспечения гарантированного качества обслуживания в IP-сетях»

Студент-дипломник
5 курс, гр. СК-62 **А.Б. Гольдштейн**

Руководитель
профессор, д.т.н. **Г.Г. Яновский**

Санкт-Петербург
2001

СОДЕРЖАНИЕ

РЕФЕРАТ

Дипломная работа посвящена разработке функциональной модели механизма обеспечения гарантированного качества обслуживания.

Пояснительная записка имеет объем 103 страницы, включает в себя 15 рисунков, 4 таблицы. При написании работы использовано 24 литературных источников.

Ключевые слова и словосочетания: IP-телефония, качество обслуживания, задержки, механизмы обеспечения качества обслуживания, потери пакетов.

В ходе выполнения дипломной работы разработана функциональная модель резервирования на основе протокола RSVP, описаны механизма измерения задержек и качества речи при передаче по сетям IP-телефонии.

ВВЕДЕНИЕ

Тип трафика, который передается по сетям IP, сегодня драматически меняется. Сети используются для прослушивания музыки, просмотра видеоклипов, организации мультимедийной конференц-связи, интернет-телефонии (голосовой и/или видео), распределенного моделирования, управления в реальном времени, сетевых игр и других приложений реального времени. Интернет-Протокол (IP) является протоколом передачи дейтаграмм, который первоначально кажется непригодным для обслуживания трафика в реальном времени, поскольку каждый пакет в потоке данных маршрутизируется независимо в сети с распределенным использованием, в силу чего такие параметры функционирования, как пропускная способность, задержка и вариации задержки, могут меняться в широких пределах.

Трафик в реальном времени больше страдает от перегрузки, чем трафик не в реальном времени. В данное время Интернет в целом не обеспечивает никаких средств для того, чтобы гарантировать качество соединений. Налицо необходимость каким-то образом гарантировать, что в периоды перегрузки трафик реального времени совершенно не будет затронут или, по крайней мере, получит более высокий приоритет, чем трафик не в реальном времени.

Все это доказывает чрезвычайно высокую актуальность предпринятой в данной дипломной работе попытки построения функциональной модели протокола RSVP обеспечения качества обслуживания IP-телефонии.

В работе представлены 3 уровня разработки функциональной модели RSVP, уместившиеся в 5 главах данной дипломной работы.

На первом уровне функциональной модели рассматриваются критерии и количественные характеристики качества обслуживания (QoS), оценки уровня обслуживания R, вероятности потери пакетов и составляющие задержки при передаче пакетов по сети IP-телефонии. Это рассмотрение (глава 2) чрезвычайно важно как с точки зрения анализа эволюции критериев и оценок качества обслуживания телефонных вызов от традиционной до IP-телефонии, от определяемой по формуле Эрланга вероятности потерь по вызовам в системах коммутации каналов до определяемых алгоритмами кодеков, организацией буферов очередей и резервированием пропускной способности оценок качества обслуживания пакетной коммутации речевой информации.

Соответственно изменяются и методы улучшения качества обслуживания вызовов. Увеличение числа обслуживающих приборов и межстанционных соединительных линий в традиционной АТС с коммутацией каналов согласно формуле Эрланга для IP-телефонии превращается в относительно сложные алгоритмы резервирования пропускной способности сети RSVP, методы и подходы модели DiffServ, протокола MPLS и др., рассматриваемых в главе 3.

Второй уровень функциональной модели RSVP представлен рисунком 4.1, описывающем функциональную структуру взаимодействия компьютера (хоста) и маршрутизатора, реализующих совместно резервирование пропускной способности сети для обеспечения гарантированного качества обслуживания. Также в главе 4 представлено подробное описание работы протокола RSVP, примеры сообщений, типов резервирования.

И, наконец, третий уровень модели – модель управления трафиком - реализуется непосредственно на маршрутизаторах поддерживающих резервирование протокола RSVP, и представлена в виде управляемых очередей. Эта часть модели показывает каким образом осуществляется непосредственно обслуживание трафика, предоставление ему необходимого уровня качества обслуживания. Также в главе 5 рассмотрены компоненты классификации трафика и планирования пакетов функции управления телетрафиком и описаны различные алгоритмы планирования пакетов и их пригодность для обеспечения различных гарантий QoS

Глава 1. ПРИНЦИПЫ IP-ТЕЛЕФОНИИ

1.1. Актуальность темы дипломной работы

Настоящая дипломная работа сосредоточена на решении одной из задач новой технологии, все более завоевывающей популярность в мире - технологии передачи речевой информации по сетям с маршрутизацией пакетов IP - Voice over IP (VoIP) или IP-телефонии.

Данная технология представляет собой компрессию голосового сигнала с последующей передачей по цифровым каналам передачи данных с использованием протокола IP. Частным случаем IP-телефонии является Интернет-телефония, при которой в качестве передающей сети используется сеть Интернет. При том, что IP-телефония - сравнительно новая технология, она уже успела зарекомендовать себя вполне перспективной и жизнеспособной. Согласно прогнозам аналитиков, в мире ожидается значительный (в ближайшие годы в десятки раз) рост речевого трафика, переносимого по сетям с маршрутизацией пакетов.

Объем рынка услуг IP-телефония сегодня составляет примерно миллиард долларов, а в течение ближайших нескольких лет, по разным прогнозам, вырастет до величин порядка 8,5-24 млрд. долл. (например, по данным IDC, в 2002 году объем рынка составит 23 млрд. долл.). Несмотря на разброс оценок величины рынка, большинство аналитиков сходятся на том, что IP-телефония является магистральным путем развития телекоммуникационной индустрии. Дополнительно IP-телефония вносит новые услуги в сферу телекоммуникаций: речевые и видеоконференции, одновременный доступ к приложениям, быстрый поиск абонента и другие.

IP - всего лишь протокол, который наравне с некоторыми другими (ATM, FR) используется для передачи по сети Интернет оцифрованной и сжатой в «пакеты» определенного объема электронной информации. В принципе, после соответствующей обработки таким способом на любые досягаемые для паутины Интернета расстояния может передаваться любой тип информации (голосовая, видео, компьютерные данные), так что собственно телефония является лишь одной составляющей общего процесса, однако, материал данной дипломной работы ограничивается только ею.

Преимущества IP-телефонии неоспоримы. В первую очередь это низкая стоимость передачи информации. А также это универсальность обработки информации независимо от ее исходного вида, а, следовательно, использование одних и тех же каналов для передачи информации разного типа. При этом популярность технологий пакетной передачи речи растет, причем как среди операторов, так и среди корпоративных пользователей. В последние годы немало компаний успешно внедрили эти технологии для организации каналов дальней телефонной связи.

Все вышеизложенное показывает, что выбранная тема дипломной работы актуальна и важна.

1.2. Историческая справка.

До 1995 года Интернет-телефония не была широко распространена. Существовали отдельные свободно распространяемые программы, созданные программистами-исследователями, без перспективы их широкого распространения, так как считалось невозможным получить качественное голосовое соединение через Интернет. Компания VocalTec, выпустившая программу Internet Phone в феврале 1995, доказала, что это не так. Установив на персональный компьютер (486/33-MHz или выше) со средствами мультимедиа (звуковая карта, микрофон, колонки) и подключением к Интернет программу Internet Phone, можно разговаривать с другим пользователем персонального компьютера в любой точке мира, где есть доступ в Интернет. В конце 1995 года общее количество активных пользователей Интернет-телефонии оценивалось уже в 500.000. Львиную долю этого рынка – 94% - занимала продукция Vocal Tec. Активные пользователи определяются как люди, которые пользуются Интернет-телефонией на постоянной основе.

В 1996 году Интернет-телефония за один год выросла на 97% от оцененного в $1,8 миллионов рынка (источник: Frost&Sullivan, 1997). К 1997 году общее число минут, переданных по IP-сетям, составило 0,1 млрд (источник: US Bancorp Piper Jaffray), а в 1998 году голосовой трафик Интернет-телефонии в компаниях, входящих в список Fortune 1000, составляет менее 1% от всех междугородных и международных звонков. В 1999 году использование IP-протокола в телефонии достигло в общей сложности 310 млн. минут и 0.48 млрд. минут.

Россия не была исключением среди стран, проявивших интерес к IP-телефонии. Интерес к новой технологии в нашей стране имеет особый практический смысл. Дело в том, что в нашей стране услуги международной телефонной связи традиционно дороже, чем на Западе. Кроме того, географические размеры нашей страны делают актуальным вопрос стоимости междугородней связи. А использование IP-технологии подразумевает возможность ведения телефонных разговоров по более дешевому тарифу, что позволяет сэкономить значительные средства удаленным на большие расстояния абонентам.

В 1996 году российско-американская компания CompTek приобрела пробный комплект оборудования для IP-связи, состоящий из двух шлюзов производства израильской фирмы VocalTec. Это были первые в мире устройства для телефонных разговоров через IP-сеть. Шлюзы поддерживали неплохое качество передачи речи. И в скором времени заключил с VocalTec соглашение о дистрибьюции шлюзов на территории России и СНГ. Хотя этот проект ожидал большой коммерческий успех, возникли некоторые проблемы. С кем будет связываться имеющийся шлюз, если развитых IP-сетей в России еще не существует? Поэтому изначально покупателями шлюзов становились в основном территориально распределенные компании, использовавшие IP-телефонию для связи между своими офисами. Очевидный выход из сложившейся ситуации заключался в налаживании партнерских отношений с фирмами-операторами. На этом этапе ведущую роль в продвижении IP-технологии в России сыграла компания “Тарио”. К тому времени она уже сотрудничала с VocalTec и появление IP-шлюзов открыло новые перспективы и возможности для совместного бизнеса. Установкой шлюзов после заключения соответствующих соглашений занялись компании “Тарио” и RGC. При этом “Тарио” строила сети шлюзов в городах России, RGC налаживала связи для предоставления услуг международной связи.

Начиная с 1997 года в России стали реально строиться сети IP-телефонии. Правда, из-за неопределенности с лицензированием эта услуга предоставлялась с пометкой “в тестовом режиме”. Тем не менее, технология широко рекламировалась в прессе и демонстрировалась на выставках. Там постоянно предоставлялась бесплатная возможность сделать городские, междугородние и международные звонки через систему IP-телефонии. Представители компании

VocalTec, в то время единственной фирмы, поставлявшей оборудование для IP-сетей, участвуя в различных семинарах и конференциях, неоднократно заявляли, что Россия обогнала многие страны на пути предоставления коммерческих услуг по междугородной и международной IP-связи.

В то же время крупные Интернет-провайдеры и телефонные операторы не спешили осваивать новую технологию. Свои услуги на рынке активно предлагали только небольшие и средние компании. Такое положение объяснялось двумя аспектами – правовым и техническим.

Порядка лицензирования для IP-телефонии еще не существовало и эта технология распространялась у нас практически нелегально. А, следовательно, крупные фирмы не могли выйти на рынок с рекламной компанией услуг IP-телефонии.

Технический аспект состоял в несовершенстве оборудования. Первые шлюзы поддерживали не более одного потока E1 (до 30 одновременных звонков), а качество связи было низким, что приводило к значительному искажению голосовых сообщений. Со временем эти проблемы начали решать, и хотя до окончательного решения проблемы качества еще далеко (примером чему является в том числе и данная дипломная работа), уже 1998 г. ознаменовался радикальным скачком в развитии технологии.

27 мая 1998 г. по решению коллегии Министерстве по связи и информатизации (тогда называвшемся Госкомсвязи) в Ассоциации документальной электросвязи (АДЭ) была создана рабочая группа Internet-телефонии, членом которой является и ЛОНИИС. В группу вошли фирмы-разработчики и фирмы – поставщики оборудования, а также традиционные операторы связи (всего 41 компания). Группе было поручено разработать пакет документов для легализации Интернет-телефонии в России.

1 июня 1999 г. IP-телефония была официально признана “телематической службой передачи речевой информации”. На сегодняшний день уже многие российские компании, получив лицензии, приступили к внедрению IP-телефонии либо рассматривают такую возможность, решая, на каком оборудовании остановить свой выбор.

Сегодня на рынке IP-телефонии весьма успешно внедряется отечественное оборудование ПРОТЕЙ-IP, разработанное в нашем городе. Стоимость оборудования IP-телефонии платформы ПРОТЕЙ-IP значительно ниже

аналогичного оборудования Cisco Systems, а качество вполне соответствует общепринятым стандартам.

Но прежде рассмотрим место IP-телефонии среди других технологий пакетной передачи речи.

1.3. Сравнение технологий пакетной коммутации

Все основные технологии для объединенных сетей - передача речи поверх IP, ATM и Frame Relay – имеют свои достоинства и недостатки, и ни одна из них сама по себе не может решить все проблемы. Вырабатывая стратегию развития сети связи, необходимо четко представлять себе, на какие уступки вам придется пойти при выборе той или иной технологии и в какой мере каждая из них будет способствовать достижению конечной цели.

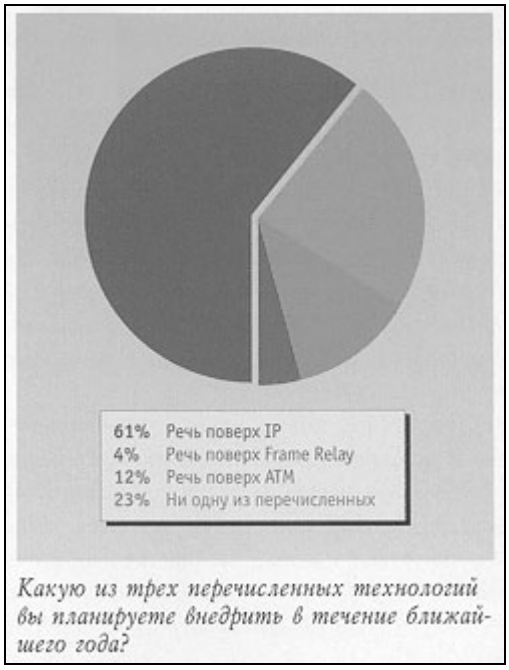


Рис.1.1.Результаты опроса операторов

Для обслуживания телефонного трафика в сети передачи данных требуется решить не одну, а множество задач, для решения которых отсутствует какой-либо универсальный рецепт. Рассмотрим отдельно разные участки сетевых инфраструктур для выбора тех или иных технологий. Всоответствии с техническим заданием на данную дипломную работу это сравнение следует рассматривать в контексте происходящей сегодня конвергенции сетей связи.

Экономические аспекты конвергенции, связанные с снижением общих расходов, складывающихся из капитальных затрат на приобретение и инсталляцию телекоммуникационного и сетевого оборудования, из затрат на его содержание, и с уменьшением необходимости в квалифицированном персонале (одни и те же люди стали бы заниматься и телефонией, и системами передачи данных) упоминаются в соответствующей технико-экономической главе данной дипломной работы. Здесь же отметим лишь очевидное: наличие всего одного

канала доступа к распределенной сети тоже основательно снизило бы ежемесячные расходы. Пуская речевой трафик через магистральную сеть передачи данных, можно существенно уменьшить затраты на традиционные телефонные услуги. И, наконец, сокращение единиц используемого оборудования позволит, очевидно, снизить расходы на его техническое обслуживание.

Наряду с экономическими факторами технология IP-телефонии привлекает в первую очередь своей универсальностью – речь может быть преобразована в поток IP-пакетов в любой точке сетевой инфрастуктуры: на магистральной сети оператора, на границе территориально распределенной сети и даже непосредственно в терминале конечного пользователя. В конце концов, она станет наиболее широко распространенной технологией пакетной телефонии, поскольку способна охватить все сегменты рынка, являясь при этом хорошо адаптируемой к новым условиям применения. Несмотря на универсальность протокола IP, большинство проектировщиков сетей не спешат с внедрением систем IP-телефонии, считая их недостаточно надежными, плохо управляемыми и не очень эффективными. Но если внедрять их в грамотно спроектированной коммутируемой сетевой инфраструктуре с эффективными механизмами обеспечения качества обслуживания (особенно с T\QoS-сигнализацией), то эти недостатки становятся несущественными. В расчете на порт стоимость систем IP-телефонии находится на уровне или немного выше стоимости систем Frame Relay, но заведомо ниже стоимости оборудования ATM. При этом в ближайшие годы цена на продукты IP-телефонии будет снижаться быстрее, чем на другие изделия, поскольку предполагается значительное обострение конкуренции на этом рынке.

1.4. Экономическая эффективность IP-телефонии

Определяющим фактором привлекательности IP-телефонии для конечных потребителей безусловно является дешевизна междугородных и международных телефонных переговоров через IP-сети по сравнению с традиционной телефонной связью. Так, например, одна минута обычного телефонного разговора с США для конечного потребителя в Москве стоит сегодня порядка $1,5 - $2,3, а через Интернет обходится ему $0,25 - $0,8, т.е. на

каждом пятиминутном разговоре потребитель услуг IP-телефонии экономит от 6 до 7,5 долларов США в зависимости от расстояния и используемой сети.

Кроме того, процесс вызова абонента для конечного потребителя услуг IP-телефонии практически не отличается от традиционной телефонной связи. Человек, который хочет позвонить через IP-сеть, на обыкновенном телефонном аппарате вместо традиционной восьмерки набирает местный номер (который является номером ближайшего телефонного сервера) и слышит голосовое приглашение с предложением набрать телефонный номер вызываемого абонента (или сначала идентификационный номер, если это оговорено договором). После набора номера система IP-телефонии (через второй сервер) соединяет потребителя с телефоном вызываемого абонента. Если разговор не может состояться, звонящий будет голосом проинформирован о причине невозможности соединения (например, «вызываемый номер занят», «все линии на удаленном телефонном сервере заняты», «удаленный телефонный сервер недоступен», «неверно набран номер» и т.п.) Система также передает звонящему абоненту такие телефонные сигналы, как сигнал «вызов», «занято» и пр. Таким образом, звонящий слышит привычные ему реальные сигналы телефонной сети.

Кроме того можно показать преимущества использования IP-телефонии для операторов:

Для операторов**:**

Что касается поставщиков услуг, т.е. операторов, то в данном случае их целесообразно разделить на 2 категории: коммерческие и ОАО «Электросвязь».

Для коммерческих операторов:

- возможность предоставления современных услуг на базе IP-телефонии: междугородней и международной связи, информационно-справочных услуг на базе Call Center, виртуальной телефонной линии, универсального номера доступа, Web-телефонии (click to dial).
- возможность предоставления услуг телефонии на базе более дешёвого оборудования, т.к. отсутствует необходимость обеспечивать качество и объем услуг, требуемых от операторов ТфОП
- возможность значительного сокращения затрат

Как отметил представитель одного международного оператора связи, переход на технологию IP-телефонии позволит ему сэкономить порядка 70% средств на капитальные затраты, 60-80% средств, выделяемых на организацию каналов доступа, и 50% средств на текущее обслуживание и ремонт сети.

Для операторов ОАО «Электросвязь»:

- возможность конкурировать с альтернативными операторами;
- возможность получения выгоды от терминизации трафика;
- привлечение новых клиентов путем снижения стоимости услуг, а, следовательно, увеличение трафика;
- возможность внедрения новых услуг: виртуальная линия, создание центров обслуживания вызовов, Web-телефония;
- легальная возможность обойти Ростелеком.

Для провайдеров Интернет:

Основным преимуществом провайдера Интернет по сравнению с другими компаниями, планирующими предоставлять услуг IP-телефонии, является наличие собственного выделенного канала, благодаря чему существенно сокращаются капитальные затраты на организацию подобного сервиса.

Основными аргументами в пользу внедрения IP-телефонии провайдерами Интернет являются:

- возможность предоставления своим клиентам новых услуг, за которые они готовы платить в несколько раз больше, чем просто за обычный доступ к Internet;
- возможность привлечения множества новых клиентов, у которых до сих пор необходимости в Internet не возникало;
- предоставление телефонных услуг через инфраструктуру IP позволяет провайдеру Интернет получать большую, по сравнению с традиционными провайдерами, прибыль благодаря тому, что функции предоставления услуг телефонии и передачи данных объединяются в общей инфраструктуре IP; основной объем обслуживаемого трафика приходится на традиционные данные Интернет, а транспортировка относительно невысокого объема трафика IP-телефонии может

осуществляться с использованием той же инфраструктуры при очень незначительных дополнительных затратах.

Для корпоративного сектора:

Использование технологии IP-телефонии предоставляет корпоративному сектору следующие возможности:

- возможность увеличения конкурентоспособности за счет снижения себестоимости выпускаемого продукта (сокращение расходов на связь и информационное обеспечение);
- возможность оптимизации производственного цикла путем оптимизации коммуникационной системы предприятия;
- возможность внедрения новых услуг.

Внедрение IP-телефонии в корпоративном секторе целесообразно в следующих случаях:

- строительство нового офиса;
- окончание срока службы УАТС или контракта о поддержке;
- необходимость перехода к электронной сетевой торговле;
- организация перерастает возможности существующей УАТС;
- организация переезжает на новое место.

Наряду с преимуществами IP-телефонии имеют место проблемы, препятствующие ее развитию. В первую очередь это уже упоминавшаяся выше проблема качества обслуживания, которая будет подробно рассмотрена далее.

1.5. Преобразование речевого сигнала в IP-пакеты

При переходе к цифровым сетям связи возникла необходимость преобразовать аналоговый электрический сигнал в цифровой формат на передающей стороне, то есть закодировать, и перевести обратно в аналоговую форму, то есть декодировать, на приемной стороне .

Процесс преобразования аналогового речевого сигнала в цифровую форму называют *анализом* или *цифровым кодированием речи*, а обратный процесс восстановления аналоговой формы речевого сигнала – *синтезом* или *декодированием речи*.

Цель любой схемы кодирования – получить такую цифровую последовательность, которая требует минимальной скорости передачи и из которой декодер может восстановить исходный речевой сигнал с минимальными искажениями.

При преобразовании речевого сигнала в цифровую форму, так или иначе, имеют место два процесса – *дискретизация* (*sampling*), т.е. формирование дискретных во времени отсчетов амплитуды сигнала, и *квантование*, т.е. дискретизация полученных отсчетов по амплитуде (кодирование непрерывной величины – амплитуды – числом с конечной точностью). Эти две функции выполняются т.н. цифро-аналоговыми (ЦАП) и аналого-цифровыми (АЦП) преобразователями, которые размещаются в современных АТС на плате абонентских комплектов, а в случае передачи речи по IP-сетям – в терминале пользователя (компьютере или IP-телефоне).

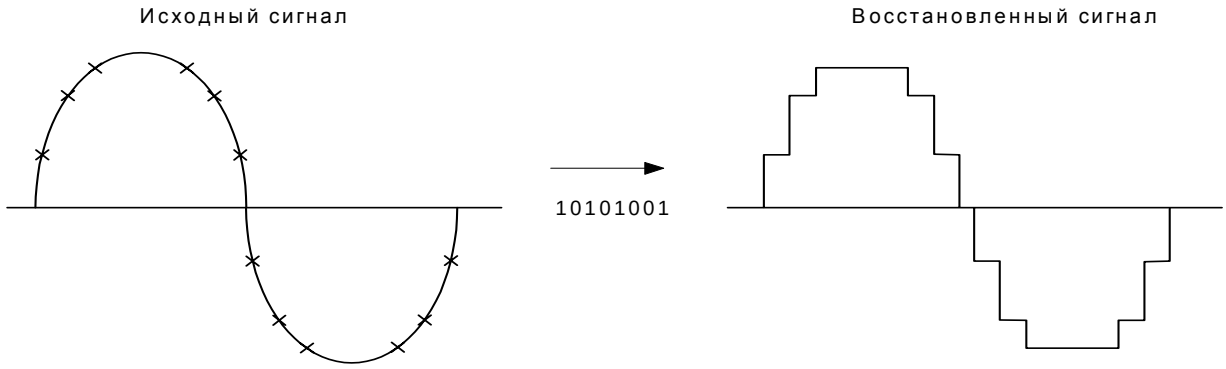


Рис.1.2. Дискретизация и квантование аналогового речевого сигнала.

Теорема Котельникова гласит, что аналоговый сигнал может быть успешно восстановлен из последовательности выборок с частотой, которая превышает, как минимум, вдвое максимальную частоту, присутствующую в спектре сигнала. В телефонных сетях полоса частот речевого сигнала намеренно, посредством специальных фильтров, ограничена диапазоном 0.3 – 3.4 Кгц, что не влияет на разборчивость речи и позволяет узнавать собеседника по голосу. По этой причине частота дискретизации при аналого-цифровом преобразовании выбрана равной 8кГц, причем такая частота используется во всех телефонных сетях на нашей планете.

При квантовании по амплитуде непрерывная величина отображается на множество дискретных значений, что, естественно, приводит к потерям информации. Для того, чтобы обеспечить в такой схеме достаточный динамический диапазон (способность передавать без искажений как сильные, так и слабые сигналы), дискретная амплитуда сигнала кодируется 12/13-ти разрядным двоичным числом по линейному закону.

Процесс аналого-цифрового преобразования получил, применительно к системам связи, название *импульсно-кодовой модуляции* (ИКМ).

Чтобы снизить необходимую скорость передачи битов, применяют нелинейный (логарифмический) закон квантования, т.е. квантованию подвергается не сам сигнал, а значение его логарифма. В данном случае имеет место процесс «сжатия» динамического диапазона сигнала, а при восстановлении сигнала происходит обратный процесс.

Каждый отсчет кодируется 8 битами, или одним байтом, который можно считать звуковым фрагментом. Для передачи последовательности таких фрагментов необходима пропускная способность канала, равная 64 Кбит/с, что определяется простыми арифметическими действиями:

4 000 Гц ∗ 2 = 8 000 отсчетов/с, 8 000 отсчетов/с ∗ 8 битов = 64 Кбит/с,

что составляет основу всей цифровой телефонии. Поскольку ИКМ была первой стандартной технологией, получившей широкое применение в цифровых системах передачи, пропускная способность канала, равная 64 Кбит/с, стала всемирным стандартом для цифровых сетей всех видов, причем – стандартом, который обеспечивает передачу речи с очень хорошим качеством. Соответствующие процедуры кодирования и декодирования стандартизованы ITU-T в рекомендации G.711.

Однако такое высокое качество передачи речевого сигнала (являющееся эталоном при оценке качества других схем кодирования) достигнуто в системах ИКМ за счет явно избыточной, при современном уровне технологии, скорости передачи информации.

Чтобы уменьшить присущую ИКМ избыточность и снизить требования к полосе пропускания, последовательность чисел, полученная в результате преобразования речевого аналогового сигнала в цифровую форму, подвергается математическим преобразованиям, позволяющим уменьшить необходимую

скорость передачи. Эти преобразования «сырого» цифрового потока в поток меньшей скорости называют «сжатием» (а часто – кодированием, рассматривая ИКМ как некую отправную точку для дальнейшей обработки информации).

Существует множество подходов к «сжатию» речевой информации; все их можно разделить на три категории: *кодирование формы сигнала* (waveform coding), *кодирование исходной информации* (source coding) и *гибридное кодирование*, представляющее собой сочетание двух предыдущих подходов.

1.5.1. Кодирование формы сигнала

Импульсно-кодовая модуляция, по сути, и представляет собой схему кодирования формы сигнала. Однако нас интересуют более сложные алгоритмы, позволяющие снизить требования к полосе пропускания.

Рассматриваемые методы кодирования формы сигнала используют то обстоятельство, что между случайными значениями нескольких следующих подряд отсчетов существует некоторая зависимость. Проще говоря, значения соседних отсчетов обычно мало отличаются одно от другого. Это позволяет с довольно высокой точностью предсказать значение любого отсчета на основе значений нескольких предшествовавших ему отсчетов.

При построении алгоритмов кодирования названная закономерность используется двумя способами. Во-первых, есть возможность изменять параметры квантования в зависимости от характера сигнала. В этом случае шаг квантования может изменяться, что позволяет до некоторой степени сгладить противоречие между уменьшением числа битов, необходимых для кодирования величины отсчета при увеличении шага квантования, и сужением динамического диапазона кодера, неизбежным без адаптации (о которой речь пойдет ниже). Некоторые алгоритмы предусматривают изменение параметров квантования приблизительно в рамках произносимых слогов, а некоторые изменяют шаг квантования на основе анализа статистических данных об амплитуде сигнала, полученных за относительно короткий промежуток времени.

Во-вторых, существует подход, называемый *дифференциальным кодированием* или *линейным предсказанием*. Вместо того, чтобы кодировать входной сигнал непосредственно, кодируют разность между входным сигналом и

«предсказанной» величиной, вычисленной на основе нескольких предыдущих значений сигнала.

Если отсчеты входного сигнала обозначить как y(i), то предсказанное значение в момент времени i представляет собой линейную комбинацию нескольких p предыдущих отсчетов:

$$y(i) = a_1 y(i-1) + a_2 y(i-2) + \ldots + a_p y(i-p),$$

где множители a_i называются *коэффициентами предсказания*.

Разность e(i)=y(i)-$y(i)$ имеет меньший динамический диапазон и может кодироваться меньшим числом битов, что позволяет снизить требования к полосе пропускания.

Описанный метод называется *линейным предсказанием*, так как он использует только линейные функции предыдущих отсчетов. Коэффициенты предсказания выбираются так, чтобы минимизировать среднеквадратическое значение ошибки предсказания e(i), при этом значения коэффициентов изменяются, в среднем, каждые 10-25 мс.

Простейшей (и представляющей сегодня, скорее, исторический интерес) реализацией последнего подхода является так называемая *дельта-модуляция* (ДМ), алгоритм которой предусматривает кодирование разности между соседними отсчетами сигнала только одним информационным битом, обеспечивая передачу, по сути, только знака разности.

Наиболее совершенным алгоритмом, построенным на описанных выше принципах, является алгоритм адаптивной дифференциальной импульсно-кодовой модуляции (АДИКМ), предложенный ITU-T в рекомендации G.726. Алгоритм предусматривает формирование сигнала ошибки предсказания и его последующее адаптивное квантование. Существует версия этого алгоритма, в которой информационные биты выходного цифрового потока организованы по иерархической схеме, что позволяет отбрасывать наименее значимую информацию, не уведомляя об этом кодер, и получать поток меньшей скорости за счет некоторого ухудшения качества. Документ G.726 специфицирует кодирование при скоростях 40, 32, 24 и 16 Кбит/с, что соответствует передаче 5, 4, 3 или 2 битов на отсчет. Качество речи, передаваемой с использованием АДИКМ G.726 при скорости 32 Кбит/с соответствует качеству речи, обеспечиваемому алгоритмом кодирования G.711.

При достаточно хороших характеристиках алгоритма, АДИКМ практически не применяется для передачи речи по сетям с коммутацией пакетов, так как этот алгоритм очень чувствителен к потерям целых блоков отсчетов, происходящим при потерях пакетов в сети. В таких случаях нарушается синхронизация кодера и декодера, что приводит к катастрофическому ухудшению качества воспроизведения речи даже при малой вероятности потерь.

1.6. Проблемы передачи речевой информации по IP-сетям

Эхо возникает при любых голосовых коммуникациях. В телефонии разделяют два вида этого явления: акустическое и гибридное. Акустическое эхо возникает вследствие отражения звука от стен помещения или между микрофоном и телефоном, а гибридное, возникает в телефонной сети, поэтому на него необходимо обращать внимание при построении телефонного шлюза. В основном эхо возникает на стыках 4-х и 2-х проводных сегментов из-за плохого согласования сопротивлений. Часть энергии речевого сигнала при переходе им из 4-х проводной линии в 2-хпроводную может отразиться в направлении источника в виде искаженной или задержанной копии сигнала. Если в канале небольшая задержка, то особых проблем не возникнет, но при увеличении задержки говоривший будет слышать «свое» эхо. Эхо с обычной задержкой в 16-20 мс называется *побочным тоном* и представляет собой нежелательное, но не существенное явление. Однако, эхо с задержкой, превышающей 32 мс может беспокоить говорящего.

По рекомендации G.131 МСЭ-Т двусторонняя задержка не должна превышать 250 мс. А в системах IP-телефонии задержка может быть гораздо больше. Только кодек построенный по рекомендации G.723.1 вносит задержку на 37,5 мс. В итоге необходимо учитывать эхо при построении IP-сети в целом. В сетях с коммутацией каналов эхо обычно устраняют на коммутаторах подключенных к каналам с высоким уровнем задержки, но шлюз IP-телефонии в основном подключают к местной АТС или даже УАТС, которые не устраняют эхо.

Существуют два основных механизма борьбы с этим явлением: подавление (suppression) – основано на рекомендации G.164, определяет направление передачи и гасит все сигналы идущие навстречу, при этом связь приобретает

полудуплексный характер и второй, более современный механизм, называемый компенсацией (cancellation). Этот механизм описан в рекомендациях G.165 и G.168, позволяет устранять отраженный сигнал при сохранении дуплексной связи.

В отличие от эхоподавителя, в эхокомпенсаторе присутствует линейный фильтр, который при прохождении через него прямого сигнала формирует имитацию эхо-сигнала и вычитает ее из обратного сигнала. Эффективность фильтра будет зависеть от точности расчета задержки и имитации, от степени адаптации фильтра к изменению условий в процессе разговора.

Однако очень часто одного линейного фильтра недостаточно для удовлетворительной эхокомпенсации и сигнал с выхода фильтра пропускают через нелинейный процессор, который полностью или частично блокирует сигнал при наличии в нем остаточного эха. Но при использовании нелинейного процессора сигнал может исказиться, слова будут звучать как «рубленые». Качество эхокомпенсаторов можно определить несколькими ключевыми параметрами:

- *время сходимости* - время, требуемое эхокомпенсатору для адаптации к наблюдаемой линии и обеспечению соответствующего эхо-подавления.
- *глубина компенсации* - достигается сокращение силы эха, измеряется в дБ.
- *устойчивость к переходному разговору* - эхокомпенсатор не теряет способности к подавлению в условиях одновременного разговора на обоих концах соединения.

Чтобы более эффективно использовать полосу пропускания речь в IP-сети сжимается сильнее, чем в ТфОП. Также используется технология подавления пауз или, как ее еще называют, технология прерывистой передачи (Discontinuous Transmission – DTX). Так как в каждый конкретный момент разговора говорит только один собеседник, то от второго будут передаваться «пустые» пакеты. Используя механизм идентификации пауз можно приостанавливать передачу на время паузы. Таким образом, можно экономить до 50 % полосы пропускания. Такой механизм кодирования

называется VAD – Voice Activity Detection, и позволяет засечь, когда человек начинает говорить и момент когда он заканчивает. Но есть сложности. Во-первых, из-за отсечки потенциально снижается качество речи, а во-вторых, первые звуки речи могут быть восприняты как шум и блокированы; когда громкость возрастет до определенного уровня, система определит это как речь и начнет передачу. Это будет воспринято вторым абонентом как раздражающее исчезновение звука. Т.е. основная задача механизма VAD будет заключаться в возможности отличать шум от речи. Но шумовой фон присутствует всегда, при любом разговоре. Значит необходимо установить некий порог – как только уровень сигнала превышает этот порог, а сам сигнал по характеру соответствует речевому, система делает вывод, что человек начал говорить. Очень важно определить этот момент как можно раньше, т.к. может потеряться первая часть слова. Для определения оптимального соотношения между вероятностью принятия шума за речь и риском отсечки звуков надо учитывать, что, во-первых, спектральная форма речи изменяется через короткие промежутки времени (примерно 20-30 мс), а шумовой фон в основном остается стационарным, а, во-вторых, уровень речевого сигнала обычно заметно выше.

Есть еще одна особенность, нельзя совсем прекращать передачу на время паузы, т.к. у второго абонента может возникнуть ощущение, что телефон первого не работает. DTX добавляет на стороне приемника так называемый «комфортный шум», который создает ощущение обычного фона. Подход DTX уже давно используется в сотовой связи, а GSM—кодеки имеют встроенные средства DTX/VDA.

В настоящее время существует множество различных кодеков и зачастую речевой шлюз может иметь несколько вариантов кодеков и для каждого отдельного вызова выбирать свой кодек. Выбор кодека может зависеть от загруженности сети, условий обслуживания данного абонента и, конечно от набора кодеков на противоположном шлюзе.

1.7. Кодеки и качество кодирования речи

Речевой кодек – это сочетание кодера, который преобразует сигнал однородной ИКМ в компактный формат, и декодера, который осуществляет обратное преобразование.

Есть два основных типа кодеров: вокодеры (vocoders) и сигнальные кодеры (waveform coders). Последние, новейшие кодеры базируются на комбинации этих двух типов и называются гибридными. Для сигнальных кодеров природа обрабатываемого сигнала не важна. Известным представителем этой группы является ИКМ-кодек G.711, используемый в традиционных сетях коммутации каналов. Сигнальные кодеры просты, быстро работают и обеспечивают хорошее качество, но требуют очень широкую полосу пропускания. Вокодеры анализируют характерные параметры речевого сигнала и передают их принимающей стороне, которая синтезирует сигнал в соответствии со значениями этих параметров. Обычно вокодер анализирует фрагменты речи от 10 до 40 мс, т.к. в течение такого короткого интервала времени речевой сигнал можно считать неизменным. Кодеки такого типа обеспечивают очень высокую степень сжатия, но, во-первых, придают речи «искусственный» характер, а во-вторых, музыку и шумовой фон передают с очень плохим качеством. Гибридные кодеки (например: GSM, G.728, G.729) сочетают высокую степень сжатия и высокое качество передачи. Но устройство таких кодеков значительно сложнее, чем у вокодеров или сигнальных кодеков.

Существует множество критериев для сравнения кодеков, но в основном применяются четыре параметра: скорость выходного потока, качество речи, вносимая задержка и сложность реализации. В таблице 1.1 приведены характеристики наиболее популярных в IP-телефонии кодеков.

Таблица 1.1. Сравнение кодеков

Кодек	Скорость потока, Кбит/сек.	Размер кадра, мс.	Сложность (загрузка процессора)
G.711	64	0,125	Низкая
G.728	16	0,625	Очень высокая
GSM FR	13	20	средняя
GSM EFR	12,2	20	Высокая
G.729.a	8	10	Высокая/очень высокая
G.723.1	5,3/6,4	30	высокая

Для достижения хорошей скорости передачи нужен сложный алгоритм кодирования, а чем сложней алгоритм, тем больше мощности процессора он требует. Значит конкретная аппаратная платформа, на которой реализован шлюз, сможет обслуживать меньшее число разговоров, чем при использование простых алгоритмов кодирования. А если архитектура шлюза позволяет использовать несколько кодеков, то нужно знать, сколько разговоров одновременно может обслуживать каждый из кодеков и желательно иметь устройство автоматического выбора оптимального для данной ситуации кодека и его настройки.

Если рассматривать проблему кодеков с точки зрения экономических проблем, то скорость выдаваемого кодеком потока бит также будет очень важна, т.к. этим будет определятся максимальное число одновременных разговоров на данной линии связи. Кроме того, не следует забывать о «расходах» на передачу служебной информации пакетов. В H.323 это будут заголовки пакетов IP, UDP, RTP, причем в сумме это будет 40 байт. И эту величину, естественно, следует учитывать при расчете полосы пропускания и суммарной задержки пакетов. Конечно можно применять технологию сжатия заголовков и вместо 40 байт передавать примерно 4, но тогда нужно задействовать дополнительную процессорную мощность, а все маршрутизаторы должны поддерживать соответствующие алгоритмы.

Размер кадра. Большинство узкополосных кодеков обрабатывает речевую информацию блоками, называемыми кадрами (frames), и им необходимо производить предварительный анализ отсчетов, следующих непосредственно за отсчетами в блоке, который они в данный момент кодируют.

Размер кадра важен, так как минимальная теоретически достижимая задержка передачи информации (алгоритмическая задержка) определяется суммой этого параметра и длины буфера предварительного анализа. В действительности процессоры цифровой обработки сигналов, которые выполняют алгоритм кодеков, имеют конечную производительность, так что реальная задержка сигнала больше теоретической.

Можно, казалось бы, заключить, что кодеки с меньшим размером кадра лучше в смысле такого важного критерия как минимизация задержки. Если,

однако, учесть, что происходит при передаче информации по сети, то мы увидим, что к кадру, сформированному кодеком, добавляется множество дополнительной информации – заголовки IP (20 байтов), UDP (8 байтов), RTP (12 байтов). Для кодека с длительностью кадра 30 мс посылка каждого кадра по сети привела бы к посылке избыточной информации со скоростью 10.6 кбит/с, что превышает скорость передачи для большинства узкополосных кодеков.

Поэтому обычно используется пересылка нескольких кадров в пакете, при этом их количество ограничено максимально допустимой задержкой. В большинстве случаев в одном пакете передается до 60 мс речевой информации. Чем меньше длительность кадра, тем больше кадров приходится упаковывать в один пакет, т.е. задержка определяется вовсе не длиной кадра, а практически приемлемым объемом полезной нагрузки в пакете.

Кроме того, кодеки с большей длиной кадра более эффективны, так как здесь действует общий принцип: чем дольше наблюдается явление (речевой сигнал), тем лучше оно может быть смоделировано.

Таким образом, с точки зрения выбора кодека для приложений IP-телефонии, длина кадра большого значения не имеет.

Чувствительность к потерям кадров. Потери пакетов являются неотъемлемым атрибутом IP-сетей. Так как пакеты содержат кадры, сформированные кодеком, то это вызывает потери кадров. Однако потери пакетов и потери кадров не обязательно напрямую связаны между собой, так как существуют подходы (такие как применение кодов с исправлением ошибок – forward error correction), позволяющие уменьшить число потерянных кадров при данном числе потерянных пакетов. Требующаяся для этого дополнительная служебная информация распределяется между несколькими пакетами, так что при потере некоторого числа пакетов кадры могут быть восстановлены.

Однако положительный эффект от введения избыточности для борьбы с потерями пакетов не столь легко достижим, поскольку потери в IP-сетях происходят пачками, т.е. значительно более вероятно то, что будет потеряно сразу несколько пакетов подряд, чем то, что потерянные пакеты распределятся в последовательности переданных пакетов по одному. Так что если применять простые схемы введения избыточности (например, повторяя каждый кадр в двух последовательно передаваемых пакетах), то в реальных условиях они, хотя

и увеличат объем избыточной информации, но, скорее всего, окажутся бесполезными.

Кроме того, введение избыточности отрицательно сказывается на задержке воспроизведения сигнала. Например, если мы повторяем один и тот же кадр в четырех пакетах подряд, чтобы обеспечить возможность восстановления информации при потере трех подряд переданных пакетов, то декодер вынужден поддерживать буфер из четырех пакетов, что вносит значительную дополнительную задержку воспроизведения.

Влияние потерь кадров на качество воспроизводимой речи зависит от используемого кодека. Если потерян кадр, состоящий из N речевых отсчетов кодека G.711, то на приемном конце будет отмечен пропуск звукового фрагмента длительностью N*125 мкс. Если используется более совершенный узкополосный кодек, то потеря одного кадра может сказаться на воспроизведении нескольких следующих, так как декодеру потребуется время для того, чтобы достичь синхронизации с кодером – потеря кадра длительностью 20 мс может приводить к слышимому эффекту в течение 150 мс и более. Кодеры типа G.723.1 разработаны так, что они функционируют без существенного ухудшения качества в условиях некоррелированных потерь до 3% кадров, однако при превышении этого порога качество ухудшается катастрофически.

Получается, что выбор кодека превращается в поиск компромисса (как и все в жизни) между качеством и полосой пропускания.

По оценкам экспертов 30% потерь качества обслуживания обусловлено именно кодеками, однако более детальное рассмотрение выходит за рамки данной дипломной работы.

1.8. Сглаживающие буферы

Еще одной проблемой, с которой сталкиваются при передаче речевого трафика, является, так называемая, «рваная» речь. Дело в том, что речевой трафик по характеру является равномерным и непрерывным потоком пакетов. Но при прохождении различных маршрутизаторов на пути следования равномерность может нарушится и если не принять мер по ее восстановлению, то выдача пакетов абоненту будет производится с разной задержкой, т.е. в слышимой абонентом речи будут появляться «провалы».

Чтобы этого не произошло, используется сглаживающий буфер. Пакеты поступают в буфер, хранятся там и передаются кодеку через определенные и равные промежутки времени. Таким образом устраняется вариация задержки, но буфер и сам вносит задержку, которая будет равна суммарной длительности кадров, хранящихся в буфере. Значит, число кадров должно быть небольшим, только чтобы устранить вариации задержки.

Для IP-телефонии в настоящее время предлагается два типа сглаживающих буферов: с фиксированной и с адаптивно изменяющейся задержкой воспроизведения. В первом случае пришедший в буфер пакет хранится там определенное время, но если задержка в сети окажется меньше той, на которую рассчитан буфер, то получится, что пакеты будут просто так, без необходимости задерживаться в буфере, тем самым увеличивая суммарную задержку. Кроме того, если пакет придет после истечения срока задержки, он сбросится, и надо будет применять «маскировку» ошибок, о которой речь пойдет чуть позднее. Во втором случае, при применении адаптивной задержки, величина задержки и ее вариация будут определяться в начале каждого речевого фрейма. В соответствии с этими значениями определяется момент выдачи кадра в декодер. А буфер также можно изменять в зависимости от состояния сети.

1.9. Процедура маскирования

В результате частых перегрузок в IP-сети маршрутизаторы могут сбросить или сильно задержать часть пакетов. В этом случае в выходном сигнале будут возникать пробелы, которые будут восприниматься абонентом как щелчки. Для устранения этого явления применяют процедуру маскирования ошибок. Декодер должен получать кадры через определенные промежутки времени, и если пакет пришел позднее и в тот момент, когда декодеру понадобился следующий фрейм буфер оказался пуст, процедура маскирования ошибок сама сформирует фрейм, копию последнего переданного, чтобы скрыть сбой. Допустимо повторять кадры несколько раз, это не окажет влияния на качество. Дело в том, что голос человека обладает некоторой избыточностью, а органы слуха неспособны выявить сигналы аналогичной частоты переданные друг за другом в короткий промежуток времени. Также можно смоделировать кадр на основе нескольких

предыдущих. Как показывают исследования проведенные рабочей группой TIPHON при потере примерно 3% пакетов качество передачи сохраняется, но при 15%, становится неприемлемым.

1.10. Выводы

На основе материалов данной главы, в котрой расматриваются основные принципы IP-телефонии можно сделать выводыо том, что

- перспективность этого направления обосновывает актуальность темы дипломной работы,
- проблемы качества IP-телефонии являются одним из основных сдерживающих факторов на пути ее внедрения,
- несмотря на существование стандартных алгоритмов кодирования речи и других средств качественной передачи речи по IP, у разработчиков есть огромный простор для деятельности, направленной на дальнейшее совершенствование технологии IP-телефонии.

Глава 2. ОЦЕНКИ КАЧЕСТВА ОБСЛУЖИВАНИЯ

2.1. Качество обслуживания вызовов в традиционной телефонии

Данный параграф является в определенном смысле ключевым в данной дипломной работе и представляет собой попытку анализа эволюции принципов и критериев качества телекоммуникационных услуг, предпринятую автором на свой страх и риск. И, возможно, несколько спорную, но с чрезвычайно интересными наблюдениями.

Итак, с самого начала создания телефонных сетей число обслуживающих устройств было меньше числа пользователей. Естественно при этом одновременное ведение разговора всеми пользователями невозможно. Следовательно возникает необходимость оценки вероятности, что абонент не получит доступак услугам связи. Этим занимается теория телетрафика, задача которой в основном сводится к умению оценивать качество обслуживания поступающих от пользователей запросов как функцию нагрузки сети или ее элементов.

Таким образом для традиционных телефонных сетей уже существует группа задач, связанных с понятием качества обслуживания. Основной вопрос состоял в том, какие потери вызовов или задержки обслуживания могут возникать в системе при той или иной нагрузке и как надо корректировать структуру и повышать ее ресурсы с повышением нагрузки, чтобы и задержки и потери оставались на некотором приемлимом уровне, причем не менее важным был вопрос, как сделать это наиболее экономично. Нужно было решить каким образом оценивать и характеризовать качество обслуживания, найти способы оценки путем расчетов и создать систему норм, которым должны удовлетворять показатели качества обслуживания.

Под качеством услуг понимается степень или мера того, насколько широки и разносторонни услуги телефонной связи, насколько легко и удобно заказать нужную связь, насколько быстро и безотказно предоставляется связь и насколько она полноценна. Данное утверждение одинаково справедливо как для традиционной телефонии, так и для IP-телефонии.

Модель телефонной сети в широком смысле характеризуется двумя признаками качества обслуживания вызовов – отказами в обслуживании (потерями) и задержками (ожиданием). С этих позиций выделяются три класса

моделей телефонной сети или ее элементов:

- системы с потерями
- системы с ожиданием
- комбинированные системы

В практике проектирования сетей и систем традиционной телефонии преимущественно используется первый класс моделей – системы с потерями. Качество обслуживания вызовов в традиционных АТС нормируется вероятностью потерь по вызовам.

Рассмотрим это несколько более подробно. Чтобы получить четкое представление об обслуживании вызовов в данной модели, дадим определения нескольких основных терминов и понятий, имея ввиду уже объявленную выше преемтсвенность подходов традиционной и IP-телефонии.

Требование обслуживания определяется как желание абонента телефонной сети получить связь с другим абонентом или с некоторой службой сети, проявляющееся в виде одной или нескольких попыток вызова. Все попытки возникшие в результате одного требования называются серией, причем первая попытка в серии считается первичной, а все остальные повторными.

Вызов для традиционной телефонии может оказаться удачным или неудачным. Если система отказывает в обслуживании вызову из-за отсутствия ресурсов, то этот вызов называется потерянным, а вызов не получивший отказа, но задержанный в обслуживании, называется ожидающим. Если вызов покидает систему до начала обслуживания, то он называется ушедшим. Ушедшие и потерянные вызовы являются неудачными.

В рекомендациях ITU-T даны два термина: успешный вызов и результативный вызов. Успешным считается вызов в результате которого либо состоялось соединение линии вызывающего с линией вызываемого, либо передача вызывающему абоненту сигнала о занятости вызываемого абонента. Результативным считается вызов, завершившийся разговором. При это надо помнить, что линия может оказаться неверной (например из-за ошибки абонента при наборе номера). Поэтому результативный вызов считается удачным только если его результатом явилось успешное соединение (т.е. соединение с нужной абонентской линией). Также сушествует понятие неадекватно обслуженного вызова

Независимо от конкретного физического смысла и структуры показателей их формальным содержанием являются вероятностные характеристики потерь и/или ожидания. Общепринятым критерием оценки в системах с потерями считается доля потерянных вызовов;этому критерию ставится в соответствие нормируемый показатель вероятность потери вызова, причем норма задается в виде условия:

$$p <= p_н ,$$

где под p понимается рабочая характеристика (оценка вероятности потери вызова в рассматриваемой системе), в под $p_н$ представляет собой то число, которое характеристика не должна превосходить.

Эти вероятности оцениваются для ЧНН – часа наибольшей нагрузки, т.е. непрерывного 60 минутного интервала внутри времени суток, когда нагрузка системы максимальна.

Изложенные выше проблемы качества обслуживания в традиционных сетях упрощенно сводятся к вопросу получит ли пользователь канал для осуществления вызова или нет. Т.е. результатов могло быть только два: «да» и «нет» или 0 и 1. В сравнении с рассматриваемыми дальше аналогичными проблемами качества обслуживания в области IP-телефонии задача вообщем-то напоминала случай с бросанием монеты, рассматриваемый в курсе «Теории вероятностей» нашего ВУЗа.

Согласно изложенной в этом и последующих курсах по теории телетрафика классификации Кендалла имеет место система M/M/m с m обслуживающими приборами и потерями, т.е. с удалением из системы вызовов, заставших все m приборов занятыми.

Качество обслуживания в этой системе определяется вероятностью

$$P_t = P_в = P_п = P_н = \frac{y^n / n!}{\sum_{j=0}^{n} y^j / j!}$$

Здесь P_t – потери по времени;

P_n – вероятность занятия всех n обслуживающих приборов;

$P_в$ – вероятность потерь по вызовам;

$P_н$ – вероятность потерь по нагрузке;

y – интенсивность поступающей нагрузки

n – число обслуживающих приборов.

Формулой потерь Эрланга (а точнее сделанными по ней таблицами) до сих пор пользуются при расчетах оборудования АТС.

Для оценок качества обслуживания IP-телефонии эти «да» и «нет» заменяются на единственное «может быть», но это «может быть» настолько по разному, что требует не только выполнения данной дипломной работы, на и написания диссертации на эту же тему, да и не только этого.

2.2. Подходы к оценке качества IP-телефонии

С учетом изложенных в главе 1 основ и проблем пакетной передачи речи для оценок качества IP-телефонии более подходит вторая из рассмотренных в предыдущем параграфе моделей – система с ожиданием.

Для обслуживания трафика в IP-телефонии выделяются следующие основные характеристики, определяющие качество.

Первая - *время задержки при передаче сигнала*. Предусматриваются следующие градации численных величин задержек:

- 1-й уровень - до 200 мс - отличное качество связи. Для сравнения, в сети ТФОП допустимы задержки до 150-200 мс.
- 2- й уровень - до 400 мс - считается хорошим качеством связи. Но если сравнивать с качеством связи по сетям ТФОП, разница будет видна. Если задержки постоянно удерживаются на верхней границе 2-го уровня (на 400 мс.), то не рекомендуется использовать эту связь для деловых переговоров.
- 3-й уровень - до 700 мс - считается приемлемым качеством связи для ведения неделовых переговоров. Такое качество связи возможно также при передаче пакетов по спутниковой связи.

Качество IP-телефонии попадает под 2-3 уровни, причем невозможно уверенно сказать, что тот или иной провайдер IP-телефонии работает по 2-му уровню, так как задержки с сети Интернет изменчивы. Более точно можно сказать о провайдерах IP-телефонии, работающих по выделенным каналам. Они

попадают под 1-2 уровни. Также необходимо учитывать задержки при кодировании/декодировании голосового сигнала. Средние суммарные задержки при использовании IP-телефонии обычно находятся в пределах 150-250 мс.

В настоящее время для оценки качества передачи речи используются единицы MOS (Mean Opinion Score). Оценка производится по пятибалльной шкале. Также существует стобалльная шкала единиц рейтинга R (Quality Rating). Соединение с качеством R<50 не рекомендуется МСЭ-Т. Высшему качеству R=100 соответствует MOS=4,5. Т.е. для соединений хорошего качества нужно обеспечивать R>70 или MOS>3,5. Все это иллюстрируется на рис.2.1.

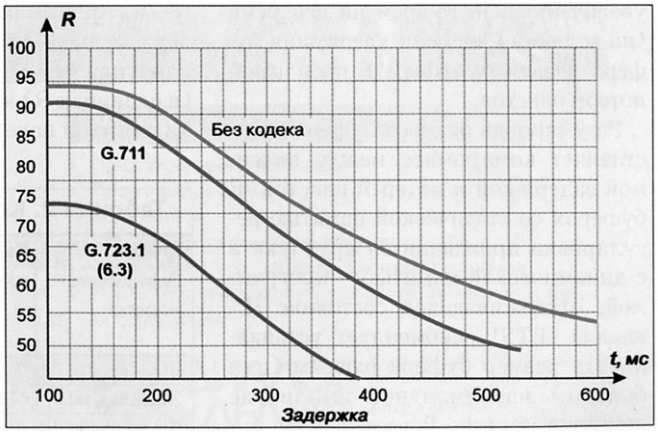


Рис.2.1. Зависимость качества обслуживания R от величины задержки

Другая важная характеристика определяет *качество передаваемого речевого сигнала*, которое зависит от многих факторов. Важны как качество микрофона и колонок абонентов, так и потери IP-пакетов при передаче через канал. Здесь есть важное преимущество перед традиционными сетями: даже в случае загруженности каналов IP и потери части пакетов со сжатым речевым сигналом, программное или аппаратное обеспечение клиента или платформы могут исправить сигнал путем интерполяции соседних пакетов с учетом особенностей речевого спектра. Об этих проблемах кое-что уже было сказано в главе 2, а в следующем параграфе рассмотрим основные метрики качества обслуживания IP-телефонии.

2.3. Измерительные средства для оценки качества передачи речи

Рассмотрим объективные методики измерения качества передачи речи, которые были исследованы и разработаны за последние несколько лет.

2.3.1 Качество передачи речи в конвергированных сетях

О качественных характеристиках в телефонных сетях общего пользования (ТфОП) уже было сказано в предыдущей главе.

Сети IP изначально были спроектированы для поддержки приложений по передаче данных не в реальном масштабе времени, таких как передача файлов или электронная почта. Такие приложения характеризуются неровным трафиком с пиками случайного типа по запросу для большой полосы пропускания, но не зависящей от задержек.

Желание соединить телефонные сети и сети передачи данных в единую сеть на основе передачи пакетов, способную поддерживать интегрированные услуги, требует от новых сетей общего пользования механизмов.

Список некоторых критичных факторов, которые влияют на всеобщий пользовательский опыт телефонных услуг находится в таблице.

Под качеством передачи речи подразумевается точность воспроизведения речи и ее разборчивость - способность извлечения информации из разговора. Эти два признака схожи на первый взгляд, но необязательно зависимы друг от друга.

Четкость тесно связана с разборчивостью передачи речи, которая является субъективным средством измерения количества информации, которую можно извлечь из разговора.

Таблица 1. Влияющие факторы на восприятие качества связи

Факторы качества обслуживания	Факторы качества передачи речи	
Сети ТфОПи IP-телефонии	Общие факторы ТфОП и VoIP	Дополнительные параметры для сетей IP
Телефонные услуги – например, телефонное сединение, дополнительные услуги, переадресация вызова, голосовая почта и т.д. *Доступность* – занятость сети *Надежность* – потеря вызовов, неправильный набор номера *Задержка после набора номера*	Громкость Задержка Эхо Четкость: - *разборчивость* - *шум* - *ослабление* - *переходный разговор*	Четкость задержки-дрожания: - *потеря пакетов* - *полоса пропускания* - *сжатие*

Следующие элементы сети, входящие в раглворный тракт, оказывают влияние на четкость передачи речи:

- Оконечные телефонные устройства на обоих концах могут влиять на четкость передачи речи из-за качества громкоговорителя и микрофона, а также возможности некомпенсированного акустического эха, генерируемого между громкоговорителем и микрофоном. Этот параметр не зависит от типа сети, используемой для передачи речи, но данный фактор очень важен в беспроводных системах связи.
- Шлюз VoIP Gateway (рассматриваемый в предыдущих главах) подключается к ТфОП и преобразовывает цифровой передачи речи по трактам ИКМ в пакетную форму. Элементами шлюза, влияющими на четкость передачи, являются: кодек передачи речи в шлюзе, механизмы возможного подавления паузы, генератор шума и тип перекодирования, который необходим для вызова, например G.711.
- Кроме того, сети IP могут влиять на доставку качества передачи речи, если нет гарантированного QoS, и это является результатом повышенного дрожания и высокой вероятности потерь пакетов.

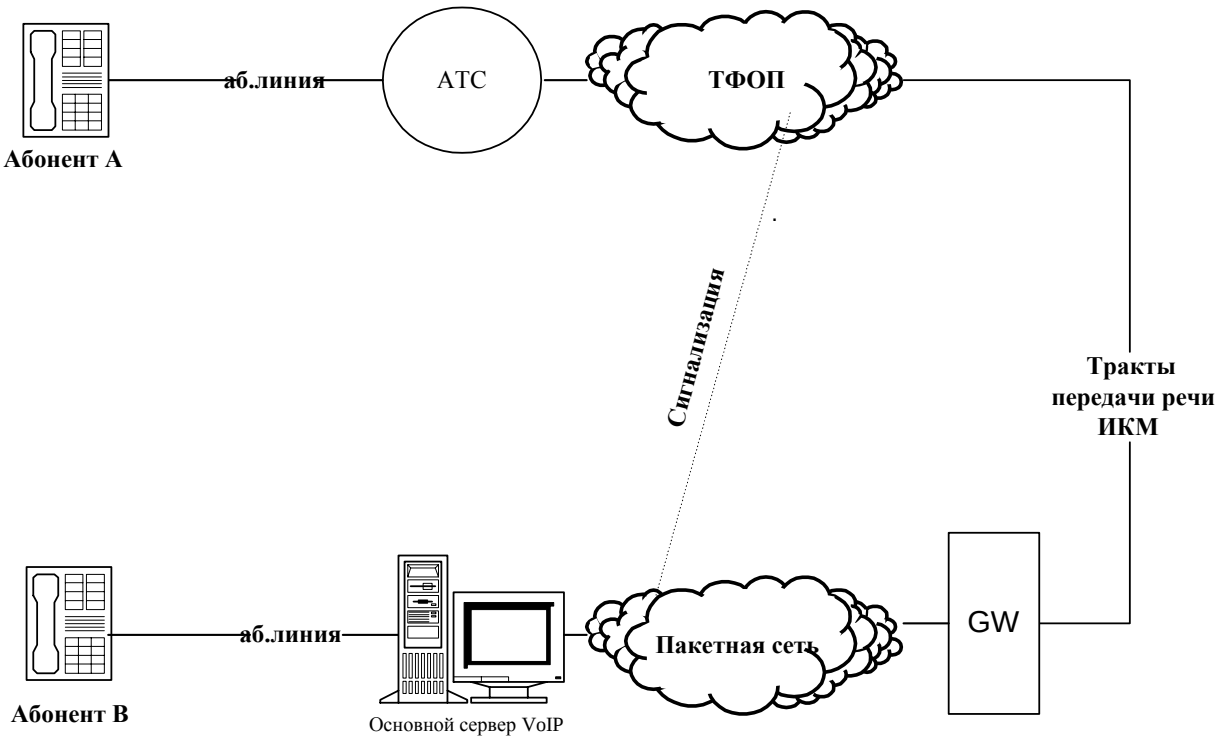


Рис.2.2. Эталонное соединение ТфОП с VoIP для измерения качества передачи речи

В общем случае ***потери пакетов*** нехарактерны для IP-сетей. Тем не менее, когда в процессе передачи пакетов, содержащих речевую информацию, происходит перегрузка, маршрутизаторы могут начинать отбрасывать пакеты. Особенно, в случаях, где мультиплексирование и группирование потоков пакетов реализованы по разному для для разных классов обслуживания, потери пакетов могут вызвать определенные проблемы с качеством обычных услуг телефонной связи.Потеря пакетов не обязательно означает, что пакеты никогда не достигают пункта назначения. Для голосовой телефонии пакет, прибывающий слишком поздно считается потерянным пакетом и отбрасыватся приемником.

Задержка - это время, требуемое для сигнала для пересечения сети. В контексте телефонии, *полная задержка* - это время, требуемое для сгенерированного сигнала от говорящего до слушающего. Полная задержка - это сумма задержек, вносимых всеми устройствами сети и каналами, через которые проходят потоки информации:

1. ***Сетевые задержки.*** В ТфОП задержка определяется прежде всего временем передачи сигналов по междугородным трактам. В пакетных сетях адержка передачи пакетов определяется прежде всего задержками при формированием буферов, очередей и задержками маршрутизации в маршрутизаторах IP.

- Задержка формирования буфера, т.е. сбора передаваемых пакетов - это время, требуемое для приема всего пакета перед его обработкой и продвижением дальше через маршрутизатор. Такая задержка определяется длиной пакета, параметрами функционирования на уровне канала, а также скоростью передачи. Использование передачи коротких пакетов по высокоскоростным трактам уменьшает задержку.
- Задержки коммутации или маршрутизации - это время, затрачееное элементом сети для перенаправления пакета. Это время используется для непосредственного или дифференцированно-косвенного анализа информации заголовков пакетов и оперативной и постоянной информации, хранимой в элементах сети, проверки таблицы маршрутизации и окончательного направления пакета в сторону исходящего порта. Время этой коммутации в элементе сети основано на его архитектуре и оптимизированных реализациях таблиц маршрутизации. Новые коммутаторы IP могут значительно ускорить процесс маршрутизации путем выполнения решений маршрутизации и направления трафика на непосредственно аппаратном уровне . .
- Время нахождения в очереди - зависит от природы статистического мульплексирования в IP-сети и асинхронной природы приема пакетов и формирования очередей из принимаемых пакетов, что требует

дополнительных задержек во входящих и исходящих портах коммутатора пакетов. Величины этихзадержек завися от трафика, длин пакетов и, возможно, статистического распределения трафика по портам коммутации. Резервирование пропускной способности маршрутизатора и увеличение пропускной способности каналов могут уменьшить, но не окончательно устранить эту задержку. Аналитическое определение размера очереди на оконечных точках является весьма сложной задачей из-за присутствия большого числа переменных, определяющих джиттер, в связи с чем ее решение выходит за рамки данного дипломного пректа, но не дальнейшей работы автора.

2. ***Задержки в шлюзах и терминалах.*** Шлюзы VoIP и терминалы VoIP также вносят существенные задержки:

- время обработки речевого сигнала на передающей и принимающей сторонах, включая времякодирования или декодирования сигнала речи из первоначальной аналоговой или цифровой формы в схему кодирования речи, выбранную для данного соединения IP-телефонии и наоборот, добавляется к сетевым задержкам. Большинство применямых для IP-телефонии кодеков также сжимают речевой сигнал, что впоследствии увеличивает задержку из-за необходимых вычислений при восстановлении этого сжатого сигнала. Чем выще сжатие, тем больше битов передачи речи необходимо собрать в буфер, и тем более сложная требуется обработка, тем дольше будет эта задержка.
- На принимающей стороне речевые пакеты передачи речи должны задерживаться для компенсации разброса во временах прибытия пакетов. Об этом явлении дрожания (джжиттера), происходящем из-за различия во временах формирования буфера и очереди пакетов в элементах сети от одной оконечной точки до другой/других, уже говорилось в предыдущей главе. Из этого вытекает необходимость сглаживание дрожания, т.к. кодек передачи речи требует постоянного потока данных без пробелов внутри. Этот компонент задержки может быть уменьшен путем проектирования сети с низким джиттером в каждом узле и даже, возможно, нескольких узлов, но нельзя гарантировать низкий джиттер для области телефонной связи общего пользования, где доминируют многочисленные сетевые элементы, передающие речевые пакеты на относительно короткие расстояния.
- Забегая вперед отметим, что используя механизмы, которые формируют приоритет трафика передачи речи по сравнению с остальным трафиком в пакетной сети, можно также значительно снизить дрожание. Верхней границей такого улучшения является джиттер в сети общего пользования с временным разделением каналов (TDM).

- И наконец, нельзя не сказать о задержке формирования пакетов на передающей стороне. Чем больше размер пакета, тем больше требуется времени.

Используя короткие размеры пакета, уменьшаются задержки, но увеличивается общие расходы, т.к. должны передаваться больше пакетов, содержащие сходную информацию в заголовке. Балансирование между качеством передачи речи, задержкой при формировании пакетов и эффективностью использования полосы пропускания является первостепенной задачей для поставщиков услуг.

Задержку ниже 100 мс пользователи не осознают. Между 100 мс и 300 мс пользователи чувствуют легкое замешательство в ответе собеседника. Выше 300 мс задержка ощутима очевидна пользователям и они должны часто переспрашивать. Задержка выше 300 мс не приемлема в междугородней телефонии, и отклик сохраняется до 100 мс для лучшего качества.

Сказанное иллюстрируется графиками [8] на рисунках 2.3 и 2.4.

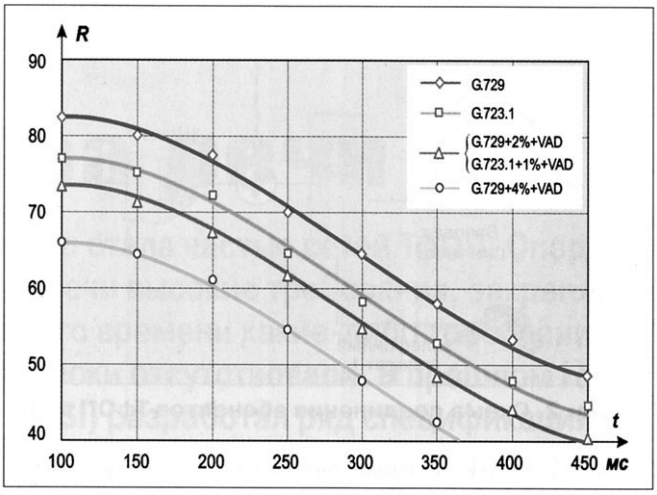


Рис.2.3. Задержки IP-телефонии для различных типов кодеков

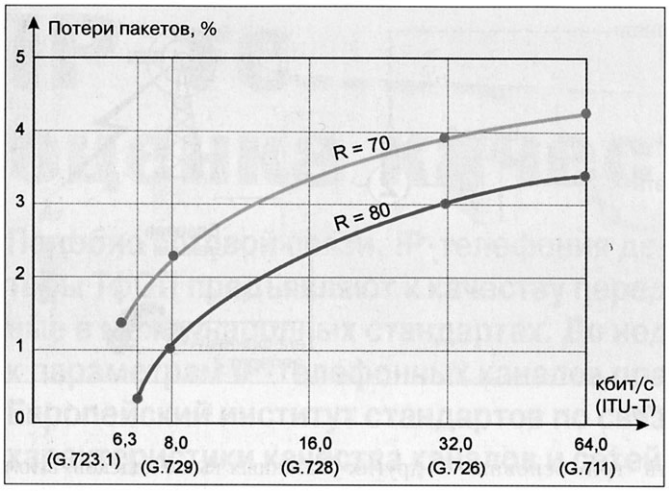


Рис.2.4. Потери пакетов IP-телефонии для различных типов кодеков

2.3.2 Средства измерений качества передачи речи

Существуют лишь несколько уже устоявшихся измеряемых характеристик качества передачи речи IP-телефонии:

1. Потери ячеек или пакетов в сети
   - скорость передачи битов ошибок при потерях ячеек
   - результат управление трафиком в доступе или магистральном канале
2. Снижение характеристики, ощутимой пользователем из-за схем кодирования
3. Задержки пакетов из-за декодирования, формирования пакетов и характеристик сети

Признаки на уровне человеческого восприятия - эхо, четкость передачи речи и видео, задержки передачи речи, фоновый шум

4. Дрожание
5. Определение живучести/неживучести при подключении оконечных точек

Восстановление магистрали с помощью методологии «живого» тестирования, оценивающей быстроту распознования нарушений, вносимых каким-либо элементом, быстроту нахождения параллельного обхода или другого средства, обеспечивающего возобновление устойчивой услуги телефонной связи.

2.3.3 О измерении задержки

Как уже было сказано выше, полная задержка - это сумма задержек в различных устройствах сети и в каналах сети, через которые проходят трафик передачи речи, включая обычные ТфОП и сети VoIP. Задержка не влияет на разборчивость, а скорее на характер разговора.

Простое измерение задержки в сети VoIP весьма затруднительно и требует рассмотрения множества вариантов. Задержки в сетях VoIP могут изменяться по времени из-за многих причин. Особенно в топологиях, где пакетам IP разрешаются выбор различных маршрутов, важно знать, разрешена ли альтернативная маршрутизация в течение периодов занятости в сети. Если динамика в определенный период времени вызывает необходимость выбора альтернативных маршрутов, то на качество передачи речи может влиять результаты конфигурации сети, алгоритмы альтернативной маршрутизации в неоднородных административных областях, и другие причины, которые вызывают выбор трактов с использованием различных критериев в течение определенных периодов времени. Кроме того, шлюзы в трактах соединения могут динамически устанавливать свои буферы дрожания, которые в будущем окажут влияние на задержку. Анализ тенденции задержки позволяет нам определить среднюю задержку на сети, также как охарактеризовать испытуемые

максимум и минимум задержки. Это может быть полезно для характеристики поведения сети под изменяющейся нагрузкой. Также важно измерение задержки независимо в обоих направлениях, т.к. тракт в сети может быть различным для каждого направления.

2.3.4. О измерении активности передачи речи

Измерение активности сигнала передачи речи показывает сколько (40-50%) времени речь не присутствует в одном направлении канала. Эффект VAD - это снижение эффективной полосы пропускания телефонного соединения. Например, если функция формирования пакетов речи вырабатывает 80 байт цифровизированного и сжатого голоса каждые 10мс, она потребует 8Кбит/с полосы пропускания, не считая дополнительных расходов на передачу и протокол канального уровня. С помощью подавления паузы, этот трафик можно снизить до 5Кбит/с либо путем исключения передачи битов во время пауз, либо путем передачи/декодирования фонового шума на низкой скорости. В любом случае требуется, чтобы принимающая сторона создавала фоновый шум во время пауз на удаленном конце для того, чтобы не создавалось впечатления, что соединение оборвалось

Другим важным фактором в этом контексте является взаимосвязь между принятым фоновым шумом и комфортным шумом, т.к. изменения в фоновом шуме могут быть назойливыми. Соотношение между двумя типами шумов можно измерить и вычислить с помощью сравнения частотных областей фона и комфортного шума.

2.3.5 Измерение эха

Этот тип измерения определяет эффект эха в сети. Если односторонняя задержка может сохраняться в пределах 16-20 мс, то такое эхо добавляет приятный дополнительный фон к разговору и дает ощущение «жизни» в телефонной линии. Некоторые виды эха могут быть терпимыми и в диапазоне выше 20 мс, но уже начиная с этих величин и выше необходимо управлять эхом.

Во время измерения эха в сетях VoIP, использующих низкоскоростные кодеки, обычные измерения эха с тонами или шумами уже не применяются. Вместо этого имеется более усложненный метод, который использует реальную речь для определения функцональных возможностей эхозаградителя. Соответсвующее полученное эхо с оценкой PSQM позволяетпроанализировать насколько полученное эхо являлось мешающим конечным пользователям.

Существуют два основных объекта измерений при анализе эхозаградителей:

- отражение эха во время разговора одного собеседника
- снижение сигнала во время разговора двух или более собеседников

В случае разговора одного человека измеряется возвращаюшееся отраженное эхо. Необходимо определить следующие параметры:

- *Величина эха в речи*: определяется максимальный уровень отношения сигнала эхо/речь. Когда эхо при передаче речи достаточно велико, это будет учтено и будет также отражено в искажении и снижении числа PSQM, следовательно, его величина будет повышаться. Этот параметр позволяет пользователю соотнести величину эха в человеческом восприятии с общим качеством передачи речи.
- *Величина эха в паузе:* определяет уровень сигнала эхо/пауза. Эхо в паузе является более ощутимым параметром для пользователя. Эхо, которое может быть незаметным во время разговора, становится очевидным во время паузы, даже на средних уровнях.
- *Процентное соотношение речи без эха*: показывает, продолжительность разговора, искаженнго эхом, и может использоваться для опеределения общих затрат пользователя во время соединения.

Все вышеизложенные факторы могут снижать качество передачи речи до уровня, который становится неприменимым для телефонной связи. Это и обуславливает необходимость анализа качества обслуживания (QoS), краткий перечень оценок которого приведен в данной главе.

Глава 3. МЕХАНИЗМЫ ОБЕСПЕЧЕНИЯ QoS

3.1. Сетевые аспекты обеспечения качества обслуживания IP-телефонии

IP-телефонию часто считают частью пакета услуг Интернет-провайдера, что не верно. Уже известно множество примеров внедрения IP-телефонии в корпоративные сети и даже построения выделенных сетей IP-телефонии. Причем проблемы при построении сети могут быть самые разные, ведь это может быть и корпоративная сеть, и сеть традиционного оператора, и отдельная выделенная сеть, или что-то еще. В каждом случае нужны свои решения и подходы. Немаловажен и тип передаваемой по этой сети информации, сеть можно использовать только для передачи речи, а могут передаваться и данные. Важно учитывать характер взаимодействия различных узлов IP-телефонии и обеспечивать минимальные задержки и минимальный уровень потерь. Иногда приходится сокращать полосу пропускания.

Упрощенно резюмируя материал предыдущей главы можно сказать, что полная временная задержка речевого трафика делиться на две основные части: задержки на кодирование и декодирование на шлюзах, и задержки вносимые самой сетью

Уменьшить общую задержку можно двумя путями, во-первых, спроектировать инфраструктуру сети таким образом, чтобы задержка в ней была минимальной, а, во-вторых, уменьшить время обработки данных в речевом шлюзе.

Для уменьшения задержки в сети нужно сокращать число транзитных участков между маршрутизаторами, а в наиболее важных местах сети использовать высокоскоростные каналы. А для уменьшения разброса задержек можно использовать эффективные методы управления трафиком, например механизмы резервирования.

Как правило, с корпоративными сетями все выглядит достаточно просто. Они имеют ограниченные размеры, контролируемую топологию, а характер трафика обычно заранее известен. Однако, возьмем простой пример: речь передается по существующей ЛВС, которая слишком загружена, чтобы обеспечивать приемлемое качество обслуживания.

Решением этой проблемы будет изоляция серверов и клиентов данного типа трафика и ресегментация сети. Разбить сеть на сегменты можно или установив коммутатор Ethernet, или добавив порты в маршрутизатор.

Выделенные сети IP-телефонии обычно используются для междугородной и международной связи. Такие сети лучше строить по принципу многоуровневой иерархической сети, где на каждый уровень возлагаются свои определенные функции. На входе в сеть главное обеспечить подключение речевых шлюзов, а внутри сети – высокоскоростную пересылку трафика. В такой сети очень просто осуществляется расширение и внедрение новых услуг и служб. Проблема проектирования также не доставляет особых хлопот: характер трафика определен, полоса пропускания также легко рассчитывается. Трафик однотипный, а значит не нужна приоритизация пакетов.

В сетях традиционных операторов передается трафик различных видов, по этому для обеспечения приемлимого качества передачи предлагается использование модели Diff-Serv.

3.2. Модель Diff-Serv

Предоставляет возможность согласованной обработки различных классов трафика. В соответствии с этой моделью, разработанной группой IETF Diff-Serv, байт ToS (Type of Service) в заголовке IP-пакета получил для модели Diff-Serv название байта DS, шесть битов которго отведены под код Diff-Serv. Каждому значению этого кода соответствует свой класс пересылки PHB (Per-Hop Behavior Forwarding Class), определяющий ожидаемый уровень обслуживания. В рамках каждого класса пакеты должны обрабатываться в соответствии сопределенными для него требованиями к качеству обслуживания. Трафик разделяется на ограниченное число классов или «групп поведения» (behavior), что обеспечивает дальнейшую возможность масштабируемой дифференциации услуг.

Модель Diff-Serv описывает архитектуру сети как совокупность пограничных участков и ядра сети. Поступающий в сеть трафик классифицируется и нормируется пограничными маршрутизаторами. Нормирование трафика предусматривает измерение его параметров, проверку соответствия заданным правилам предоставления услуг,

профилирование (при этом пакеты, не укладывающиеся в рамки установленных правил, могут быть отсеяны) и другие операции. В ядре магистральные маришрутизаторы или коммутаторы пересылают трафик в соответствии с классом PHB, код которого указан в поле DS.

Достоинства модели Diff-Serv состоят в том, что она, во-первых, обеспечивает единое понимание того, как должен обрабатываться трафик определенного класса, а во-вторых, позволяет разделить весь трафик на относительно небольшое число классов, вместо того чтобы отдельно анализировать каждый поток.

К настоящему времени определены два класса трафика в рамках модели Diff-Serv:

- Срочная доставка (Expedited Forwarding PHB Group)
- Гарантированная доставка (Assured Forwarding PHB Group).

Механизм обеспечения QoS на уровне сетевого устройства, применяемый в Diff-Serv, включает в себя четыре операции. Сначала на основании информации заголовка, относящейся к уровням со 2-го по 4-й, пакеты классифицируются. Затем они маркируются в соответствии с произведенной классификацией и данными битов Diff-Serv. В зависимости от маркировки выбирается алгоритм передачи трафика (при необходимости – с выборочным удалением пакетов), позволяющий избежать заторов. Формирование трафика чаще всего состоит в организации очередей с учетом приоритетов.

Хотя эта модель не гарантирует 100% качество обслуживания, у нее есть серьезные преимущества. Например, нет необходимости в организации предварительного соединения и резервирования ресурсов. А так как при использовании Diff-Serv используется небольшое, фиксированное количество классов обслуживания и трафик абонентов распределяется по общим очередям, не требуются высокая производительность.

3.3. Протокол резервирования ресурсов RSVP – ReSource Reservation Protocol.

Для того, чтобы удовлетворить требования IP-телефонии и другим аудио и видео приложений к пропускной способности и качеству

обслуживания трафика (задержка, величина джиттера), необходим механизм, позволяющий приложениям информировать сеть о своих требованиях. На основе данной информации сеть может резервировать внутри себя ресурсы для того, чтобы гарантировать выполнение требований, или отказать приложению, заставляя его пересмотреть первоначальные требования к качеству услуг или прекратить установление соединений, отложив сеанс связи. В роли такого механизма и выступает протокол резервирования ресурсов RSVP (Resource Reservation Protocol).

При одноадресной передаче процесс резервирования выглядит довольно просто. Многоадресная же рассылка ставит гораздо более сложные задачи по резервированию ресурсов. Потенциально приложения, использующие многоадресную рассылку, могут генерировать огромные объемы трафика- например, в случае организации видеоконференции с большой и рассредоточенной группой получателей.

Однако трафик в данной ситуации (видеоконференция) может быть значительно снижен.

Во-первых, некоторые члены группы могут не нуждаться в доставке данных от конкретного источника в определенный период времени - например, получатель может быть заинтересован в получении только аудио - информации и не заинтересован в получении видео.

Во - вторых, оконечное оборудование некоторых членов группы может быть в состоянии обрабатывать только часть передаваемой информации. Например, поток видеоданных может состоять из двух компонентов - базового и дополнительного для повышения качества изображения. Оборудование некоторых получателей может не иметь достаточной вычислительной мощности для обработки компонентов с высоким разрешением или может быть подключено к сети через канал, не обладающий необходимой пропускной способностью для пропуска всего сигнала.

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 усеченных версий потоков отдельным получателям

RSVP – это протокол сигнализации, который обеспечивает резервирование ресурсов и управление ими с целью предоставления интегрированных сервисов, предназначенных для эмуляции выделенных каналов в IP-сетях. Резервирование ресурсов позволяет маршрутизаторам заранее определить в состоянии ли они осуществить доставку многоадресного трафика всем получателям.

RSVP позволяет системам запрашивать сервисы у сети, например: гарантированную пропускную способность, предсказуемую задержку, максимальный уровень потерь.

Сообщения пути RSVP рассылаются отправителем и отслеживают маршрут, оставляя указатели на маршрутизаторах. Это позволяет производить резервирование по пути передачи. На маршрутизаторах сообщения о резервировании объединяются при их возвращении к источнику, и в итоге отправитель получает только одно сообщение от ближайшего маршрутизатора. Но резервирование выполняется только если есть гарантия (т.е. ресурс достаточен).

В основе RSVP лежат три концепции:

- Сеанс – поток данных, идентифицируемый по адресату;
- Спецификация потока – определяет требуемое качество услуг и используется узлом для задания параметров планировщика пакетов;
- Спецификация фильтра – определяет набор пакетов, под которые запрашивает ресурс.

RSVP не определяет содержание спецификации потока, он просто передает запрос. Спецификация потока включает класс услуг: Rspec – резерв R – определяет требуемое качество услуг и Tspec – трафик T – описывает поток данных.

3.3.1. Процедура резервирования ресурсов по протоколу RSVP

Источник данных посылает по уникальному или групповому адресам получателя специальное сообщение *Path,* в котором он указывает рекомендуемые параметры для качественного приема своего трафика: верхней и нижней границ пропускной способности, задержки и вариации задержки. Сообщение *Path* передается маршрутизаторами сети в

направлении к приемнику с использованием таблиц маршрутизации роутеров.

Каждый поддерживающий протокол RSVP маршрутизатор, получив сообщение *Path*, фиксирует «состояние пути», которое включает предыдущий адрес источника сообщения *Path*. В сети образуется фиксированный маршрут передачи сообщений в рамках сессии RSVP.

Сообщение Path должно нести в себе шаблон отправителя (Sender Template), который описывает формат пакетов данных, посылаемых отправителем. Этот шаблон имеет форму спецификации фильтра, которая может использоваться для отделения пакетов данного отправителя от других пакетов в пределах сессии. Кроме того, сообщения Path должны содержать спецификацию отправителя Tspec, которая определяет характеристики информационного трафика, формируемого отправителем. Спецификация Tspec используется для предотвращения избыточного резервирования. Также сообщение Path может нести в себе пакет данных оповещения OPWA, известный, как "Adspec". Пакет Adspec, полученный с сообщением Path, передается системе управления трафиком, которая присылает скорректированную версию Adspec. Последняя пересылается далее в виде сообщения Path. Сообщения Path посылаются с теми же адресами отправителя и получателя, что и данные, так что они будут корректно маршрутизироваться даже через сетевые области, не поддерживающие RSVP.

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 опционно UDP/TCP порт, с учетом идентификатора протокола, заданного для сессии.

После получения сообщения *Path* приемник отправляет маршуризатору, от которого он получил это сообщение, запрос резервирования ресурсов *Resv* (reservation request). В дополнение к информации *Tspec* сообщение *Resv* включает спецификацию запроса (*Rspec*), в которой указываются требуемые приемнику параметры качества обслуживания, и спецификацию фильтра (filterspec), определяющую, к каким пакетам сессии применяется данное резервирование. Вместе Rspec и filterspec представляют собой дескриптор потока, который маршрутизатор

использует для идентификации каждой новой процедуры резервирования ресурсов.

Когда получатель данных отправляет запрос резервирования, он может запросить посылку ответного сообщения, подтверждающего резервирование.

При получении сообщения *Resv* каждый поддерживающий RSVP маршрутизатор вдоль восходящего пути применит две процедуры для определения приемлемости указанных в запросе параметров резервирования. С помощью процедуры управления доступом (admission control) проверяется, имеются ли у маршрутизатора ресурсы, необходимые для поддержание запрашиваемого уровня QoS, а с помощью процедуры администрирования (policy control) – имеет ли пользователь право на резервирование ресурсов. Если запрос не может быть удовлетворен, маршрутизатор возвращает сообщение об ошибке отправителю. Если же запрос принимается, то маршрутизатор отправляет сообщение *Resv* вверх по пути следующему маршрутизатору, а данные о требуемом QoS передаются механизмам маршрутизатора, ответственным за управление трафиком. Сообщение *Resv*, переадресованное предшествующему узлу, несет в себе спецификацию *flowspec*, которая содержит информацию из всех *flowspec*, запрошенных последующими узлами, которым будут посылаться данные.

Так как *flowspecs* непрозрачны для RSVP, действительные правила для сравнения *flowspecs* должны быть определены и реализованы вне рамок этого протокола. Реализация RSVP потребует обращения к специальной программе для выполнения объединения спецификаций *flowspec*.

Прием запроса резервирования означает также обработку праметорв QoS соответствующими механизмами маршрутизатора. Конкретный способ обработки параметров QoS маршрутизатором в протоколе RSVP не описывается.

Когда последний маршрутизатор получает сообщение *Resv* и принимает запрос, то он посылает подтверждающие сообщение назад узлу-приемнику (последний маршрутизатор располагается ближе всего к отправителю, а для групповых потоков – в точке слияния резервирования). При выполнении группового резервирования учитывается тот факт, что в

точках разветвления дерева доставки несколько резервируемых потоков сливаются в один.

После установления резервирования источник начинает отправлять данные, которые обслуживаются на всем пути к приемнику с заданным качеством обслуживания.

Для простой выделенной линии, желаемый QoS будет получен с помощью диспетчера пакетов в драйвере канального уровня. Если технология канального уровня поддерживает свои средства управления QoS, тогда RSVP должен согласовать с канальным уровнем получение требуемого QoS.

Отменить резервирование можно двумя путями: прямо или косвенно. В первом случае отмена идет по инициативе источника или приемника с помощью специальных сообщений RSVP. Во втором – по тайм-ауту: состояние резервирования имеет срок жизни .

RSVP поддерживает улучшенную версию однопроходного варианта алгоритма, известного под названием OPWA (One Pass With Advertising). С помощью OPWA управляющие пакеты RSVP, посланные вдоль маршрута для сбора данных, которые могут быть использованы для предсказания значения QoS маршрута в целом. Рассмотрение алгоритма OPWA в рамках разрабатываемой в дипломной работе функциональной модели не предусматривается заданием на дипломное проектирование, поэтому более детально этот вопроос здесь не рассматривается.

С точки зрения хоста (узла маршрутизации, рабочей станции) работа RSVP выглядит так:

1. Получатель вступает в группу многоадресной рассылки посредством отправки сообщения по протоколу IGMP соседнему маршрутизатору;
2. Потенциальный отправитель отправляет сообщение по адресу группы;
3. Получатель принимает сообщение Path, идентифицирующее отправителя;
4. Теперь Получатель имеет информацию об обратном пути и может отправлять сообщение Resv с дескрипторами потока;
5. Сообщения Resv передаются по сети отправителю;

6. Отправитель начинает передачу данных;
7. Получатель начинает передачу данных.

Несмотря на то, что RSVP является важным инструментом в арсенале QoS, этот протокол не может решить все проблемы, связанные с QoS. RSVP имеет три недостатка: большой объем, управление допуском и время, требуемое для установления оконечного резервирования.

RSVP не ориентирован на крупномасштабные сети. В крайнем случае , магистральный маршрутизатор может управлять несколькими тысячами резервирований по RSVP и соответствующими очередями для каждого потока, но сложность и требуемая производительность для такого тотального управления являются сдерживающими факторами.

RSVP работает с фиксированным размером IP-пакета без учета дополнительных схем сжатия, циклического контроля с введением избыточности (CRCs) или механизмов формирования данных в канале (Frame Relay, PPP или HDLC). Например, во время использования RSVP и G.729 для VoIP, резервирование запроса для ПО Cisco IOS является 24 Кбит/с, по сравнению с действительным значением около 11 Кбит/с при использовании cRTP (RTP Header Compression). Другими словами, в тракте 56 Кбит/с разрешены только два резервирования 24 Кбит/с, даже если полоса пропускания доступна на три потока VoIP 11 Кбит/с.

Возможна подписка доступных полос пропускания тракта с целью разрешения RSVP резервировать больше полос пропускания, чем действительно доступных. Например, в тракте 56 Кбит/с установка полосы пропускания сообщает интерфейсу, что уже существует полоса пропускания 100 Кбит/с. Можно использовать RSVP для разрешения 75% доступной полосы пропускания использовать RSVP трафик. Такой сценарий разрешает RSVP резервировать необходимую полосу пропускания для трех вызовов стандарта G.729 VoIP. Очевидным фактом является присущая этому опасность, т.к. если не используется cRTP, то линия перегружена.

Более подробно эти примеры и спецификации RSVP в целом будут рассмотрены в следующей главе дипломной работы.

3.4. Протокол MPLS

MPLS (MultiProtocol Label Switching) — это технология быстрой коммутации пакетов в многопротокольных сетях, основанная на использовании меток. MPLS разрабатывался и продолжает разрабатываться как способ построения высокоскоростных IP-магистралей, однако область ее применения не ограничивается протоколом IP, а распространяется на трафик любого маршрутизируемого сетевого протокола.

Архитектура MPLS обеспечивает построение магистральных сетей, имеющих практически неограниченные возможности масштабирования, повышенную скорость обработки трафика и беспрецедентную гибкость с точки зрения организации дополнительных сервисов. Кроме того, технология MPLS позволяет интегрировать сети IP и ATM, за счет чего поставщики услуг смогут не только сохранить средства, инвестированные в оборудование асинхронной передачи, но и извлечь дополнительную выгоду из совместного использования этих протоколов.

За развитие архитектуры MPLS отвечает рабочая группа с одноименным названием, входящая в секцию по маршрутизации консорциума IETF. В деятельности группы принимают активное участие представители крупнейших поставщиков сетевых решений и оборудования. Эта архитектура выросла из системы Tag Switching, предложенной Cisco Systems, однако некоторые идеи были заимствованы у конкурирующей технологии IP-коммутации, созданной компанией Ipsilon, и проекта ARIS корпорации IBM. В архитектуре MPLS собраны наиболее удачные элементы всех упомянутых разработок, и вскоре она должна превратиться в стандарт Internet благодаря усилиям IETF и компаний, заинтересованных в скорейшем продвижении данной технологии на рынок.

3.4.1 Принцип коммутации

В основе MPLS лежит принцип обмена меток. Любой передаваемый пакет ассоциируется с тем или иным классом сетевого уровня (Forwarding Equivalence Class, FEC), каждый из которых идентифицируется определенной меткой. Значение метки уникально лишь для участка пути между соседними узлами сети MPLS, которые называются также маршрутизаторами, коммутирующими по меткам (Label Switching Router, LSR). Метка передается в составе любого

пакета, причем способ ее привязки к пакету зависит от используемой технологии канального уровня.

Маршрутизатор LSR получает топологическую информацию о сети, участвуя в работе алгоритма маршрутизации — OSPF, BGP, IS-IS. Затем он начинает взаимодействовать с соседними маршрутизаторами, распределяя метки, которые в дальнейшем будут применяться для коммутации. Обмен метками может производиться с помощью как специального протокола распределения меток (Label Distribution Protocol, LDP), так и модифицированных версий других протоколов сигнализации в сети (например, незначительно видоизмененных протоколов маршрутизации, резервирования ресурсов RSVP и др.).

Распределение меток между LSR приводит к установлению внутри домена MPLS путей с коммутацией по меткам (Label Switching Path, LSP). Каждый маршрутизатор LSR содержит таблицу, которая ставит в соответствие паре «входной интерфейс, входная метка» тройку «префикс адреса получателя, выходной интерфейс, выходная метка». Получая пакет, LSR по номеру интерфейса, на который пришел пакет, и по значению привязанной к пакету метки определяет для него выходной интерфейс. (Значение префикса применяется лишь для построения таблицы и в самом процессе коммутации не используется.) Старое значение метки заменяется новым, содержавшимся в поле «выходная метка» таблицы, и пакет отправляется к следующему устройству на пути LSP.

Вся операция требует лишь одноразовой идентификации значений полей в одной строке таблицы. Это занимает гораздо меньше времени, чем сравнение IP-адреса отправителя с наиболее длинным адресным префиксом в таблице маршрутизации, которое используется при традиционной маршрутизации.

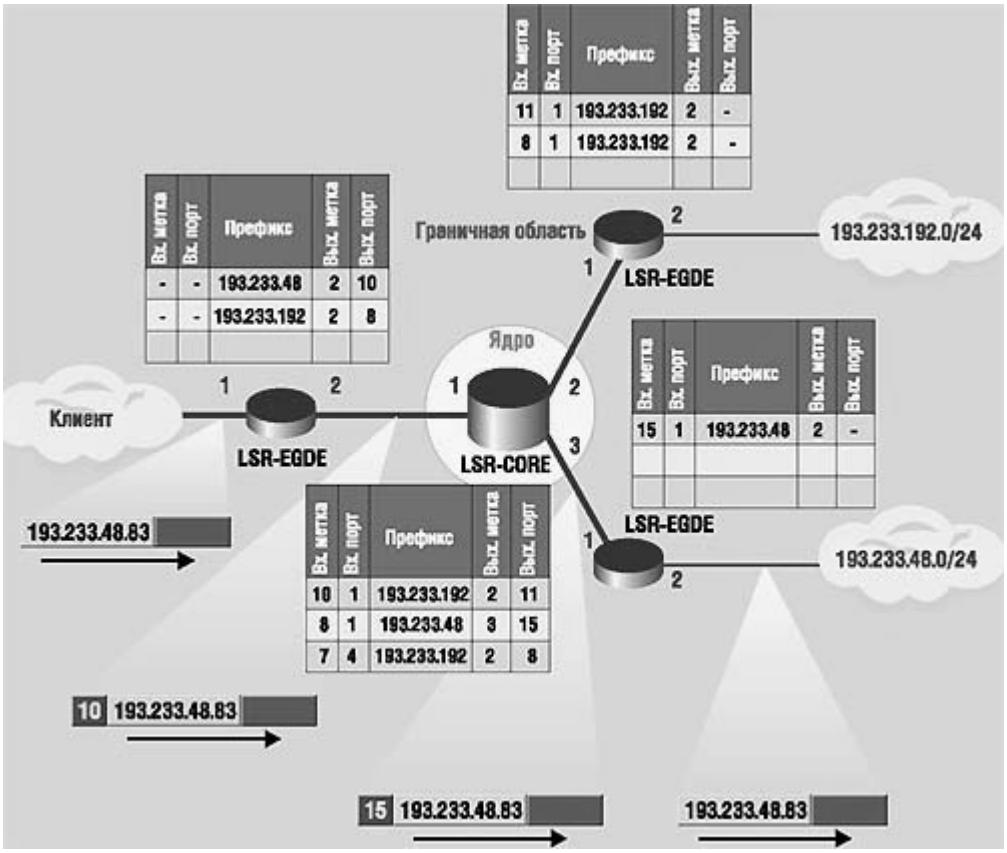


Рис 3.1: Схема коммутации MPLS

Сеть MPLS делится на две функционально различные области — ядро и граничную область (рис.3.1). Ядро образуют устройства, минимальным требованием к которым является поддержка MPLS и участие в процессе маршрутизации трафика для того протокола, который коммутируется с помощью MPLS. Маршрутизаторы ядра занимаются только коммутацией. Все функции классификации пакетов по различным FEC, а также реализацию таких дополнительных сервисов, как фильтрация, явная маршрутизация, выравнивание нагрузки и управление трафиком, берут на себя граничные LSR. В результате интенсивные вычисления приходятся на граничную область, а высокопроизводительная коммутация выполняется в ядре, что позволяет оптимизировать конфигурацию устройств MPLS в зависимости от их местоположения в сети.

Таким образом, главная особенность MPLS — отделение процесса коммутации пакета от анализа IP-адресов в его заголовке, что открывает ряд привлекательных возможностей. Очевидным следствием описанного подхода

является тот факт, что очередной сегмент LSP может не совпадать с очередным сегментом маршрута, который был бы выбран при традиционной маршрутизации.

Поскольку на установление соответствия пакетов определенным классам FEC могут влиять не только IP-адреса, но и другие параметры, нетрудно реализовать, например, назначение различных LSP пакетам, относящимся к различным потокам RSVP или имеющим разные приоритеты обслуживания. Конечно, подобный сценарий удается осуществить и в обычных маршрутизируемых сетях, но решение на базе MPLS оказывается проще и к тому же гораздо лучше масштабируется.

Каждый из классов FEC обрабатывается отдельно от остальных — не только потому, что для него строится свой путь LSP, но и в смысле доступа к общим ресурсам (полосе пропускания канала и буферному пространству). В результате технология MPLS позволяет очень эффективно поддерживать требуемое качество обслуживания, не нарушая предоставленных пользователю гарантий. Применение в LSR таких механизмов управления буферизацией и очередями, как WRED, WFQ или CBWFQ, дает возможность оператору сети MPLS контролировать распределение ресурсов и изолировать трафик отдельных пользователей.

Использование явно задаваемого маршрута в сети MPLS свободно от недостатков стандартной IP-маршрутизации от источника, поскольку вся информация о маршруте содержится в метке и пакету не требуется нести адреса промежуточных узлов, что улучшает управление распределением нагрузки в сети.

3.4.2. Метки и способы маркировки

Метка — это короткий идентификатор фиксированной длины, который определяет класс FEC. По значению метки пакета определяется его принадлежность к определенному классу на каждом из участков коммутируемого маршрута.

Как уже отмечалось, метка должна быть уникальной лишь в пределах соединения между каждой парой логически соседних LSR. Поэтому одно и то же ее значение может использоваться LSR для связи с различными соседними маршрутизаторами, если только имеется возможность определить, от какого из

них пришел пакет с данной меткой. Другими словами, в соединениях «точка—точка» допускается применять один набор меток на интерфейс, а для сред с множественным доступом необходим один набор меток на модуль или все устройство. В реальных условиях угроза исчерпания пространства меток очень маловероятна.

Перед включением в состав пакета метка определенным образом кодируется. В случае использования протокола IP она помещается в специальный «тонкий» заголовок пакета, инкапсулирующего IP. В других ситуациях метка записывается в заголовок протокола канального уровня или кодируется в виде определенного значения VPI/VCI (в сети ATM). Для пакетов протокола IPv6 метку можно разместить в поле идентификатора потока.

3.4.3. Стек меток

В рамках архитектуры MPLS вместе с пакетом разрешено передавать не одну метку, а целый их стек. Операции добавления/изъятия метки определены как операции на стеке (push/pop). Результат коммутации задает лишь верхняя метка стека, нижние же передаются прозрачно до операции изъятия верхней. Такой подход позволяет создавать иерархию потоков в сети MPLS и организовывать туннельные передачи. Стек состоит из произвольного числа элементов, каждый из которых имеет длину 32 бита: 20 бит составляют собственно метку, 8 отводятся под счетчик времени жизни пакета, один указывает на нижний предел стека, а три не используются. Метка может принимать любое значение, кроме нескольких зарезервированных.

3.4.4. Компоненты коммутируемого маршрута

Коммутируемый путь (LSP) одного уровня состоит из последовательного набора участков, коммутация на которых происходит с помощью метки данного уровня (рис. 2). Например, LSP нулевого уровня проходит через устройства LSR 0, LSR 1, LSR 3, LSR 4 и LSR 5. При этом LSR 0 и LSR 5 являются, соответственно, входным (ingress) и выходным (egress) маршрутизаторами для пути нулевого уровня. LSR 1 и LSR 3 играют ту же роль для LSP первого уровня; первый из них производит операцию добавления метки в стек, а второй — ее изъятия. С точки зрения трафика нулевого уровня, LSP первого уровня является прозрачным туннелем. В любом сегменте LSP можно выделить

верхний и нижний LSR по отношению к трафику. Например, для сегмента «LSR 4 — LSR 5» четвертый маршрутизатор будет верхним, а пятый — нижним.

3.4.5. Привязка и распределение меток

Под привязкой понимают соответствие между определенным классом FEC и значением метки для данного сегмента LSP. Привязку всегда осуществляет «нижний» маршрутизатор LSR, поэтому и информация о ней распространяется только в направлении от нижнего LSR к верхнему. Вместе с этими сведениями могут предаваться атрибуты привязки.

Обмен информацией о привязке меток и атрибутах осуществляется между соседними LSR с помощью протокола распределения меток. Архитектура MPLS не зависит от конкретного протокола, поэтому в сети могут применяться разные протоколы сетевой сигнализации. Очень перспективно в данном отношении — использование RSVP для совмещения резервирования ресурсов и организации LSP для различных потоков.

Существуют два режима распределения меток: независимый и упорядоченный. Первый предусматривает возможность уведомления верхнего узла о привязке до того, как конкретный LSR получит информацию о привязке для данного класса от своего нижнего соседа. Второй режим разрешает высылать подобное уведомление только после получения таких сведений от нижнего LSR, за исключением случая, когда маршрутизатор LSR является выходным для этого FEC.

Распространение информации о привязке может быть инициировано запросом от верхнего устройства LSR (downstream on-demand) либо осуществляться спонтанно (unsolicited downstream).

Глава 4. ФУНКЦИОНАЛЬНАЯ МОДЕЛЬ RSVP.

4.1. Основные принципы

Продолжим рассмотрение протокола RSVP, начатое в параграфе 3.2 предыдущей главы. Там уже было отмечено, что RSVP запрашивает ресурсы только для одного из направлений трафика и только по указанию получателя, работая поверх IPv4 или IPv6. Из материала параграфа 3.2 также следует, что протокол RSVP относится к числу управляющих, а не транспортных. Суть описания RSVP в параграфе 3.2 связана с сетевым аспектом модели RSVP.

Теперь перейдем к структурному аспекту функциональной модели процесса RSVP, показанному на рис.4.1.

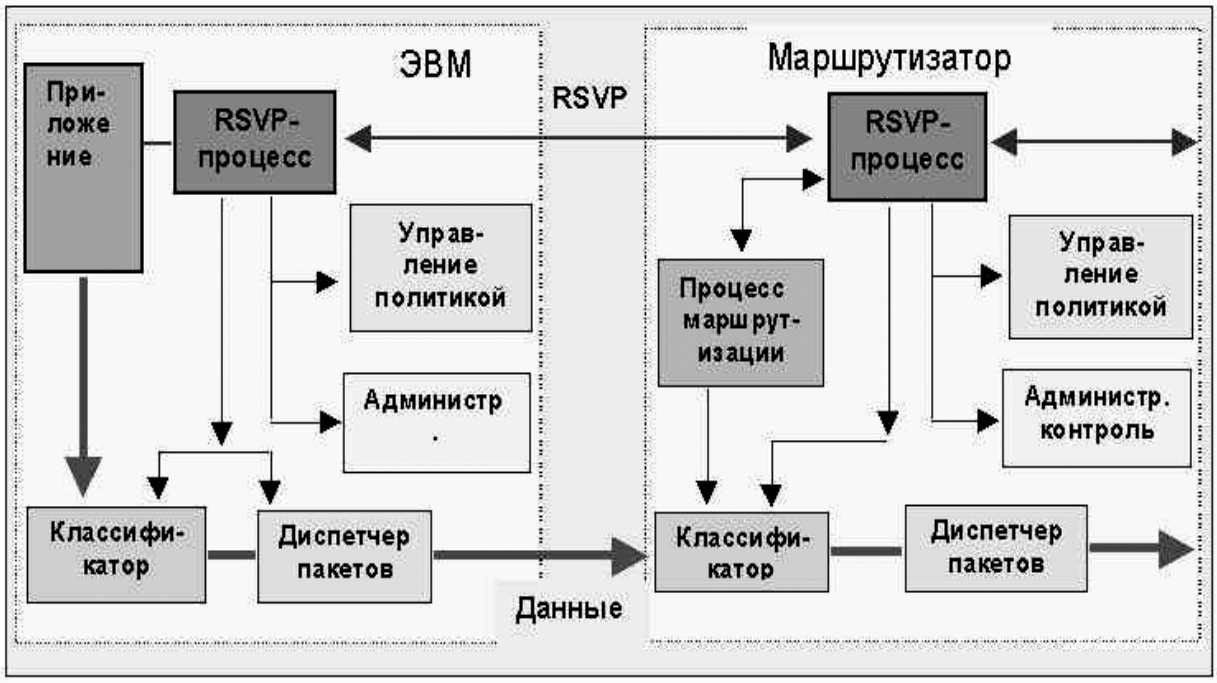


Рис.4.1. приложение RSVP в узле сети

4.2. Потоки данных

RSVP определяет сессию как поток данных с определенным местом назначения и заданным транспортным протоколом. Каждая сессия является совершенно независимой.

Сессия RSVP описывается тремя параметрами: DestAddress, ProtocolId [, DstPort]. DestAddress – IP-адрес места назначения информационных пакетов (уникаст или мультикаст). ProtocolId – идентификатор IP протокола. необязательный (опционный) параметр DstPort – обобщенный порт места

назначения, т.е. еще одна точка демультиплексирования на транспортном или прикладном уровне. DstPort может быть определено полем порта места назначения UDP/TCP.

Заметим, что, строго говоря, не обязательно включать в описание сессии DstPort, когда DestAddress является мультикастным, так как различные сессии могут всегда иметь различные мультикаст-адреса. Однако, DstPort необходим для того, чтобы разрешить более одной уникаст-сессии для одной и той же PC-получателя.

Для уникастной передачи может быть один получатель, но много отправителей; RSVP может выполнить резервирование для передачи много_точек -> одна_точка.

4.3. Модель резервирования

Как уже отмечалось в 3.2. простой запрос резервирования RSVP состоит из *flowspec* (спецификация потока) и *filterspec* (спецификация фильтра). Спецификация flowspec определяет желательное значение QoS. Спецификация фильтра в сочетании со спецификацией сессии определяют тип набора пакетов. Спецификация flowspec используется для задания параметров диспетчеров в узлах, через которые транспортируется поток, а спецификация фильтра – для определения параметров классификатора пакетов. Информационные пакеты, адресованные конкретной сессии, но не удовлетворяющие какой-либо спецификации фильтра обрабатываются без гарантий обеспечения оговоренного QOS.

Спецификация flowspec в запросе резервирования включает в себя значение класса услуг и два набора параметров:

- "Rspec", который определяет желательное значение QoS, и
- "Tspec", который описывает информационный поток.

Например, спецификация фильтра может использоваться для выделения некоторых составных частей информационного потока, осуществляя отбор с учетом полей пакетов прикладного уровня. В описываемом стандарте RSVP спецификация фильтра имеет довольно ограниченную форму: IP-адрес отправителя и номер порта SrcPort (UDP/TCP).

Так как номера портов UDP/TCP используются для классификации пакетов, каждый маршрутизатор должен уметь анализировать эти поля.

Сообщения RSVP, несущие запросы резервирования, исходят со стороны получателя и направляются отправителю информации. В каждом промежуточном узле запрос резервирования запускает две процедуры:

4.3.1.Резервирование канала

Процесс RSVP проходит стадии контроля допуска и политики. Если какой-либо тест не прошел, резервирование отвергается и посылается сообщение об ошибке. Если все тесты прошли успешно, узел устанавливает классификатор пакетов, для того чтобы отбирать пакеты, указанные в спецификации фильтра. Далее устанавливается контакт с соответствующим канальным уровнем для получения желательного QoS, заданного в flowspec.

Если технология канального уровня поддерживает свои средства управления QoS, тогда RSVP должен согласовать с канальным уровнем получение требуемого QoS.

4.3.2. Переадресация запроса в обратном направлении

Запрос резервирования посылается от получателя отправителю (или отправителям) данных. Запрос резервирования, который переадресуется узлом дальше может отличаться от того, который он получил. Дело в том, что механизм управления трафиком модифицирует flowspec от узла к узлу и, что более важно, запросы резервирования, поступающие от получателей мультикастинг-дерева должны объединяться по мере продвижения процесса резервирования в направлении отправителя данных.

Когда получатель данных отправляет запрос резервирования, он может запросить также присылку сообщения, подтверждающего резервирование. Процесс резервирования распространяется от получателей к отправителям, от узла к узлу. В каждом узле требования резервирования объединяются и сопоставляются с имеющимися возможностями. Это продолжается до тех пор, пока запрос не достигнет отправителя или пока не возникнет конфликт перегрузки. В результате получатель данных, направивший запрос резервирования, получит сообщение об успехе или ошибке.

Базовая модель резервирования RSVP является однопроходной: получатель посылает запрос резервирования вдоль мультикастинг-дерева отправителю данных и каждый узел по пути воспринимает или отвергает этот запрос. Результаты доставляются протоколом RSVP в PC получателя. Эти данные могут позднее служить для динамической адаптации соответствующих запросов резервирования.

4.4. Стили резервирования

Запрос резервирования включает в себя набор опций, которые в совокупности называются *стилем*. Одна опция резервирования определяет способ резервирования различными отправителями в пределах одной сессии.

Другая опция резервирования контролирует выбор отправителей. В одних случаях каждому отправителю ставится в соответствие определенная спецификация фильтра, в других – таких спецификаций не требуется вовсе. В настоящее время определены следующие стили:

4.4.1. Стиль WF (Wildcard-Filter)

Стиль WF использует опции: "разделенного" резервирования и произвольного выбора отправителя ("wildcard"). Т. е. резервация со стилем WF создает резервирование, которое делится между потоками всех отправителей. Это резервирование может рассматриваться как общая “труба", чей размер равен наибольшему из ресурсных запросов от получателей, и не зависит от числа отправителей. Стиль резервирования WF передается в направлении отправителей и автоматически распространяется на новых отправителей при их появлении. Символически можно представить запрос резервирования стиля WF как:

$$WF(* \{Q\}),$$

где звездочка представляет произвольную подстановку при выборе отправителя, а Q – спецификация flowspec

4.4.2. Стиль FF (Fixed-Filter)

Стиль FF использует опции: "четкое" (distinct) резервирование и "явный" (explicit) выбор отправителя. Таким образом, простой запрос со стилем FF создает точно заданное резервирование для информационных пакетов от

определенного отправителя, без совместного использования ресурса с другими отправителями в пределах одной и той же сессии. Символически простой запрос резервирования FF можно представить как:

FF(S{Q}),

где S – выбранный отправитель, а Q – соответствующая спецификация flowspec; эта пара параметров образуют дескриптор потока. RSVP позволяет применение нескольких простых стилей резервирования FF одновременно, при этом формируется список дескрипторов потоков:

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

Полное резервирование в канале для данной сессии характеризуется суммой Q1, Q2, ... для всех отправителей, куда посланы запросы.

4.4.3. Стиль SE (Shared Explicit)

Стиль SE использует опции: "разделенное" (shared) резервирование и "явный" (explicit) выбор отправителя. Таким образом, стиль резервирования SE формирует одно резервирование, которое совместно используется несколькими отправителями. В отличие от стиля WF, SE позволяет получателю непосредственно специфицировать набор отправителей. Запрос резервирования SE, содержащий flowspec Q и список отправителей S1, S2, ... можно представить в символьной форме как:

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

Разделенное резервирование, выполненное с применением стилей WF и SE, пригодно для мультикастных приложений, где несколько источников данных редко осуществляют передачу одновременно. Пакетная передача голоса может служить примером разделенного резервирования, так как лишь ограниченное число людей говорят одновременно. Каждый получатель может направить запрос резервирования WF или SE на удвоенную полосу пропускания, необходимую одному отправителю, позволяя тем самым говорить обоим партнерам одновременно. С другой стороны стиль FF, который осуществляет четкое резервирование для потоков отдельных отправителей, подходит для передачи видеосигналов.

Правила RSVP не позволяют объединять разделенные резервирования с четкими резервированиями, так как эти модели абсолютно несовместимы. Не допускается также объединение явного и произвольного (wildcard) выбора

отправителей, так как это может вызвать предоставление незаказанных услуг получателю, который указал тип услуг явно. Таким образом, стили WF, SE и FF не совместимы.

Можно моделировать эффект WF резервирования, используя стиль SE. Когда приложение запрашивает WF, процесс RSVP получателя может использовать местный статус для выполнения эквивалентного резервирования SE, которое в явном виде перечисляет всех отправителей. Однако резервирование SE вынуждает классификатор пакетов в каждом узле в явном виде выбрать каждого отправителя из списка, в то время как WF позволяет классификатору пакетов осуществить произвольный выбор отправителя и порта с помощью "wild card". Когда список отправителей велик, стиль резервирования WF обеспечивает значительно меньшие издержки, чем SE.

4.5. Примеры стилей

На рис.4.2. показан пример маршрутизатора с двумя входными интерфейсами $I_А$ и $I_Б$, через которые проходят входные потоки, и двумя выходными интерфейсами $I_В$ и $I_Г$, через которые осуществляется переадресация входных потоков. Пусть существует три отправителя S1 (S2 и S3), подключенные к интерфейсам $I_А$ и $I_Б$, соответственно. Имеется три получателя R1 (R2 и R3), которые маршрутизированы через выходные интерфейсы $I_В$ и $I_Г$, соответственно. Будем также предполагать, что интерфейс $I_Г$ подключен к широковещательной сети, а R2 и R3 достижимы через разные маршрутизаторы, не показанные на рисунке.

Здесь нужно специфицировать мультикастные маршруты в пределах узла, отображенного на рис.4.2. Предположим сначала, что информационные пакеты от каждого из отправителей S_i, показанных на рисунке, маршрутизованы на оба выходных интерфейса. При этих предположениях на рисунках4.3, 4.4 и 4.5 проиллюстрированы стили резервирования WF, FF и SE, соответственно.

Рис.4.2. Конфигурация маршрутизатора

Для простоты эти примеры показывают flowspec как одномерное кратное повторение некоторого базового качества ресурса B. Колонка "Резервирует" показывает запросы резервирования RSVP, полученные через выходные интерфейсы $I_В$ и $I_Г$, а колонка "Получает" показывает результирующее состояние резервирования для каждого интерфейса. Колонка "Посылает" показывает запросы резервирования, посланные предшествующим узлам (I_A и $I_Б$). В колонке "Резервирует" каждая рамка представляет один зарезервированный виртуальный канал с соответствующим дескриптором потока.

Рис.4.3, демонстрируя стиль WF, показывает две ситуации, в которых требуется объединение.

Каждый из двух узлов, следующих за интерфейсом $I_Г$ посылают независимые запросы резервирования RSVP, эти два запроса должны быть объединены в одну спецификацию flowspec (3B), которая используется для выполнения резервирования в интерфейсе $I_Г$.

Резервирования для интерфейсов $I_В$ и $I_Г$ должны быть объединены, для того чтобы осуществить переадресацию запросов резервирования далее и получить спецификацию flowspec (4B).

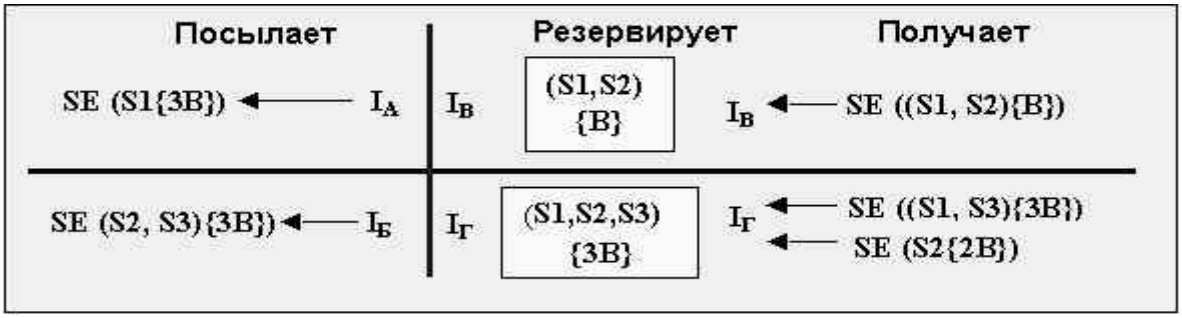


Рис.4.3. Пример резервирования WF (Wildcard-Filter)

На рис.4.4 проиллюстрирован стиль резервирования FF (Fixed-Filter). Для каждого выходного интерфейса, имеется отдельное резервирование для каждого запрошенного источника, но это резервирование будет общим для всех получателей, которые послали запрос. Дескрипторы для получателей S2 и S3, полученные через выходные интерфейсы I_B и $I_Г$, вкладываются в пакеты запросов, направляемых предыдущему узлу ($I_Б$). С другой стороны, три различных дескриптора потоков, специфицирующих отправителя S1, объединяются в один запрос FF(S1{4B}), который посылается предыдущему узлу (I_A).

На рис.4.5 показан пример стиля резервирования SE. Когда резервирования стиля SE объединяются, результирующая спецификация фильтра является объединением исходных спецификаций, а результирующая спецификация flowspec равна наибольшей из flowspec.

Рис.4.4. Пример резервирования FF (Fixed-Filter)

Рис.4.5. Пример резервирования SE (Shared-Explicit)

Приведенные примеры предполагают, что информационные пакеты от S1, S2 и S3 маршрутизируются через оба выходных интерфейса. Нижняя часть рис.2 показывает еще одно предположение о маршрутизации: информационные пакеты от S2 и S3 не переадресуются интерфейсу I_B, напр., из-за того что сеть обеспечивает более короткий путь для пакетов отправителя к R1. Рис.3

показывает пример резервирования WF именно при этом предположении (стрелками отмечены допустимые маршруты). Так как нет пути от $I_Б$ к $I_В$, резервирование, переадресуемое интерфейсом $I_Б$, рассматривает резервирование только для интерфейса $I_Г$.

4.6. Сообщения RSVP

О сообщения RSVP уже говорилось в параграфе 3.2 предыдущей главы. Теперь рассмотрим эти сообщения на модели RSVP узла маршрутизатора на рис.4.6.

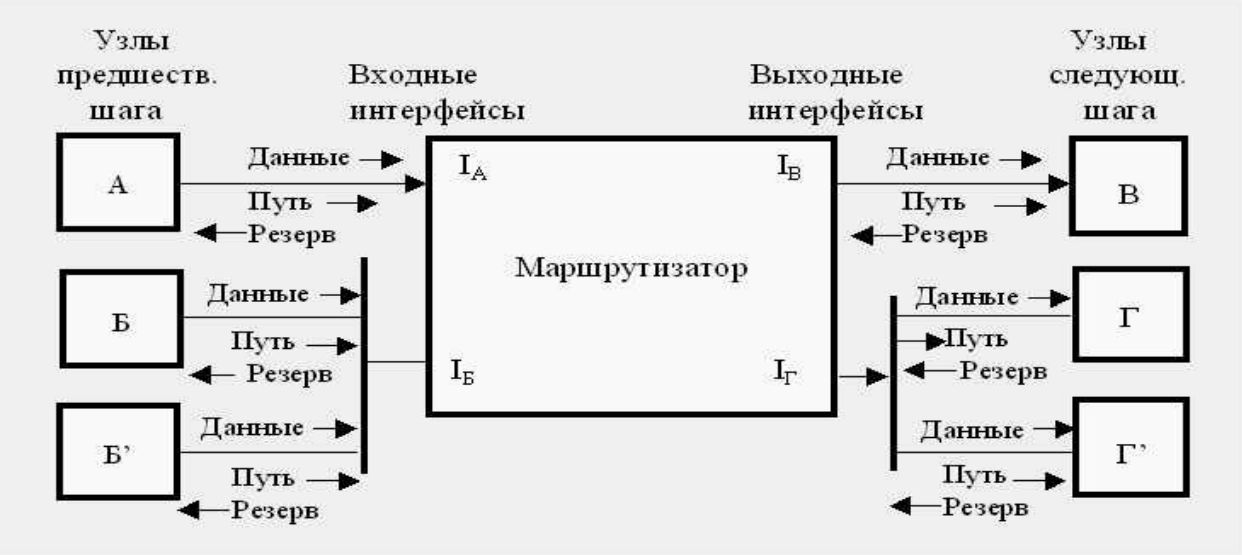


Рис.4.6. Маршрутизатор, использующий RSVP

На рис.4.6 модели RSVP узла маршрутизатора каждый поток данных приходит со стороны предшествующего узла через соответствующий входной интерфейс и выходит из маршрутизатора через один или несколько выходных интерфейсов. Один и тот же интерфейс для разных потоков в пределах одной сессии может выполнять как входную, так и выходную роль. Несколько предшествующих узлов и/или последующих узлов могут для коммуникаций использовать один и тот же физический интерфейс; например, на рисунке два узла Г и Г' подключены к широковещательной сети через интерфейс $I_Г$.

как уже отмечалось, существуют два фундаментальных типа сообщений RSVP: Resv и Path. Каждый получатель посылает свой RSVP-запрос резервирования в виде сообщений (Resv) отправителям данных. Эти сообщения должны двигаться в точности тем же маршрутом с учетом выбора отправителей, что и данные только в противоположном направлении. Они создают и

поддерживают состояние резервирования в каждом узле вдоль маршрута. Сообщения Resv должны быть, в конце концов, доставлены PC-отправителям, таким образом, PC устанавливают параметры управления трафиком.

Каждая PC отправитель передает RSVP сообщения "Path" вдоль уникаст/мультикаст маршрутов, сформированных с помощью маршрутных протоколов. Эти сообщения Path запоминают состояние пути в каждом узле вдоль маршрута. Это состояние пути включает в себя уникастный IP-адрес предыдущего узла, который используется для маршрутизации сообщений Resv от узла к узлу в противоположном направлении. Сообщение Path содержит помимо этого следующую информацию:

- ***Шаблон отправителя***

Сообщение Path должно нести в себе шаблон отправителя (Sender Template), который описывает формат пакетов данных, посылаемых отправителем. Этот шаблон имеет форму спецификации фильтра, которая может использоваться для отделения пакетов данного отправителя от других пакетов в пределах сессии.

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 опционно UDP/TCP порт, с учетом идентификатора протокола, заданного для сессии.

- ***Спецификация Tspec отправителя***

Сообщения Path должны содержать спецификацию отправителя Tspec, которая определяет характеристики информационного трафика, формируемого отправителем. Спецификация Tspec используется для предотвращения избыточного резервирования.

Сообщения Path посылаются с теми же адресами отправителя и получателя, что и данные, так что они будут корректно маршрутизироваться даже через сетевые области, не поддерживающие RSVP. С другой стороны, сообщения Resv посылаются от узла к узлу; каждый узел, поддерживающий RSVP, переправляет сообщение Resv по уникальному адресу предшествующего узла RSVP.

4.7. Объединение Flowspecs

Сообщение Resv, переадресованное предшествующему узлу, несет в себе спецификацию flowspec, которая является “наибольшей” из всех flowspec, запрошенных последующими узлами, которым будут посылаться данные.

Так как flowspecs непрозрачны для RSVP, действительные правила для сравнения flowspecs должны быть определены и реализованы вне рамок этого протокола. Реализация RSVP потребует обращения к специальной программе для выполнения объединения спецификаций flowspec.

Заметим, что спецификации flowspecs представляют собой в общем случае многомерные векторы; они могут содержать как Tspec, так и Rspec компоненты, каждая из которых может сама быть многомерной. Например, если один запрос требует высокой пропускной способности, а другой – жесткого ограничения задержек, один не может быть "больше" другого. В таком случае, вместо взятия большего, прикладная программа объединения должна быть способна сформировать такую спецификацию flowspec, которая по крайне мере столь же велика, как и каждая из составляющих; математически это наименьший верхний предел LUB (least upper bound). В некоторых случаях спецификация flowspec по крайне мере настолько мала насколько возможно; это наибольший верхний предел GLB (Greatest Lower Bound).

Для вычисления эффективного значения flowspec (Re, Te), инсталлируемого в интерфейс, используются следующие шаги. Здесь Te – эффективная спецификация Tspec, а Re – эффективная спецификация Rspec.

Определяется эффективная спецификация flowspec для выходного интерфейса. В зависимости от технологии канального уровня, это может требовать объединения спецификаций flowspecs различных последующих узлов. Это означает вычисление эффективной спецификации flowspec, как LUB flowspecs. Какие спецификации следует объединять, определяется средой канального уровня, в то время как процедура объединения определяется используемой моделью обслуживания . В результате получается спецификация flowspec, которая непрозрачна для RSVP, но в действительности состоит из пары (Re, Resv_Te), где Re является эффективной спецификацией Rspec, а Resv_Te - эффективная спецификация Tspec.

Производится вычисление спецификации Path_Te, зависящей от приложения и представляющей собой сумму всех Tspecs, которые были

присланы в сообщениях Path, пришедших от различных предшествующих узлов (например, некоторые или все узлы А, Б, и Б' на рис.4.6).

(Re, Resv_Te) и Path_Te передаются системе управления трафиком. Управление трафиком вычислит эффективную спецификацию flowspec, как минимум Path_Te и Resv_Te.

4.8. Программируемое состояние

RSVP использует подход "soft state" (программируемое состояние) для управления резервированием в маршрутизаторах и PC. Прогаммируемое состояние RSVP создается и периодически обновляется посредством сообщений Path и Resv. Состояние ликвидиркуется, если не приходит подтверждения в течение заданного времени, определяемого так называемым тайм-аутом очистки. Состояние может быть ликвидировано также посредством сообщения "teardown" (уничтожение). По истечении каждого таймаута обновления и после любых изменений состояния RSVP осуществляет проверку, для того чтобы подготовить и отправить сообщения обновления Path и Resv последующим узлам.

Сообщения Path и Resv практически эквивалентны . Когда маршрут меняется, следующее сообщение Path инициализирует состояние прохода для нового маршрута, а последующие сообщения Resv установят для него резервирование. Состояние же на неиспользованном в данный момент сегменте маршрута будет аннулировано по таймауту. Следовательно, решение о том, является ли сообщение "новым" или "обновляющим" принимается отдельно для каждого узла в зависимости от его текущего состояния.

Протокол RSVP посылает свои сообщения в виде IP-дейтограмм без какого-либо дополнительных мер по обеспечению надежности доставки. Периодическая передача сообщений обновления от PC и маршрутизаторов позволяет компенсировать случайные потери отдельных RSVP-сообщений. Если таймаут удаления установлен равным K периодам обновления, то такая реализация RSVP может допускать потерю K-1 RSVP-пакетов подряд без аннулирования состояния. Механизм управления сетевым трафиком должен быть конфигурирован так, чтобы предоставить минимальную полосу пропускания для сообщений RSVP, чтобы предотвратить их потерю из-за перегрузки канала.

Состояние, поддерживаемое RSVP, является динамическим. Для изменения набора отправителей Si или изменения любого запроса QoS, PC просто начинает посылать измененные сообщения Path и/или Resv. В результате будет осуществлено соответствующее изменение RSVP-состояния во всех узлах вдоль пути, неиспользуемые состояния будут аннулированы по тайм-ауту, если не поступит прямых указаний по их ликвидации до этого.

В стабильном состоянии осуществляется обновление статуса узел за узлом. Когда полученное состояние отличается от хранящегося, последнее обновляется. О модификации состояния соседи оповещаются с помощью сообщений обновления, которые рассылаются сразу после изменения состояния. Но эта волна изменений может остановиться в узле, где в результате слияния получается состояние, которое не отличается от прежнего. Это минимизирует трафик управления RSVP, что весьма существенно для больших мультикастинг-групп.

Состояние, которое получено через конкретный интерфейс I* никогда не должно переадресовываться этому интерфейсу. Состояние, которое направляется интерфейсу I* должно вычисляться на основе состояний, полученных от других интерфейсов, исключая I*. Тривиальный пример, поясняющий это правило приведен на рис.4.7, на котором показан маршрутизатор с одним отправителем и одним получателем на каждом интерфейсе (R1, S1 и R2, S2). Интерфейсы $I_А$ и $I_Б$ выполняют как входные, так и выходные функции. Оба получателя используют WF-стиль резервирования, в котором сообщения Resv переадресуются всем предшествующим узлам группы вдоль маршрута, за исключением узла, от которого это сообщение получено. В результате достигается независимое резервирование для обоих направлений (на рис.4.7 “Получает” и “отправляет” подразумевает внешнее направление по отношению к маршрутизатору).

Интерфейс IA		Интерфейс IБ	
Получает	Отправляет	Получает	Отправляет
WF (* {4B})	WF (* {3B})	WF (* {3B})	WF(*{4B})
Резервирует			
* {4B}		* {3B}	

Рис.4.7. Независимые резервирования

Существует еще одно правило, которое управляет процессом переадресации сообщений Resv: состояние из сообщения Resv, полученное через выходной интерфейс Io, следует передавать входному интерфейсу Ii только в том случае, когда сообщение Path от Ii переадресовано к Io.

4.9. Аннулирование

Сообщения RSVP "аннулирование" удаляет проход или состояние резервирования. Хотя прямое уничтожение старого резервирования не является обязательным, оно настоятельно рекомендуется, так как ускоряет переходные процессы в сети.

Существует два типа RSVP сообщений аннулирования: PathTear и ResvTear. Сообщение PathTear направляется всем получателям и ликвидирует состояние прохода, а также все зависящие от него состояния резервирования. Сообщение ResvTear уничтожает состояние резервирования и направляется всем отправителям.

Запрос *аннулирование* (teardown) может посылаться приложением оконечной системы (получатель или отправитель), или маршрутизатором в результате таймаута или при появлении привилегированной задачи. После инициализации запрос-аннулирование должен переадресовываться от узла к узлу без задержки. Сообщение аннулирование уничтожает специфицированное состояние в узле-получателе.

Подобно другим сообщениям RSVP, запросы-аннулирования доставляются без гарантии надежности. Потеря такого запроса не вызовет катастрофы, так как не используемое состояние будет рано или поздно ликвидировано по таймауту.

Если маршрутизатор не получил сообщения аннулирования, он ликвидирует соответствующее состояние по таймауту и формирует сообщение аннулирование, рассылаемое последующим узлам. Предполагая, что вероятность потери сообщения RSVP мала, наибольшее среднее время ликвидации ненужного состояния не превышает периода обновления.

Необходимо иметь возможность ликвидировать любой субнабор установленных состояний. Для состояний прохода минимально это может быть один отправитель. Для состояний резервирования таким объектом является спецификация фильтра. Например, в случае, показанном на рис.4.7, получатель R1 может послать сообщение ResvTear только отправителю S2 (или любому субнабору из списка спецификаций фильтрации), оставляя S1 без изменений.

Сообщение ResvTear специфицирует стиль и фильтры, любая спецификация flowspec игнорируется. Любая рабочая спецификация flowspec будет убрана, если все ее спецификации фильтров будут ликвидированы.

4.10. Ошибки

Существует два типа RSVP-сообщений об ошибках: ResvErr и PathErr. Сообщения PathErr очень просты, они посылаются отправителю виновнику ошибки и не изменяют состояния прохода в узлах, через которые проходят. Существует всего несколько причин ошибок прохода.

Однако для синтаксически верных запросов резервирования имеется много способов быть отвергнутыми. Узел может решить аннулировать установленное резервирование из-за более приоритетных заданий. Так как неудовлетворение запроса может быть вызвано объединением нескольких запросов, ошибка резервирования должна быть ретранслирована всем получателям группы. Кроме того, объединение разнородных запросов создает потенциальную трудность, известную как проблема "резервирования-килера", в которой один запрос может блокировать услуги другого. В действительности существует две такие проблемы:

Первая проблема резервирования-килера (KR-I) возникает, когда уже имеется резервирование Q0. Если другой получатель делает новое Q1 > Q0, результирующее объединенное резервирование Q0 и Q1 может быть отвергнуто системой контроля доступа в некотором последующем узле. Это не должно вредить услугам на уровне Q0. Решение этой проблемы весьма просто: когда

контроль доступа не пропускает запрос резервирования, существующее состояние резервирования сохраняется.

Вторая проблема (KR-II) противоположна первой: получатель, выполняющий резервирование Q1, сохраняется даже в случае не прохождения контроля доступа для Q1 в каком-то узле. Это не должно мешать другому получателю, установить меньшее резервирование Q0, которое бы прошло, если бы не было объединено с Q1.

Чтобы решить эту проблему сообщения ResvErr устанавливают дополнительное состояние, называемое, "состояние блокады", в каждом из узлов, через которые проходит это сообщение. Состояние блокады в узле модифицирует процедуру объединения, так чтобы игнорировать блокирующие спецификации flowspec (Q1 в вышеприведенном примере), позволяя скромным запросам проходить и осуществлять свое резервирование. Состояние резервирования Q1 считается в данном случае заблокированным.

Запрос резервирования, не прошедший контроль допуска создает состояние блокады в соответствующем узле, но остается действующим во всех предшествующих узлах. Было предложено, чтобы эти резервирования до точки отказа были удалены. Однако, эти резервирования были сохранены по следующим причинам:

- Имеется две возможные причины получателю настаивать на резервировании: а) заказываемый ресурс доступен по всей длине пути, или б) нужно получить желаемый уровень QoS вдоль оговоренного пути так далеко, как это возможно. Конечно, во втором случае, а может быть и в первом, получатель захочет настаивать на резервировании, осуществленном вплоть до точки блокировки.
- Если бы эти резервирования в предыдущих узлах не будут сохранены, реагирование RSVP на некоторые переходные отказы станет хуже. Например, предположим, что маршрут переключился на альтернативный, который сильно перегружен, так что существующие резервирования не могут быть удовлетворены, и система возвращается к исходному маршруту. Состояние блокады в каждом из маршрутизаторов до узкого места не должно быть немедленно удалено, так как это не позволит системе быстро восстановиться.

- Если бы мы не обновляли резервирование в предшествующих узлах каждые Tb секунд, они могли бы быть удалены по таймауту (Tb время таймаута состояния блокады).

4.11. Подтверждение

Чтобы запросить подтверждение на свое резервирование, получатель Rj включает в сообщение Resv объект запроса подтверждения, содержащий IP-адрес Rj. В каждой точке объединения только наибольшая из спецификаций flowspec и соответствующий объект запроса подтверждения посылаются далее. Если запрос резервирования от Rj равен или меньше уже существующего резервирования, его Resv не переадресуется последующим узлам, и, если Resv включает в себя запрос подтверждения, отправителю Rj посылается сообщение ResvConf. Если запрос подтверждения переадресуется, это делается немедленно и не более одного раза на каждый запрос.

Этот механизм подтверждения имеет следующую последовательность:

- Новый запрос резервирования со спецификацией flowspec больше чем любая из действующих в данной точке спецификаций сессии обычно вызывает либо сообщение ResvErr, либо ResvConf, отправляемое получателю каждым из отправителей данных. В этом случае, сообщение ResvConf будет подтверждением, относящимся ко всему пути.
- Получение ResvConf не предоставляет никаких гарантий. Предположим, что два запроса резервирования от получателей R1 и R2 пришли в узел, где они были объединены. R2, чье резервирование было вторым по времени, может получить подтверждение ResvConf от данного узла, в то время как запрос R1 еще не прошел весь путь и он может еще быть отвергнут каким-то последующим узлом. Таким образом, R2 может получить ResvConf, когда не имеется полномасштабного резервирования вдоль всего пути; более того, R2 может получить ResvConf, за которым последует сообщение ResvErr.

4.12. Администрирование (управление политикой)

Механизм управления политикой определяет, каким пользователям или приложениям позволено осуществлять резервирование и в каком объеме. RSVP-

запросы QoS позволяют определенным пользователям получить предпочтительный доступ к сетевым ресурсам. Для предотвращения злоупотреблений, необходима некоторая обратная связь. Такого рода обратная связь может быть реализована с помощью административной политики обеспечения доступа, или путем введения прямой или виртуальной оплаты резервирования. В любом случае требуется идентификация пользователя.

Когда запрашивается новое резервирование, каждый узел должен ответить на два вопроса: "Имеется ли достаточно ресурсов, чтобы удовлетворить запрос?" и "Позволено ли данному пользователю осуществлять резервирование?" Эти два решения называются "управлением доступа" и "управлением политикой", соответственно. Различные административные домены в Интернет могут иметь разные политики резервирования.

На вход управления политикой поступают специфические блоки данных, которые заключены в объектах POLICY_DATA протокола RSVP. Эти блоки данных могут включать в себя параметры доступа пользователя, его класс, номер счета, границы квоты и пр. Подобно flowspecs, эти данные недоступны для RSVP, который просто передает их, когда требуется, системе управления политикой. Аналогично, объединение этих данных должно выполняться системой управления политикой, а не самим протоколом RSVP. Заметим, что точки объединения данных, характеризующих политику, должны находиться на границах административных доменов.

Перенос таких данных, поставляемых пользователями, в сообщениях Resv может представлять проблему в случае существенного увеличения числа пользователей. Когда мультикастинг-группа содержит большое число получателей, может оказаться невозможно или нежелательно транспортировать данные, описывающие политику, вдоль всего маршрута. Эти данные должны объединяться как можно ближе к получателям, чтобы избежать чрезмерного информационного потока.

Глава 5. МОДЕЛЬ УПРАВЛЕНИЯ ТРАФИКОМ В RSVP.

5.1. Основные принципы

Перерисуем структурную схему резервирования в RSVP на рис.4.1 с учетом введенных понятий управления трафиком. Результат этой операции приведен на рис.5.1. Рассмотрим эту модель более пристально.

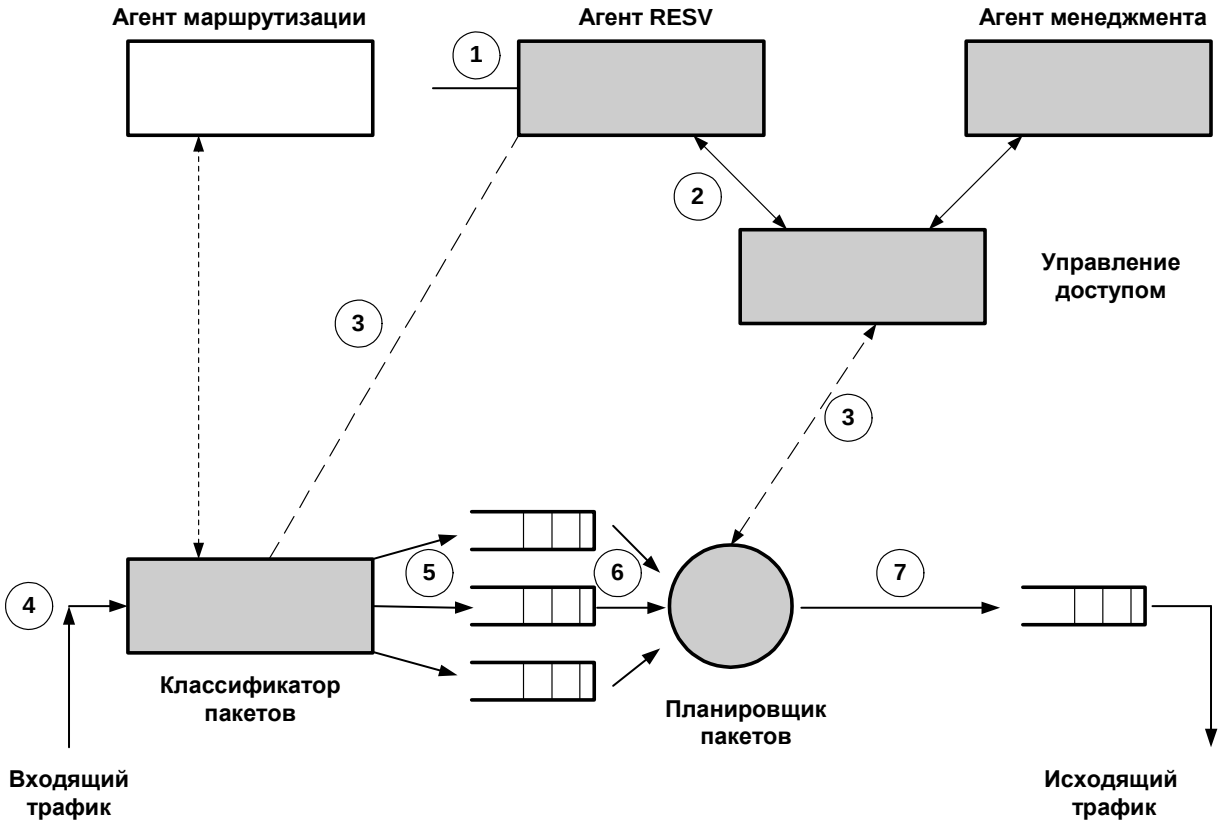


Рис.5.1.Модель управления трафиком в RSVP

Начнем рассмотрениее с отмеченного в главе 1 дипломной работы факта, что в стандартном Интернет все пакеты обслуживаются одинаково на основе принципа best effort. Однако, когда какое-либо имеющее соответствующее право приложение запрашивает QoS для конкретного потока, то пакеты из этого потока должны каким-то образом распознаваться и обслуживаться специальным образом для обеспечения специального качества обслуживания на всем пути следования. Вначале рассмотрим, что требуется для обеспечения такого QoS:

- Необходим стандартный способ определения желаемых показателей QoS потока.
- Необходимы метод и протокол информирования сети о требованиях к QoS потока (такую возможность обеспечивает рассмотренный в предыдущих главах протокол резервирования ресурсов RSVP)
- На каждом сетевом узле необходимо реализовать процедуру *управления (контроля) допуска (admission control)* для того, чтобы проверить, имеются ли необходимые ресурсы для обеспечения требований QoS, и если имеются - зарезервировать ресурсы (буферы, пропускную способность канала) для их использования соответствующим потоком, допущенным в сеть.
- После того, как запрос на QoS потока принят к обслуживанию и пакеты этого потока начинают передаваться через сеть, эти пакеты должны идентифицироваться и различаться от пакетов, принадлежащих другим потокам на каждом сетевом узле по пути прохождения пакетов. Для этой цели в хостах и маршрутизаторах необходим механизм *классификации пакетов (packet classification).*
- Для того, чтобы поток получал запрошенное QoS, пакетам в потоке должен предоставляться своевременный доступ к зарезервированным ресурсам на каждом узле. Для этой цели требуется *планировщик пакетов (packet scheduler).* Когда приложение запрашивает у сети гарантию QoS, оно по существу заключает двустороннее соглашение с сетью. По условиям данного соглашения, сеть обязана обеспечить предоставление запрошенных ресурсов допущенному в сеть потоку, а приложение обязано функционировать таким образом, чтобы его поток не превышал оговоренных соглашением показателей трафика. Сетевые узлы должны предпринимать шаги по защите самих себя и «хорошо ведущих себя» потоков от «плохих» потоков, которые нарушают требования по трафику. Как правило, планировщик пакетов обеспечивает такую функцию. В эту функцию входит механизм межсетевого экрана, который обеспечивает отделение или защиту пакетов в каждом потоке от трафика, передаваемого в других потоках. Кроме того, часто в комбинации с планировщиком пакетов используется механизм *регулирования трафика* (*traffic policing),* который следит за тем, чтобы поток не нарушал или не превышал свои заданные показатели трафика.

Модуль контроля допуска, классификатор пакетов и планировщик пакетов вместе составляют совокупность компонентов управления трафиком в устройстве (маршрутизаторе или PC), реализующем RSVP. Рис.5.1 иллюстрирует в упрощённом виде RSVP и компоненты управления трафиком внутри маршрутизатора и/или PC, представленных на рис.4.1 предыдущей главы. Для резервирования QoS на базе протокола RSVP, эти компоненты взаимодействуют следующим образом:

1. Агенту RESV в маршрутизаторе поступает RSVP-сообщение RESV. После обработки сообщения RESV с целью проверки его корректности и выделения его содержания, агент RESV вызывает модуль управления допуском. Агент направляет модулю управления допуском спецификацию потока (flowspec), содержащую параметры интегрированных услуг (IntServ) и запрашиваемый тип обслуживания.
2. Модуль управления допуском использует свой алгоритм контроля допуска и информацию по доступным ресурсам для того, чтобы определить, принять к обслуживанию запрос RESV или нет. Если запрос принят, модуль управления допуском информирует об этом решении агента RESV.
3. После успешного допуска, агент RESV направляет классификатору пакетов информацию фильтрации пакетов, которая описывает, каким образом идентифицировать пакеты в допущенном к обслуживанию потоке. Как говорилось выше, эта информация представляет собой кортеж из пяти элементов, состоящий из IP-адреса отправителя и адресата, идентификатора транспортного протокола и номеров портов отправителя и адресата. Классификатор пакетов ведёт внутреннюю базу данных, содержащую такую информацию фильтрации для нескольких потоков. Аналогичным образом, модуль управления допуском также связывается с планировщиком пакетов и направляет ему информацию о параметрах flowspec (например, параметры token bucket), с тем, чтобы планировщик мог должным образом спланировать пакеты в потоке.
4. Когда трафик поступает в маршрутизатор, входной драйвер принимает каждый пакет и направляет пакет классификатору.
5. Классификатор пакетов осуществляет поиск по своей базе данных фильтрации для идентификации потока, которому принадлежит пакет, и

затем ставит пакет в соответствующую очередь. В зависимости от используемых механизмов планирования очередей, планировщик может использовать отдельную очередь для каждого потока, либо потоки могут для целей планирования объединяться в один класс или очередь. В некоторых реализациях, поступающие пакеты могут сперва направляться планировщику пакетов, а планировщик может затем вызвать классификатор.

6. При наличии одной или более входных очередей, планировщик пакетов использует свой механизм планирования для вытаскивания пакета из одной из очередей и планирует время его передачи по исходящему каналу. Этот шаг является критическим, поскольку он в значительной степени определяет QoS, которое поток получает в этом узле. Планировщик принимает во внимание требования по задержке и по пропускной способности, предъявляемые потоками, для приоритезации порядка выбора и передачи пакетов из разных потоков.
7. Когда наступает время передачи пакета согласно расписанию, планировщик направляет пакет на выходной драйвер для его последующей передачи. В зависимости от скорости и типа тракта передачи, могут ставиться в очередь для передачи на выходном канале более одного пакета.

В следующих параграфах этой главы подробно рассматриваются спецификации QoS, используемые для описания требований потока к QoS, а затем компоненты управления трафиком.

5.2. Спецификации QoS

Для спецификации требований QoS, протокол RSVP опирается на принцип интегрированных услуг. Согласно модели IntServ, спецификация QoS состоит из двух частей: Tspec (спецификация трафика), описываемой в Tspec передатчика или FlowSpec приёмника, и Rspec (спецификация приёмника). Tspec описывает показатели трафика потока, создаваемого передатчиком, в то время как Rspec определяет требования по производительности, ожидаемые приёмником, и уровень ресурсов, которые необходимо зарезервировать для потока.

В Интернет трафик обычно является пульсирующим. Трафик реального времени является пульсирующим в связи с тем, что он обычно сжат и состоит из группы пакетов, генерируемых и передаваемых с переменной скоростью. Таким

образом, характеристики трафика потока описываются наилучшим образом путём использования модели token bucket (“ведро с жетонами”). Token bucket моделирует поток трафика в виде серии импульсов (блоков) с интервалами между импульсами, где каждый импульс (блок) состоит из одного или более пакетов, связанных вместе.

5.3. Модель token bucket

Цель модели token bucket - описать поток трафика, когда он входит в сеть и передаётся через сеть. Идея состоит в том, чтобы взять произвольный поток трафика и сформировать его таким образом, чтобы он соответствовал модели трафика, основанной на определённых, измеряемых параметрах с тем, чтобы было легко описать, на какой скорости и в каком количестве он поступает на сетевой элемент. При такой спецификации трафика будет просто говорить о количестве сетевых ресурсов, которое будет потребляет поток, и о том, как можно обслужить этот трафик.

Один из самых лёгких методов формирования потока трафика - это представить его в виде регулярного потока пакетов с одинаковым (равномерным) интервалом времени между следующими один за другим пакетами одинакового размера. Модель leaky bucket (“дырявое ведро”), предложенная Тернером, преобразовывает входной поток трафика в такой выходной поток. Эта модель представлена на рис. 5.2. Каждый поток имеет своё собственное “ведро” размером b байт. При поступлении потока пакетов, каждый пакет размещается в ведре. В любое время в ведре может находиться максимум b байт. Пакет длиной d байт ($d < b$) может быть помещён в ведро при условии, что в ведре имеется d байт свободного места; в противном случае пакет отбрасывается. На дне ведра находится регулятор трафика, который отправляет трафик с постоянной скоростью p байт/с. Таким образом, пакет (длиной d байт) наверху ведра удерживается в течение промежутка времени d/p перед его передачей.

В результате, на выходе leaky bucket мы имеем поток данных, передаваемых с постоянной скоростью. Очевидно, что эта модель особенно хорошо подходит для потоков данных с фиксированной скоростью передачи. Однако, если скорость передачи данных чрезвычайно изменчивая и пульсирующая, в качестве выходной скорости должна выбираться скорость p

байт/с, равная пиковой скорости потока для предотвращения отсева пакетов потока на входе в ведро. Это действительно так, поскольку если выбранная скорость *p* ниже, чем пиковая скорость, то когда порция трафика поступит с пиковой скоростью, она заполнит ведро и приведёт к отсеву пакетов. На практике, многие потоки дейтаграмм являются довольно пульсирующими; отношение пиковой скорости к средней скорости передачи составляет от 5 до 10. Рассмотрим, например, поток сжатых видеоданных для видеоизображений частотой 30 кадров в секунду. Блок пакетов будет поступать каждые 30 миллисекунд, соответствующих каждому кадру. При этом поток может иметь скорость передачи 100 кбайт/с. Тем не менее, случайное изменение сцены может вызвать появление блока пакетов со скоростью 500 кбайт/с в течение короткого интервала времени, соответствующего первым нескольким кадрам в новой сцене. Таким образом, здесь необходима своевременная доставка этих кадров для того, чтобы принимающее приложение обеспечило плавную и непрерывную картинку. Такой поток называется потоком с переменной скоростью передачи в битах (VBR).

Рис. 5.2. Метод leaky bucket (“дырявое ведро”)

Для схемы leaky bucket потребуется формирование потока с VBR на случайной пиковой скорости для предотвращения потери данных. Однако, если сетевые ресурсы зарезервированы под пиковую скорость, это приведёт к расточительному использованию сетевых ресурсов, поскольку поток с VBR только изредка посылает данные на пиковой скорости.

Схема token bucket является разновидностью предложенной Тернером модели leaky bucket. На рис. 5.3 показана работа схемы token bucket. Важное отличие этих двух схем состоит в том, что схема token bucket использует “ведро” для управления регулятором потока, а не для контроля поступления данных потока. В системе token bucket скорость *r* - это скорость, с которой жетоны (tokens) помещаются в ведро. Предположим, что каждый жетон соответствует одному байту данных. Ведро имеет ёмкость *b* байтов, а жетоны помещаются в ведро со скоростью *r* байт/с. Если ведро наполняется, вновь поступившие жетоны отбрасываются.

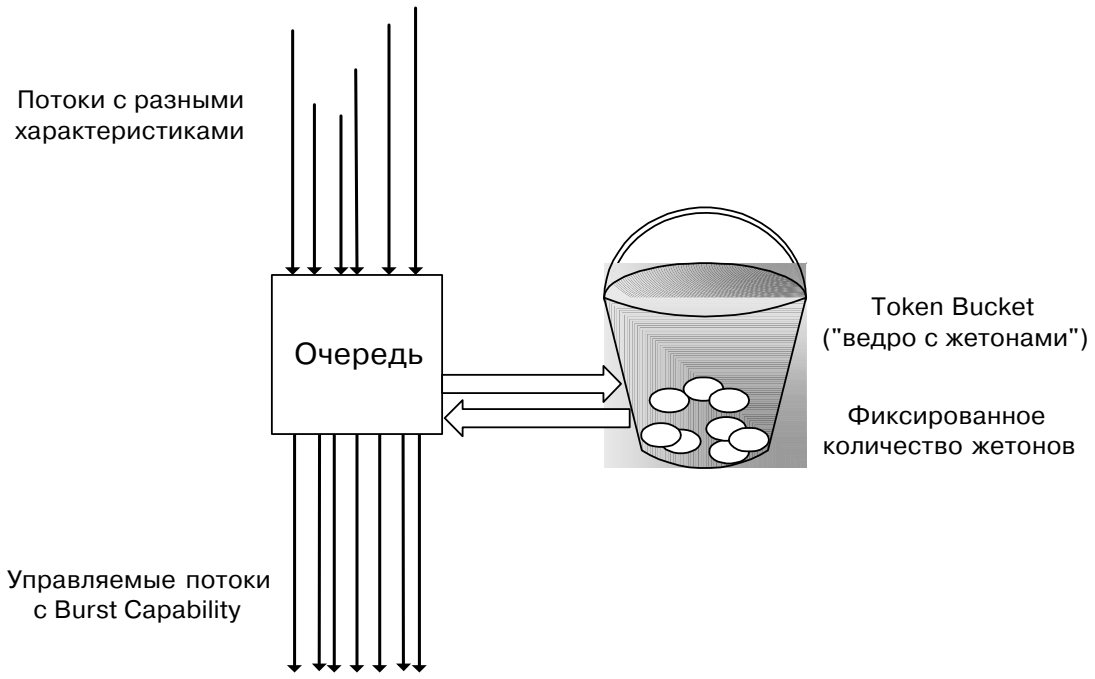


Рис.5.3. Метод token bucket (“ведро с жетонами”)

Когда прибывает пакет, он помещается в буфер с левой стороны. Предположим, что перед буфером, слева от него, находится пакет длиной *d* байт. Для передачи пакета регулятор должен вынуть *d* жетонов из ведра. Если в

ведре не имеется достаточного количества жетонов (*d*), пакет, прежде чем он сможет быть передан, должен ждать, пока в ведре не накопится *d* жетонов.

Если ведро полное, и с левой стороны в буфер поступает блок пакетов, блок пакетов (суммарной длиной *b* байт) выйдет из регулятора немедленно без всяких интервалов между ними.

По сравнению со схемой leaky bucket, которая преобразовала бы такой блок (импульс) в поток пакетов, выходящих с постоянной скоростью, схема token bucket гарантирует две вещи.

Во-первых, она никогда не отправит более чем ($b + r \times t$) байт данных за любой интервал времени *t* для потока с параметрами token bucket *b* и *r*. Во-вторых, долговременная средняя скорость передачи такого потока будет составлять *r* байт/с. Таким образом, схема token bucket является более гибкой по сравнению со схемой leaky bucket. Она позволяет передавать пульсирующий трафик без необходимости запрашивать скорость передачи данных, равную пиковой скорости передачи данных, но ограничивает размер блока (импульса) *b* байтами. Единственная проблема при использовании системы token bucket - каким образом блок пакетов вводится в сеть. Предположим, что поток не передаётся в течение некоторого времени и ведро заполняется полностью. Когда поток снова активизируется, он сможет передать *b* байт данных со скоростью передачи по каналу без остановки. Это может в принципе лишить другие потоки доступа к сетевым ресурсам на время передачи блока пакетов. Более того, при таком поведении может потребоваться большой объём буферизации в сетевых устройствах на пути прохождения пакета.

5.4. Tspec передатчика

Согласно модели интегрированных услуг, Tspec приемника описывает поток трафика, используя параметры <*b, r, p, m, M*>. Первые три параметра (*b, r, p*) являются параметрами token bucket, как описывалось выше. В дополнение к ним используется параметр *M*, определяющий максимальную длину дейтаграммы в потоке, и параметр *m*, определяющий минимальную контролируемую (policed) единицу. Параметр *m* используется сетевыми узлами для оценки необходимой полосы пропускания, которая должна выделяться потоку. Следует отметить, что приложение может определить только минимальную длину пакета, который оно генерирует, в плане данных или

полезной нагрузки внутри пакета. К этим данным прикрепляются дополнительные заголовки для образования пакета, который может быть передан с применением конкретной технологии канального уровня. Таким образом, сетевые узлы используют значение *m* для расчёта максимальной полосы пропускания, требуемой для переноса пакетов потока по конкретной технологии канального уровня. Это расчёт основан на отношении *m* к длине заголовка канального уровня [RFC2210]. Максимальная длина *M* необходима узлам для того, чтобы определить, могут они или нет обслужить такой пакет, а передатчик может выбирать *M* на основе максимальной единицы передачи (MTU), поддерживаемой по тракту передачи.

5.5. Rspec приемника

Приемник использует Rspec для указания количества ресурсов, которое он желает зарезервировать для получения желаемого уровня функционирования. Когда приемник посылает сообщение RESV, в сообщение входит объект RSVP FlowSpec, состоящий из спецификации трафика (Tspec приемника) и Rspec. Параметры Tspec (*r, b* и *p*) устанавливаются для отражения параметров трафика желаемого приемником резервирования. Rspec задаёт требуемый тип обслуживания и параметры, относящиеся к обслуживанию.

В случае обслуживания с контролируемой нагрузкой (Controlled Load, CL), приемник задаёт только параметры Tspec, относящиеся к выделению необходимой полосы пропускания, и не задаются никакие параметры Rspec, кроме требуемого типа обслуживания (CL).

В случае гарантированного обслуживания (Guaranteed Service), приемник будет задавать Tspec и два относящихся к обслуживанию параметра в Rspec. Этими параметрами являются требуемая скорость обслуживания *R*, которая должна быть не ниже, чем параметр *r* token bucket, содержащийся в Tspec, а также элемент резерва времени (slack term). Приемник выбирает элементы *R* и *S* для получения требуемых гарантий по полосе пропускания и задержке. Использование этих элементов будет рассмотрено позже.

5.6. Регулирование и формирование потока

Возможность описывать характеристики потока трафика является необходимой для надлежащего обеспечения ресурсов в инфраструктуре сети. После того, как определена соответствующая спецификация трафика, важно убедиться, что поток остаётся в указанных в спецификации трафика рамках и не передаёт избыточный трафик. Если потокам позволить передавать избыточный (дополнительный) трафик, сетевые ресурсы не смогут обеспечиваться должным образом. Необходимо также, чтобы поток поддерживал свои заданные показатели трафика при его перемещении через сеть. Всё это обеспечивают функции регулирования и формирования трафика в модели IntServ.

Регулирование или *полисинг (policing)* (обеспечение соответствия пользовательских трафиков заданным показателям)- это термин, используемый для описания функции принудительного регулирования. Она обычно реализуется в хостах или на границах сети, где поток трафика впервые входит в сеть. В протоколе RSVP трафик потока описывается путём использования упомянутой выше модели token bucket. Навязывание параметров Tspec потока является задачей регулирующего агента, путём использования системы token bucket. Для каждого потока этот регулятор назначает входной буфер для хранения пакетов, поступающих к регулятору. Регулятор также поддерживает состояние потока в плане состояния потока модели token bucket и пиковую скорость, которая должна принудительно назначаться на выходе регулятора. Используя эту модель, трафик приложения будет принужден регулятором оставаться в рамках заданных границ; в противном случае будет риск задержки или даже потери трафика. При этом сеть может быть уверена, что потоки соответствуют своим заданным показателям трафика, что позволяет точно обеспечивать ресурсы.

Поточное регулирование трафика может оказаться дорогим. Таким образом, предпочтительно, чтобы хосты регулировали их зарезервированный трафик и освобождали от этого бремени инфраструктуру сети. Такая распределённая модель, однако, эффективна только в том случае, если хостам можно передать полномочия по фактическому регулированию их трафика. Когда таких полномочий у хостов нет, осуществление функций регулирования трафика с целью обеспечения гарантий, что потоки с неправильным поведением не повлияют отрицательно на QoS других, правильных потоков, ложится на инфраструктуру сети.

Формирование (shaping) - концепция, схожая с регулированием, только формирование трафика обычно применяется к потокам внутри сети. Если даже поток уже прошёл процедуру регулирования и соответствует своим заданным параметрам (спецификациям) трафика, его путешествие через сеть может тем не менее привести к возможной потере своих пакетов. В результате этого поток с правильным поведением может иметь потери просто по причине наложения задержек доставки пакетов, вносимых самой сетью. Для предотвращения самой возможности такой потери может использоваться процедура формирования, которая переформировывает поток с неправильной формой обратно в поток с заданными показателями трафика. Опять, формирователь будет использовать механизм token bucket. Однако, при использовании формирователя будет обеспечиваться обширная буферная область, такая, чтобы большие пачки данных из потока могли в нёй храниться во время процедуры их формирования.

5.7. Классификатор пакетов

В сегодняшней сети Интернет традиционный маршрутизатор выполняет две основные задачи по пересылке пакета. Во-первых, он должен выполнить поиск адреса назначения пакета для определения исходящего интерфейса, который будет использоваться для пересылки пакета в сторону своего пункта назначения. Эта задача является формой классификации пакетов, при которой маршрутизатор ищет одно из полей в заголовке пакета. Во-вторых, он должен скоммутировать (перенести) пакет из входящего канала в один из исходящих каналов на основе найденного адреса. В связи с последними усовершенствованиями аппаратной коммутации, скорость пересылки пакета зависит главным образом от эффективности алгоритма поиска адреса.

До недавнего времени задача поиска адреса была относительно лёгкой. Маршрутизатор ведёт базу данных пересылки, которая состоит из IP-адресов назначения и адреса ближайшего маршрутизатора в сети, которому должны пересылаться пакеты. В случае трафика с адресацией конкретному устройству, по прибытии IP-дейтаграммы маршрутизатор ищет только IP-адрес назначения (либо частичный, либо полный) для принятия решения, каким образом переслать пакет. В некоторых случаях, для многоадресного трафика, маршрутизатор будет также анализировать IP-адрес источника в заголовке входящей IP-дейтаграммы для принятия решения о способе его пересылки.

Маршрутизаторы на периферии сети Интернет должны вести только небольшую базу данных пересылки, состоящую из нескольких сотен записей и, в таких случаях, алгоритмы поиска пакета или классификации могут использовать простой механизм, такой например, как алгоритм просмотра хешированных таблиц. Однако, в связи с ростом сети Интернет возросло количество возможных пунктов назначения и сетей адресата, что привело к резкому увеличению числа записей в таблицах пересылки. Для маршрутизатора, находящегося в центре сети Интернет, число таких записей может составлять десятки тысяч или больше. В этом случае эффективность алгоритма табличного поиска существенным образом влияет на скорость и эффективность пересылки пакетов. Это особенно справедливо в связи с продолжающимся ростом скорости передачи по каналу до гигабит в секунду и выше. В последние несколько лет были разработаны несколько алгоритмов быстрого поиска адресов, которые решают эту проблему, позволяя таким маршрутизаторам с лёгкостью обслуживать гигабитные каналы.

Однако, быстрый поиск адресов решает только часть проблемы, если мы говорим о маршрутизаторах, поддерживающих механизмы QoS. Надо иметь в виду, что поток, обслуживаемый протоколом RSVP, идентифицируется просмотром более чем одного поля (IP-адрес назначения) в IP-заголовке. В частности, классификатор пакетов должен выполнить просмотр ни много ни мало пяти полей в IP-заголовке и заголовке транспортного протокола верхнего уровня (IP-адрес источника, порт источника, идентификатор IP-протокола, IP-адрес назначения, порт назначения) входящей дейтаграммы для того, чтобы определить, какому потоку он принадлежит. Обычно это требует механизма классификации, который должен находить точное или наилучшее соответствие путём анализа пяти полей в заголовке (заголовках) пакета. Если соответствие получается путём последовательного анализа полей, процесс будет намного более медленным, чем в случае поиска одного-единственного адреса. Кроме того, такой поиск должен осуществляться эффективно для возможности поддержки мультимегабитных или выше скоростей передачи по каналу. Это требование определённо является препятствием на пути внедрения RSVP в магистральных маршрутизаторах, которые пересылают трафик на гигабитных скоростях или выше.

К счастью, исследования, проводимые в настоящее время, показывают, что существует эффективный механизм поиска, который позволяет находить наилучший фильтр для пакета на гигабитных скоростях даже когда таблица поиска содержит 100 тысяч записей. Многие производители коммутационных систем и маршрутизаторов внедряют такие алгоритмы в своих аппаратных реализациях системы поиска, снижая вероятность узких мест для обеспечения QoS. Другим способом снижения непроизводительных потерь при поиске является анализ одного-единственного поля “метка потока” (flow label) в заголовке IP-дейтаграммы для идентификации потока. Такую возможность обеспечивает Интернет-протокол версии 6 (IPv6) путём включения поля “метка потока” в свой IP-заголовок. Метка потока может затем использоваться как единый смежный фильтр в таблице пересылки, позволяя использовать намного более простые и эффективные алгоритмы поиска. Поскольку популярность IPv6 растёт, простота данного подхода должна положительно сказаться на реализации QoS.

5.8. Планировщик пакетов

Как показано на рис. 5.1, планировщик пакетов функционирует в мультиплексной точке, куда поступает трафик от одного или более входных каналов и где он затем группируется в одну или более очередей или классов. Каждый из этих классов содержит один или более потоков, в зависимости от алгоритма планирования. Каждый класс может представлять собой класс трафика в зависимости от требуемых показателей обслуживания или же административный класс в зависимости от способа совместного использования канала разными организациями. Задачей планировщика является управлять взаимодействиями между различными классами и предоставлять требуемые рабочие характеристики для каждого из классов. При рассмотрении различных альтернативных вариантов планирования пакетов, мы будем учитывать следующие желаемые характеристики, так как они важны для предоставления необходимого QoS:

- *Защита потоков*: Планировщик должен гарантировать, что потоки с неправильным поведением не влияют отрицательно на работу правильных потоков, которые остаются в рамках своих спецификаций трафика. В противном случае, если поток-нарушитель передаёт трафик на высокой

скорости, он может захватывать значительную долю пропускной способности канала, что не позволит другим потокам получать предназначенную им долю пропускной способности канала. Планировщик должен предотвращать такие ситуации с помощью создания межсетевых экранов между потоками, и это обычно достигается путём ведения отдельной очереди для каждого потока.

- *Сквозные гарантии*: В зависимости от требуемого класса обслуживания IntServ, потокам нужны гарантии по пропускной способности, задержке или по обеим, и планировщик должен реализовать механизмы назначения ресурсов для обеспечения таких гарантий.
- *Гибкое назначение ресурсов*: Планировщик должен иметь возможность гибкого назначения ресурсов таким образом, чтобы он мог отдельно управлять попакетной задержкой из полосы пропускания, выделенной потоку. Например, некоторые алгоритмы планирования могут связывать вместе задержку и пропускную способность при определении количества назначаемых ресурсов. В таких случаях, связывание может привести к неэффективному использованию ресурса, если потоку с низкими требованиями по задержке присваивается более значительная полоса пропускания, чем требуется для выполнения требований по задержке.
- *Интегрированная поддержка трафика реального времени и трафика наилучшей попытки*: В связи с тем, что значительная доля Интернет-трафика является трафиком наилучшей попытки, важно, чтобы планировщик поддерживал совместно как трафик реального времени, так и трафик наилучшей попытки, без зависания трафика наилучшей попытки.
- *Простота и эффективность реализации*: Реализация механизмов управления телетрафиком производителями сетевого оборудования важна для широкого внедрения RSVP и IntServ. Используемый алгоритм планирования должен быть простым в реализации и должен обеспечивать эффективное использование ресурсов. В частности, пространственные и временные требования алгоритма планирования важны для определения его эффективности. При наличии набора очередей с определёнными максимальными длинами, время их упорядочивания в соответствии с последовательностью передачи и время вытаскивания следующего пакета

для передачи из ряда пакетов, могущих быть переданными, определяют временную сложность алгоритма.

Ниже будут рассмотрены некоторые алгоритмы планирования, используемые на практике, и их ценность в обеспечении сквозных гарантий QoS.

5.9. Планирование по алгоритму FCFS

Большинство традиционных маршрутизаторов в сети Интернет используют дисциплину обслуживания «первым пришёл, первым обслужен» (FCFS) на выходных каналах. В FCFS пакеты обслуживаются в порядке их поступления и назначаются в выходную очередь для передачи друг за другом. Если поток-нарушитель посылает пакеты на высокой скорости он может заполнить большинство буферов в выходной очереди и, следовательно, может занять произвольную долю пропускной способности выходного канала, препятствуя обслуживанию других, правильных потоков. На практике, такая ситуация не является большой проблемой в сети Интернет до тех пор пока для большей части трафика используется в качестве транспортного протокола протокол TCP, поскольку TCP поддерживает механизм избежания перегрузок в составе функции управления перегрузками на хосте источника. В соответствии с функцией управления перегрузками, реализованной в протоколе TCP, для каждого потока периодически проверяется доступную пропускную способность в узких местах на пути прохождения потока и снижает скорость передачи при обнаружении состояния перегрузки. Таким образом, потоки, использующие TCP, обычно получают справедливую долю доступной пропускной способности. Однако, если поток-нарушитель использует неправильную реализацию TCP или иного транспортного протокола, не обеспечивающую механизм ограничения, он может захватывать всё большую и большую долю

пропускной способности за счёт других потоков, и алгоритм планирования FCFS не может этому воспрепятствовать.

5.10. Простое планирование с приоритетами

В принципе, алгоритм планирования FCFS не различает требования разных потоков. По мере роста популярности сети Интернета, передаваемый по ней трафик уже не ограничен передачей данных с использованием протокола TCP. Мультимедийные приложения, такие как аудио- и видеоконференции и презентации в реальном времени также используют сеть Интернет и нуждаются в своевременной доставке своего трафика. Одним из простейших способов различать потоки является реализация схемы приоритетов, в которой очередность передачи пакетов определяется приоритетом, и пакеты с наивысшим приоритетом из числа пакетов, находящихся в буфере, всегда передаются первыми.

Этот алгоритм планирования прост, эффективен и быстр в принятии решения о планировании, пока непустые очереди всегда упорядочиваются в порядке снижения приоритета.

Приоритет может присваиваться потокам на основе их заданных показателей QoS. Например, пакетам аудиоинформации, для которых задержки должны быть минимальными, присваивается наивысший приоритет. Вслед за ними по приоритету могут стоять пакеты видеоданных, для которых требуется бóльшая полоса пропускания, но допустимы более высокие задержки, и т.д. Однако, у простого планирования с приоритетами есть ряд недостатков. Во-первых, даже если алгоритм контроля допуска гарантирует, что ресурсы не назначаются избыточно на долговременной основе, индивидуальным потокам, имеющим более низкий приоритет, не может гарантироваться доля ресурса за более короткий отрезок времени. Например, когда почти одновременно поступают пачки пакетов, имеющих более высокий приоритет, пакеты, имеющие более низкий приоритет, остаются в состоянии ожидания до тех пор, пока не будут переданы все пакеты с более высоким приоритетом. Это будет приводить к зависанию ресурса для потоков с более низким приоритетом, а также к очень большим задержкам для других потоков. Во-вторых, трудно обеспечить попакетные гарантии по величине задержки для индивидуальных

потоков. Это справедливо даже для потоков, которым присвоен наивысший приоритет. К примеру, потоки аудиоинформации состоят из небольших пакетов (например, 64 байт) и каждому пакету может требоваться строгая гарантия по попакетной задержке. Однако, при схеме с приоритетами не обеспечивается различение пакетов, принадлежащих одному и тому же классу приоритета. Таким образом, каждый поступающий пакет должен ждать передачи всех предыдущих пакетов, имеющих такой же приоритет, что приводит к непредсказуемым задержкам.

Проще говоря, базовые схемы приоритетов не годятся, когда требуется обеспечить попоточные гарантии QoS.

5.11. Циклическое круговое планирование

Альтернативным способом планирования является схема циклического кругового планирования (round-robin scheduling), в которой планировщик посещает каждую непустую очередь по очереди, и за каждое такое помещение обслуживает один пакет. Эта схема реализует некоторое подобие справедливого распределения между различными классами трафика и препятствует получению любым потоком неконтролируемой доли пропускной способности выходного канала. Опять же, эту схему легко реализовать. Она также является эффективной, поскольку планировщику требуется постоянный отрезок времени для принятия решения по планированию, потому что планировщику нужно только просмотреть следующую по кругу непустую очередь для выбора следующего пакета для передачи.

Однако, эта схема имеет два недостатка. Во-первых, она не предоставляет никакой гибкости в назначении полосы пропускания пропорционально требованиям индивидуальных потоков. Во-вторых, она не принимает во внимание длины пакетов. В IP-сети дейтаграммы имеют переменную длину, и таким образом циклическое круговое планирование может привести к неравному распределению пропускной способности. Предположим, что все пакеты в первой очереди имеют длину 100 байт, в то время как все пакеты в *n*-ой очереди имеют длину 1000 байт. При использовании циклической круговой схемы планирования, потоки в *n*-ой очереди получат гораздо большую (примерно в 10 раз) долю пропускной способности, чем потоки в первой

очереди. Кроме того, здесь отсутствует контроль за полосой пропускания, выделяемой каждой из очередей.

Разновидностью циклического кругового планирования является *взвешенное циклическое планирование (weighted round-robin, WRR),* когда каждой очереди присваивается вес, пропорциональный доли пропускной способности, присвоенной ей. Предположим, что каждой очереди *i* присвоен вес W_i. В алгоритме планирования WRR, планировщик обслуживает W_i пакетов из очереди *i* при каждом своём посещении очереди. Таким образом, если для учёта различных длин пакетов и назначений полосы пропускания присвоены веса, WRR может обеспечить требуемое разделение пропускной способности между очередями. При этом предполагается, что средняя длина пакета для каждого потока известна заранее, а это условие, вполне вероятно, для многих потоков не может быть выполнено. Однако, схема WRR не удовлетворяет другого требования. Она не учитывает требований по задержке, налагаемых индивидуальными потоками. Пакеты в потоке, для которых требуются очень короткие задержки (например, аудио), должны ждать своей очереди на передачу, покуда не будут обслужены по схеме циклического кругового планирования пакеты из других непустых очередей. Ниже будет рассмотрена разновидность циклического кругового планирования, так называемая схема дефицитного циклического кругового планирования (deficit round-robin scheduling), которая позволяет решать некоторых из этих проблем.

5.12. Взвешенная справедливая очередь

До сих пор мы рассматривали алгоритмы планирования, которые являются всё более и более эффективными для обеспечения изоляции (развязки) потока среди многих потоков и также обеспечивают справедливое или пропорционально распределение пропускной способности канала между различными потоками. Демерсом и коллегами был предложен идеальный алгоритм справедливого распределения, называемый *побитным циклическим круговым планированием (bit-by-bit round robin, BRR*), в котором каждый поток посылает в любой момент времени один бит по схеме циклического кругового планирования. Эту концепцию иллюстрирует рис. 5.7, где планировщик имитирует модель временного мультиплексирования (TDM) для *N* потоков. Один поток присваивается каждому каналу TDM, и бит из отличного потока

обслуживается в каждом временном канале. Таким образом, когда пакет поступает для потока F1, будет передаваться один-единственный бит из пакета, и будет передаваться один бит из других потоков до передачи другого бита из того же пакета. В результате, в любой момент времени каждый поток получает одинаковую долю пропускной способности.

Поскольку на практике реализовать такую схему невозможно, Демерс и коллеги предложили приближённый вариант BRR, называемый *справедливая очередь* (*fair queuing, FQ*), который имитирует алгоритм BRR. В схеме FQ, когда поступает пакет для потока, планировщик рассчитывает время, когда маршрутизатором должен был бы быть передан последний бит пакета согласно схеме BRR. На основе этого времени отправления пакет затем помещается в очередь пакетов, отсортированных по срокам их отправления. Разновидностью FQ является *взвешенная справедливая очередь* (*weighted fair queuing, WRQ*), также предложенная Демерсом и коллегами, в которой потокам назначаются различные веса, отражающие их требования по полосе пропускания или задержке. При WFQ дисциплина обслуживания аналогична FQ за исключением того, что в каждом цикле потоком передаётся количество бит, пропорциональное его весу, а не единичный бит, как в схеме BRR.

Схема WFQ имеет большинство требуемых свойств, перечисленных в начале данного раздела. Во-первых, WFQ создаёт межсетевые экраны между потоками, поскольку каждый поток обслуживается в соответствии со своей справедливой долей.

Во-вторых, WFQ обеспечивает чрезвычайную гибкость распределения ресурса для разнообразных целей, например совместного использования канала или обеспечения гарантированных граничных значений задержки. При совместном использовании канала задача состоит в том, чтобы обеспечить распределение пропускной способности канала на основе заранее заданных критериев, таких как тип протокола (TCP, IPX или SNA), тип приложения или источник трафика (подсеть или организация, которая платит за долю пропускной способности канала). Важным свойством является возможность обеспечить такое распределение при построении межсетевых экранов, но одновременно с этим - возможность использовать канал любой смесью трафика, если имеется свободная ёмкость канала. WFQ облегчает совместное использование канала, потому что может использоваться отдельный класс для

группирования потоков, имеющих одинаковый критерий, и каждому классу может назначаться вес на основе выделенной ему ёмкости. При WFQ, избыточный трафик обслуживается только тогда, когда класс с выделенной ёмкостью не имеет никакого незавершённого трафика. WFQ может также объединять обработку трафика реального времени с обслуживанием трафика наилучшей попытки. Это возможно по той причине, что трафику наилучшей попытки может назначаться фиксированная доля пропускной способности канала для избежания ситуаций зависания. Кроме того, любая ёмкость, не используемая потоками реального времени, может использоваться в WFQ для пересылки трафика наилучшей попытки.

В-третьих, если допустить, что поток пересекает на своём пути несколько WFQ-планировщиков в маршрутизаторах, WFQ может использоваться для обеспечения сквозной гарантии как по пропускной способности, так и по порогу задержки в наихудших условиях. Следует отметить, что WFQ гарантирует пропускную способность, потому что каждый участок переприёма гарантирует свою долю пропускной способности на основе назначенного веса. Вес может подсчитываться на основании запроса резервирования, в котором задаются параметры token bucket. Поскольку доля пропускной способности гарантируется на каждом участке переприёма, Парех и Галлагер определили важное пороговое значение для сквозной задержки (“из конца в конец”) для наихудших условий, которую получает каждый пакет в потоке, пересекающий на своём пути ряд WFQ-планировщиков. Их результат рассчитан на базе скорости обслуживания *R*, назначенной каждому потоку в каждом WFQ-планировщике. Чем выше скорость обслуживания, тем быстрее обслуживаются пакеты потока на каждом маршрутизаторе, что приводит к более низкой сквозной задержке. Таким образом, приёмник может должным образом выбирать свои параметры обслуживания для того, чтобы запросить скорость обслуживания *R* такой, чтобы это привело к требуемому порогу задержки при использовании на пути прохождения механизма планирования WFQ. Например, при заданных параметрах token bucket (*b, r, p*) в Tspec, приёмник должен выбирать скорость обслуживания *R* (в Rspec) более высокой, чем *r*. При возрастании *R*, сквозная задержка будет уменьшаться, поскольку будет уменьшаться время ожидания обслуживания, вызывающее задержку, на каждом узле.

Класс гарантированного обслуживания использует данный результат. Следует помнить, что сквозная задержка имеет две компоненты, а именно фиксированную компоненту, определяемую задержкой передачи и распространения, и переменную компоненту (задержку, связанную с нахождением в очереди). Задержка, связанная с нахождением в очереди, определяется Tspec (параметрами token bucket) и Rspec, который задаёт скорость обслуживания *R*, которую запрашивает приложение. Ниже будет рассмотрено в упрощённом виде, каким образом планирование WFQ обеспечивает порог сквозной задержки на основе параметров гарантированного обслуживания.

Рассмотрим источник, который задаёт размер token bucket *b* как часть своего Tspec *<p, r, b, m, M>* , и предположим, что процедура резервирования запрашивает скорость обслуживания *R*. Для источника, ограниченного token bucket (*b, r*) и обслуживаемого со скоростью *R* (R не меньше *r*), Парех и Галлагер установили, что граничное значение задержки, вносимой в такой поток при идеальном BRR-планировщике, составляет *b/R*.

Осмыслим этот результат в свете WFQ-планировщика, который является приближением идеального BRR-планировщика. Для простоты проигнорируем параметры Tspec и Rspec за исключением *b* и *r* (*R>r*). В этом случае, если источник посылает поток с заданными Tspec (*b*) и Rspec (*R*), мы можем допустить, что такой поток будет посылать одновременно пачку из не более чем *b* байт. Кроме того, мы будем предполагать, что поток будет обслуживаться рядом WFQ-планировщиков на пути прохождения потока с одинаковой скоростью обслуживания *R*. Чтобы ещё больше упростить наше обсуждение, предположим, что пачка состоит из одного пакета длиной *b* байт, и ограничим наше рассмотрение сквозной задержки только одним пакетом, передаваемым в потоке. Парех и Галлагер доказали, что сквозная задержка, которую имеет такой пакет, представляет собой сумму трёх частей:

1. *b/R* секунд времени, требуемого для обслуживания *b* байт со скоростью *R* байт/с. Даже несмотря на то, что пакет пересекает на своём пути ряд WFQ-планировщиков, этот ряд планировщиков ведёт себя так, как если бы пакет обслуживался одним-единственным планировщиков, игнорируя задержку передачи на каждом участке переприёма.

2. Дополнительная задержка вносится в WFQ-планировщике, если пакет поступает сразу после того, как он бы получил обслуживание при аппроксимации BRR. Надо учитывать, что потоку гарантировано посещение и обслуживание со скоростью *R*, так что пакет длиной *b* байт получит шанс получить обслуживание не позднее чем через *b/R* единиц (интервалов) времени. Эта задержка должна суммироваться по всем узлам, кроме одного их узлов на маршруте, для учёта наихудшей ситуации, когда пакет поступает на каждый WFQ-планировщик сразу после того, как он получил бы обслуживание.
3. Исходя их допущения, что скорость передачи по каналу составляет *L* (байт/с) на каждом участке переприёма, пакет будет иметь задержку передачи, равную *b/L* на каждом участке переприёма, которая должна суммироваться по всем участкам переприёма на пути прохождения пакета.

Это упрощённый анализ, основанный на идеальной ситуации. На практике, спецификация гарантированного обслуживания должна учитывать переменные скорости передачи по каналу, длины пакетов и пиковые скорости. Кроме того, надо также учитывать различные реализации механизма WFQ, предоставляя сетевым элементам на маршруте возможность задавать параметры ошибок (error terms) *C* и *D*, которые учитывают отклонения от точной модели WFQ.

5.13. WFQ и гарантированное обслуживание

Гарантированное обслуживание специфицируется исходя из допущения, что поведение каждого сетевого узла, который поддерживает обслуживание, может быть смоделировано использованием так называемого струйного сервера (fluid server), функционирующего со скоростью обслуживания *R*. Достоинство этой модели состоит в том, что можно описать обслуживание, которое получит поток на зарезервированной скорости *R* независимо от текущей нагрузки на узел или количества других обслуживаемых потоков. Модель струйного сервера была введена Парехом и Галлагером и имеет свойства, описанные ранее. Для простоты обсуждения мы будем исходить из допущения, что узел, реализующий вышеупомянутый идеальный BRR-планировщик, имитирует такой струйный сервер.

Как часть процедуры RSVP AdSpec, каждый узел *i* экспортирует параметры C_i и D_i , описывающие уровень обслуживания, который он может предоставить потокам, проходящим через него. Приёмник принимает эти параметры и интерпретирует их в контексте требуемой скорости обслуживания *R*. В частности, C_i и D_i учитывают отклонение узла *i* от струйного сервера, который функционирует на скорости *R*. Вспомним, что мы обсуждали, каким образом можно подсчитать сквозную задержку на основе результата Пареха и Галлагера, исходя из идеального струйного сервера. Члены ошибок C_i и D_i определяют, что поток, обслуживаемый на скорости *R*, получит дополнительную задержку в $(C_i/R + D_i)$ единиц на узле *i* вследствие его отклонения от идеального поведения. Учитывая сумму членов ошибок на пути прохождения потока, приёмник может теперь подсчитать максимальную сквозную задержку, ожидаемую при использовании гарантированного обслуживания.

Для гарантированного обслуживания, в Rspec приёмника задаётся, помимо скорости обслуживания *R*, ещё и член резерва времени (slack term) *S*. Член резерва времени указывает разницу между требуемой задержкой и задержкой, полученной при использовании скорости обслуживания *R*. Этот член резерва времени может использоваться сетевым элементом для сокращения резервируемого им ресурса для потока. Приёмник может использовать комбинацию членов *R* и *S* для принятия решения о запрашиваемом уровне резервирования ресурса. Например, приёмник может использовать дополнительную задержку для сокращённого резервирования, повышая таким образом вероятность его резервирования для сокращения суммы счета за услугу.

Он также может использовать эту информацию для подстройки своих собственных параметров. Рассмотрим видео-плейер, получающий видео-презентацию, которая осуществляет резервирование на скорости *R*. Предположим, что сквозная задержка, ожидаемая при использовании скорости *R* (при $S = 0$), составляет всего 100 миллисекунд. Предположим затем, что плейер может поместить в буфер 250 миллисекунд видео перед его воспроизведением без ухудшения качества воспроизведения. Плейер может позволить себе задать резерв времени 150 миллисекунд и значительно снизить требуемый уровень резервирования ресурса без ущерба качеству изображения.

5.14. Недостатки WFQ

Одним из существенных недостатков метода FQ (и WFQ) является то, что он требует ведения сортированной очереди и вставки в сортированную очередь. Когда задействовано большое число потоков, эта процедура дорогая и повышает стоимость реализуемых маршрутизаторов. Кроме того, WFQ требует запоминания состояния планировщика на попоточной основе, и это увеличивает объём требуемой памяти в маршрутизаторах, в которых реализован WFQ.

5.15. Дефицитное циклическое круговое планирование

Более дешёвая схема приближения к планированию на основе справедливой очереди, так называемое *дефицитное циклическое круговое планирование (deficit round-robin)* (DRR) преодолевает некоторые недостатки, присущие схеме планирования WFQ. В отличие от WFQ, которая может обеспечивать гарантии как по задержке, так и по пропускной способности, DRR концентрируется только на справедливом распределении пропускной способности, аналогично схеме взвешенного циклического планирования (WRR). Вспомним, что WRR требует, чтобы средняя длина пакетов в потоках была известна заранее с тем, чтобы обеспечить справедливое назначение полосы пропускания индивидуальным потокам. DRR снимает это требование.

DRR-планировщик ассоциирует с каждым потоком, который инициализирован в 0, относящийся к потоку параметр, называемый *дефицитом*. Кроме того, DRR-планировщик использует длину шага квантования (кванта) для того, чтобы решить, сколько байт данных обслуживать из каждого потока во время каждого посещения. Планировщик работает следующим образом:

1. Каждый раз, когда планировщик посещает очередь, он пытается обслужить из очереди количество байт, равное шагу квантования.

2. Если пакет в голове очереди не длиннее кванта, планировщик обслуживает пакет. Если пакет крупнее кванта, планировщик добавляет квант к счётчику дефицита потока и откладывает его обслуживание на следующий цикл.

3. При каждом посещении, если сумма показаний счётчика дефицита потока и длины кванта превосходит длину пакета в голове очереди, пакет обслуживается, а из показаний счетчика дефицита вычитается длина пакета.

Рассмотрим пример, в котором длина кванта составляет 500 байт, и имеются три потока F1, F2 и F3 с пакетами длиной 800, 300 и 900 байт соответственно в своих очередях. За первый цикл, показание счётчика дефицита потока F1 будет увеличено до 500, а пакет обслужен не будет. Также за первый цикл, пакет из потока F2 обслуживается, а показания его счётчика дефицита станут равными 200 (500 - 300). Наконец, пакет из потока F3 не обслуживается, а показания его счётчика дефицита также повышаются до 500. За второй цикл, пакет из потока F1 будет обслужен, а его счётчик дефицита будет установлен в 200 (500 + 500 - 800). Тем временем, очередь потока F2 теперь пуста, и его счётчик дефицита будет сброшен в 0. Наконец, пакет из потока F3 будет обслужен, а его счётчик дефицита будет установлен в 100. Следует обратить внимание, что потокам не позволяется накапливать кредиты, когда в очереди нет пакетов, с тем, чтобы в будущем не допустить несправедливости. По сравнению с предыдущим примером, длина кванта должна выбираться таким образом, чтобы она была больше, чем максимальная разрешенная длина пакета с тем, чтобы гарантировать, что при каждом цикле обслуживается по меньшей мере один пакет.

Во взвешенной версии DRR, каждому потоку назначается свой собственный квант в зависимости от требуемой доли пропускной способности. Например, если длина кванта по умолчанию составляет 500 байт, а потоку назначается квант длиной 750 байт, этот поток получает на 50 процентов больше полосы пропускания, чем другой поток, если только два эти потока является активными.

Самыми большими достоинствами схемы DRR являются её простота и лёгкость реализации, что делает её очень привлекательной как для программных, так и для аппаратных реализаций. Единственный недостаток DRR заключается в том, что, в отличие от схемы WFQ, она сама по себе не обеспечивает твёрдые граничные значения задержки, поскольку пакет аудиоинформации малой длины может быть задержан за счёт любого другого активного потока на каждом узле на пути его следования.

Взвешенная версия DRR годится для реализации обслуживания с Контролируемой Нагрузкой (CL), так как она может гарантировать полосу пропускания потока CL, независимо от объёма трафика, передаваемого в других потоках. Однако, для соответствия требованиям обслуживания с CL, длина кванта, назначаемого каждому потоку CL, должна определяться его параметрами token bucket (Tspec), а функция контроля допуска должна ограничивать допуск других потоков с тем, чтобы обеспечить своевременное обслуживание потока CL. Поток CL с параметрами token bucket <*p*, *r*, *b*> ограничен передачей за любой интервал времени *t* не более ($b + r * t$) байт. Если DRR-планировщик может обеспечить, что ($b + r * T$) байт из каждого потока будут обслужены в каждом цикле DRR, длящемся интервал времени *T*, каждый поток CL получит надлежащее обслуживание.

5.16. Выводы

В данной главе была рассмотрена функция управления трафиком в маршрутизаторах и хостах, которая требуется для обеспечения сквозных гарантий QoS. В частности, рассмотрены компоненты классификации трафика и планирования пакетов функции управления телетрафиком. Были описаны различные алгоритмы планирования пакетов и их пригодность для обеспечения различных гарантий QoS. Было показано, что алгоритм взвешенной справедливой очереди (WFQ) имеет все требуемые свойства для обеспечения таких гарантий, но он дорог в реализации. Разновидность алгоритма справедливой очереди, DRR, прост и эффективен в реализации и может обеспечить обслуживание, необходимое для реализации гарантий по совместному использованию канала и полосе пропускания. В главе также обсуждалась специальная форма планирования, называемая регулированием трафика, которая принудительным образом обеспечивает соответствие потоков заданным показателям их трафика.

И еще один, весьма существенный вывод из материала данной главы – вывод о целеобразности продолжения исследований, рассмотренных выше.

ЗАКЛЮЧЕНИЕ

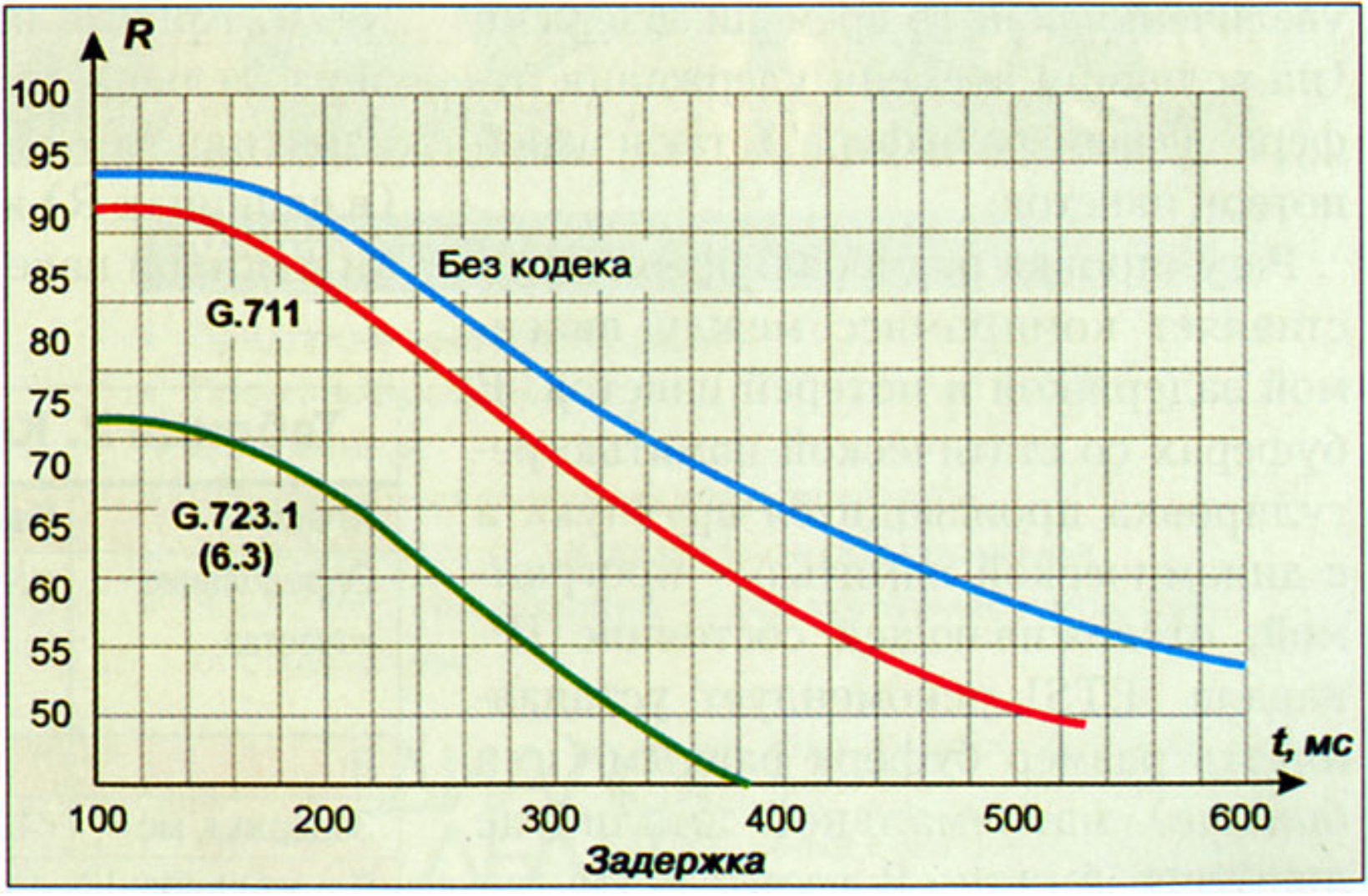
Литература

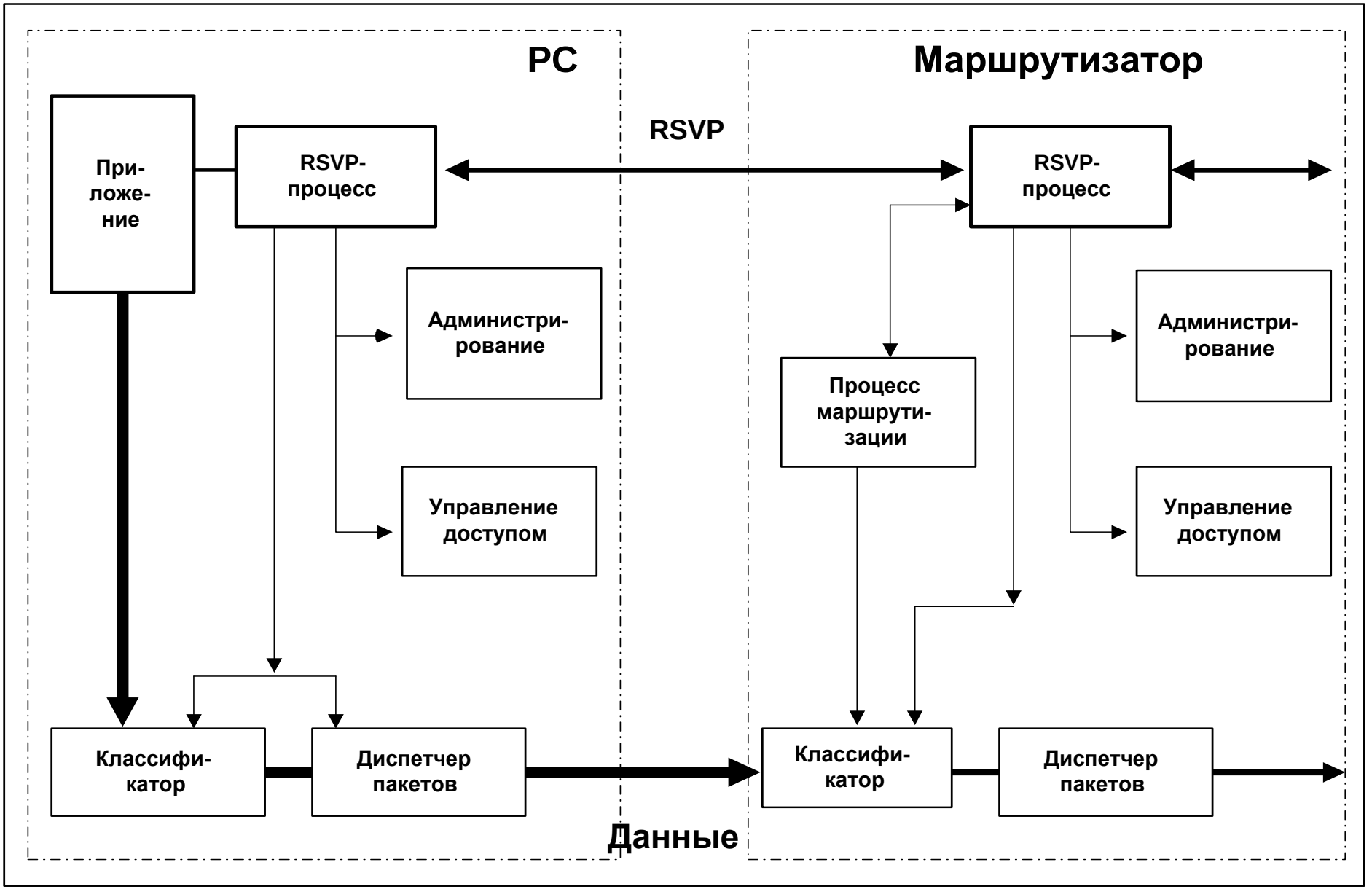
1. Будников В.Ю., Пономарев Б.А. Технологии обеспечения качества обслуживания в мультисервисных сетях / Вестник связи, 2000, №9.
2. Варламова Е. IP-телефония в России / Connect! Мир связи, 1999, №9.
3. Гольдштейн Б.С., А.В.Пинчук, А.Л.Суховицкий IP-телефония. М.: Радио и связь, 2001.
4. Кузнецов А.Е., Пинчук А. В., Суховицкий А.Л. Построение сетей IP-телефонии / Компьютерная телефония, 2000, №6.
5. Кульгин М. Технологии корпоративных сетей. Изд. «Питер», 1999.
6. Клейнрок Л. Теория массового обслуживания. Москва Машиностроение 1979
7. Мюнх Б., Скворцова С. Сигнализация в сетях IP-телефонии. – Часть I, II / Сети и системы связи, 1999. – №13(47), 14(48).
8. Уиллис Д. Интеграция речи и данных. В начале долгого пути. / Сети и системы связи, 1999. – №16.
9. Armitage Grenville. Quality of Service in IP Networks. – Macmillan Technical Publishing, 2000.
10. Black Uyless. Voice Over IP. – Prentice Hall, 1999.
11. Davidson J., Peters J. Voice Over IP Fundamentals. – Cisco Press, 2000.
12. Durham D., Yavatkar R.. Inside the Internet's Resource Reservation Protocol: Foundations for Quality of Service, 2000
13. Goncalves M. Voice Over IP Networks. – McGraw Hill Publishing, 1998.
14. Goralski W., Kolon M. IP Telephony. – McGraw Hill Publishing, 1999.
15. Hersent O, Gurle D., Petit Jean-Pierre. IP Telephony: Packet-Based Multimedia Communications Systems.- Addison-Wesley Pub Co, 2000.
16. ITU-T Recommendation G.711. Pulse Code Modulation of 3kHz Audio Channel. – 1988.
17. ITU-T Recommendation G.723.1. Dual Rate speech coder for multimedia communication transmitting at 5.3 and 6.3 kit / sec. – 1996.
18. ITU-T Recommendation G.728. Coding of Speech at 16 kbit / s Using Low-delay Code Excited Linear Prediction (LD-CELP). –1992.
19. ITU-T Recommendation G.729. Speech codec for multimedia telecommunications transmitting at 8 / 13 kbit / s. – 1996.

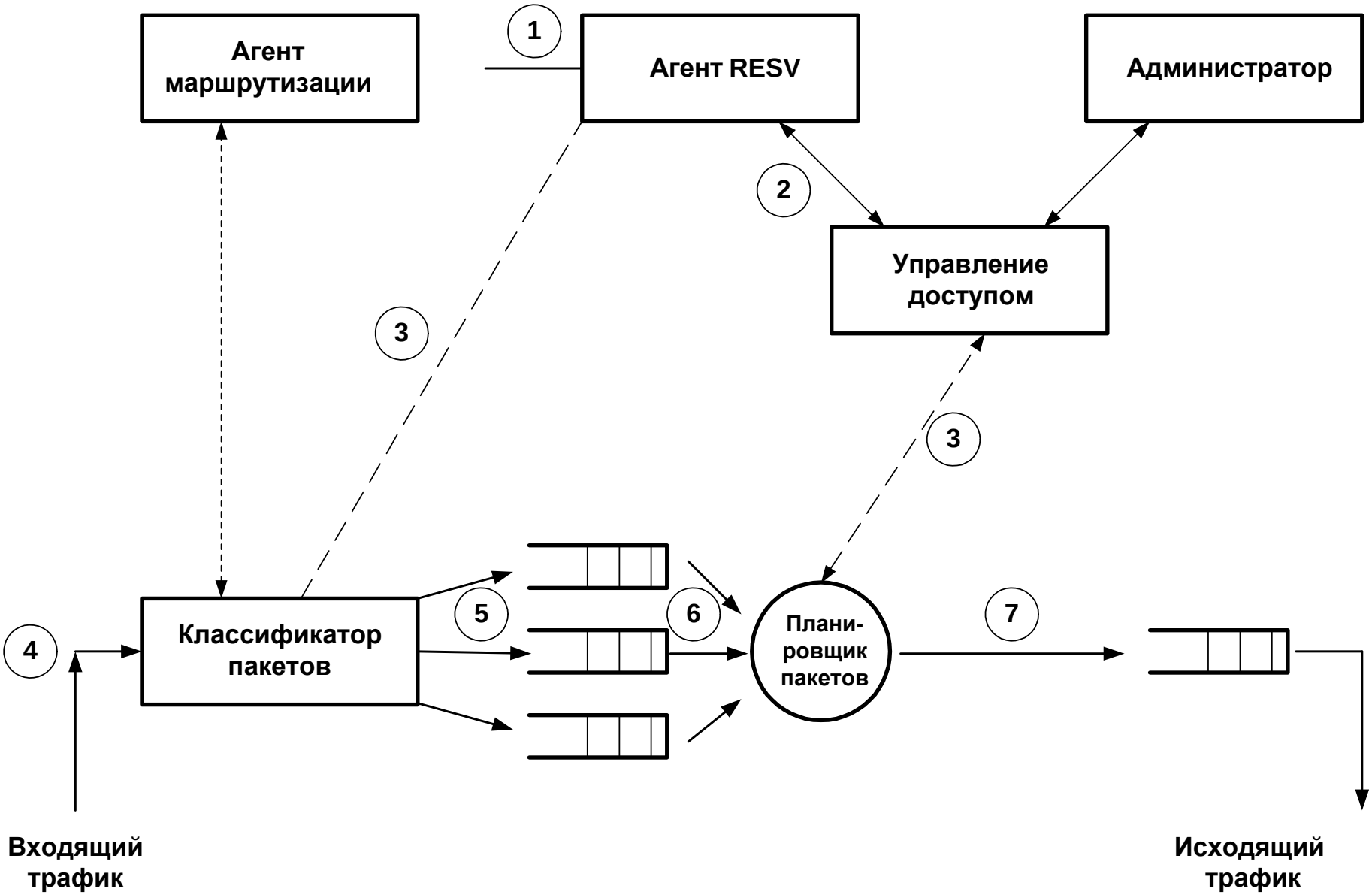
20. McDysan D. Phd.Qos & Traffic Management in Ip & Atm Networks

21. Minoli D., Minoli E. Delivering Voice over IP Networks / John Willey & Sons, Inc., 1998.
22. RFC 2205. Resource Reservation Protocol (RSVP). Ver.1. Functional Specification. – September 1997.
23. Uyless Black. Voice over IP, Prentice Hall PTR, 2000.
24. Walter J. Goralski, Matthew C. Kolon. IP telephony / The McGraw-Hill Co., Inc., 2000.
25. www.loniis.ru
26. www.cisco.com
27. www.networld.ru
28. www.etsi.com
29. ccc.ru
30. www.comptek.ru
31. www.rans.ru

Разработка функциональных моделей механизмов обеспечения гарантированного качества обслуживания в IP-сетях


R
100
95
90
85
80
75
70
65
60
55
50
Без кодека
G.711
G.723.1
(6.3)
t, мс
100
200
300
400
500
600
Задержка


PC
Маршрутизатор
RSVP
При-
ложе-
ние
RSVP-
процесс
RSVP-
процесс
Администри-
рование
Процесс
маршрути-
зации
Администри-
рование
Управление
доступом
Управление
доступом
Классифи-
катор
Диспетчер
пакетов
Классифи-
катор
Диспетчер
пакетов
Данные


Агент маршрутизации
1
Агент RESV
Администратор
2
Управление доступом
3
3
4
Классификатор пакетов
5
6
Плани-ровщик пакетов
7
Входящий трафик
Исходящий трафик