# 8. КАЧЕСТВО ОБСЛУЖИВАНИЯ В ВЫЧИСЛИТЕЛЬНЫХ СЕТЯХ

Под качеством обслуживания (Quality of Service, QoS) принято понимать предоставление пользователям и приложениям в сети предсказуемого сервиса по доставке данных. Конкретное же определение и параметры качества обслуживания, главным образом, определяются типом приложения. Так, например, для передачи голосового трафика, важнейшими параметрами QoS являются - задержка и вариация задержки на определенном интервале времени, в то время как потеря некоторой части пакетов допустима. Параметры качества обслуживания можно разбить на три группы:

- параметры пропускной способности (минимальная, средняя и максимальная скорость передачи) ;
- параметры задержек передачи пакетов (средние и максимальные величины задержек и вариаций задержек) ;
- параметры надежности передачи (уровень потерь и искажений пакетов).

Первые две группы относятся к производительности сети, последняя – к надежности. Качество обслуживания гарантируется для какого-либо потока данных.

Поток данных (flow) – последовательность пакетов, движущихся от источника A в пункт назначения B (C), каждый из которых может быть однозначно идентифицирован по 16-байтной комбинации из первых 64 байт IP-заголовка и/или заголовка TCP/UDP (номер порта приложения).

К потокам возможно применение таких понятий как агрегирование и дифференцирование. Агрегирование – представление потоков ПК предприятия в одном потоке данных абонента некоторого сервис-провайдера (прокси–сервер). Дифференцирование – поток данных от одного ПК представляется как совокупность потоков от разных приложений. Протоколы DiffServ работают с агрегированными потоками, а протокол RSVP – только с элементарными потоками. Выделяются три типа служб QoS:

1. Сервис «с максимальными усилиями» (с полным отсутствием QoS) - классические сети IP с дисциплиной обслуживания очередей FIFO.

2. Сервис с предпочтением (мягкий QoS) некоторые типы трафика обслуживаются лучше, чем остальные. Пример службы DiffServ и IEEE 802.1p.

3. Гарантированный сервис (истинный QoS) – основан на предварительном резервировании сетевых ресурсов для определенного потока перед передачей. Пример: службы RSVP, CBR, VBR.

Возможно комбинированное использование всех служб.

*Требования разных типов сетевых приложений*

В настоящее время нужны новые механизмы качества обслуживания, учитываюшие все многообразие требований, предъявляемых приложениями к сети.

В качестве основных критериев классификации были приняты три характеристики трафика: *относительная предсказуемость скорости передачи данных, чувствительность трафика к задержкам пакетов, чувствительность трафика к потерям и искажениям пакетов.*

В отношении предсказуемости скорости трафика приложения делятся на два больших класса.

*Потоки с равномерным трафиком* **(Stream)**- нагрузка поступает в сеть с более или менее постоянной битовой скоростью (CBR). Скорость потока может меняться, но она имеет легко вычисляемую верхнюю границу. Например, аудио потоки являются трафиком ПБС, а для элементарного голосового потока верхняя граница составляет 64 кбит/с.

*Потоки с пульсирующим трафиком* (Burst) - интенсивность нагрузки отличается высокой степенью непредсказуемости, когда периоды затишья сменяются передачей больших блоков данных. Для трафика характерна изменяющаяся битовая скорость (VBR). Так, в случае запроса данных интенсивность трафика может увеличиваться от нуля, когда данные не передаются, до полного насыщения полосы пропускания канала, когда служба стремится как можно быстрее передать данные. Строго говоря, любые сервисы

генерируют неравномерный трафик. Только коэффициент пачечности (пульсации) у пульсирующих трафиков находится в пределах 10÷200, а у потоковых - существенно меньше.

По критерию ***чувствительности трафика к задержкам*** различаются потоки (в порядке повышения чувствительности к задержкам пакетов).

*Асинхронные* - практически нет ограничения на время задержки. Пример - электронная почта.

*Синхронные* - чувствительны к задержкам, но допускают их.

*Интерактивные* - задержки могут быть замечены пользователями, но не сказываются на функциональности приложения (телеуслуги).

*Изохронные* - при превышении порога чувствительности к задержкам функциональность приложения резко снижается. Пример: передача голоса, когда при превышении порога задержек 100-150 мс качество воспроизводимого голоса резко ухудшается.

*Сверхчувствительные* к задержкам приложения - задержки доставки данных сводят функциональность к нулю. Пример: управление объектом в реальном времени.

Существует и более грубое деление приложений по этому принципу:

- *асинхронные* - приложения, не чувствительные к задержкам в широком диапазоне - до нескольких секунд;
- *синхронные* - приложения, на которые задержки оказывают негативное влияние.

Здесь под асинхронными понимаются асинхронные и синхронные из предыдущей классификации. Под синхронными - изохронные и сверхчувствительные. Интерактивные приложения могут быть отнесены как к асинхронным, так и к синхронным. Классификация приложений по их чувствительности к потерям пакетов делит приложения на две группы:

- **Чувствительные к потерям** приложения. Все приложения, передающие алфавитно-цифровые данные (текстовые документы, коды программ, числовые массивы и т.п.), обладают высокой

чувствительностью к потере отдельных фрагментов данных. Это традиционные сетевые приложения (файловый сервис, сервис баз данных, электронная почта и т. д.).

- **Устойчивые к потерям** приложения. Устойчивость к потерям объясняется тем, что небольшое количество отсутствующих данных можно определить на основе принятых (аудио- и видеоприложения). Устойчивость, т.е. процент допустимых потерь, оговаривается количественно ( например, 1%).

Между значениями трех характеристик качества обслуживания нет строгой взаимосвязи. То есть приложение с равномерным потоком может быть как асинхронным, так и синхронным, а, например, синхронное приложение может быть как чувствительным, так и нечувствительным к потерям пакетов.

Например, следующее сочетание характеристик приложения "порождаемый трафик - равномерный поток, изохронное, устойчивое к потерям" соответствует таким популярным приложениям, как IP-телефония, поддержка видео-конференций, аудиовещание через Internet.

*Параметры качества обслуживания*

Трем критериям классификации приложений (предсказуемость скорости трафика, чувствительность к задержкам и чувствительность к потерям) соответствуют три группы параметров, используемых при определении и задании требуемого качества обслуживания:

- **Параметры пропускной способности**. Это средняя, максимальная (пиковая) и минимальная скорости передачи данных.
- **Параметры задержек**. Используется средняя и максимальная величины задержек, а также среднее и максимальное значения вариаций задержек, то есть отклонений межпакетных интервалов в прибывающем трафике по сравнению с отправленным.
- **Параметры надежности передачи.** Используется процент потерянных пакетов, а также процент искаженных пакетов.

При определении всех этих параметров важно, на каком периоде измеряется данный параметр. Чем меньше этот период, тем более жесткими являются соответствующие требования качества обслуживания и тем труднее для сети их выдержать.

При заключении соглашения SLA с провайдером пользователь берет на себя обязательства, что предлагаемый трафик приложения не будет превышать максимальную скорость N. Провайдер, в свою очередь, гарантирует со стороны сети, что минимальная величина пропускной способности, предоставляемой этому приложению, будет не меньше N. Это необходимо для обеспечения приемлемого качества обслуживания трафика данного приложения.

Для приложений с пульсирующим трафиком качество обслуживания лучше всего характеризуется средней скоростью и максимальной скоростью, которая требуется в период пульсации. Обычно при этом оговаривается либо максимальное время пульсации, в течение которого приложению разрешено передавать данные с максимальной скоростью, либо максимальный объем данных, который можно передать в виде пульсации. Часто используется также вариант с заданием максимальной и минимальной границ скорости. В этом случае приложению гарантируется пропускная способность на уровне минимальной границы, достаточная для его удовлетворительного функционирования, а само приложение обязуется не направлять в сеть трафик со скоростью, превышающей максимальную границу.

### 8.1. Служба QoS

Сеть - это распределенная среда, состоящая из большого коггичества устройств для поддержки различных технологий и протоколов. Поэтому достаточно сложно заставить ее соблюдать единые требования по качественному обслуживанию различных видов трафика на всем протяжении составного пути от одного конечного узла до другого, то есть "из-конца-в-конец" *(end-to-end)*. Особенно, если в сети существуют многочисленные потоки

данных с трудно совместимыми характеристиками, например, пульсирующий файловый трафик и синхронный голосовой.

Для решения поставленных задач в сети необходима служба QoS. Эта служба имеет распределенный характер, так как ее элементы должны присутствовать во всех сетевых устройствах, продвигающих пакеты: коммутаторах, маршрутизаторах, серверах доступа. С другой стороны, работу отдельных сетевых устройств по обеспечению поддержки QoS нужно скоординировать, чтобы качество обслуживания было однородным вдоль всего пути, по которому следуют пакеты потока. Поэтому служба QoS должна включать также элементы централизованного управления, с помощью которого администратор сети может согласованно конфигурировать механизмы QoS в отдельных устройствах сети. Базовая архитектура службы QoS включает элементы трех основных типов, представленных на рисунке 8.1.

1. **Средства QoS** узла, выполняющие обработку поступающего в узел трафика в соответствии с требованиями качества обслуживания.
2. **Протоколы QoS-сигнализации** для координации работы сетевых элементов по поддержке качества обслуживания "из-конца-в-конец".
3. **Централизованные функции политики, управления и учета QoS**, позволяющие администраторам сети целенаправленно воздействовать на сетевые элементы для разделения ресурсов сети между различными видами трафика с требуемым уровнем QoS.

Средства QoS узла являются основным исполнительным механизмом службы QoS, так как именно они непосредственно влияют на процесс продвижения пакетов между входными и выходными интерфейсами коммутаторов и маршрутизаторов. Средства QoS узла могут включать элементы двух типов:

- *механизмы обслуживания очередей*;
- *механизмы "кондиционирования" трафика*.

*Механизмы обслуживания очередей* являются необходимым элементом любого устройства, работающего по принципу коммутации пакетов. Они могут поддерживать различные алгоритмы обработки пакетов, попавших в очередь, от

самых простых типа FIFO ("первым пришел - первым обслужен") до весьма сложных, поддерживающих обработку нескольких классов потоков, например, алгоритмов приоритетного или взвешенного обслуживания. По умолчанию в сетевых устройствах действует алгоритм обслуживания очереди F I F O, но он достаточен только для реализации сервиса "с максимальными усилиями", а для поддержки "истинных" сервисов QoS нужны более сложные механизмы.

*Механизмы второго типа ("кондиционирования" трафика*) могут реализовываться в сетевом узле для поддержания QoS. Дело в том, что поддержание качества обслуживания всегда означает создание таких условий, когда скорость продвижения трафика потока согласуется со скоростью поступления этого трафика в узел сети.

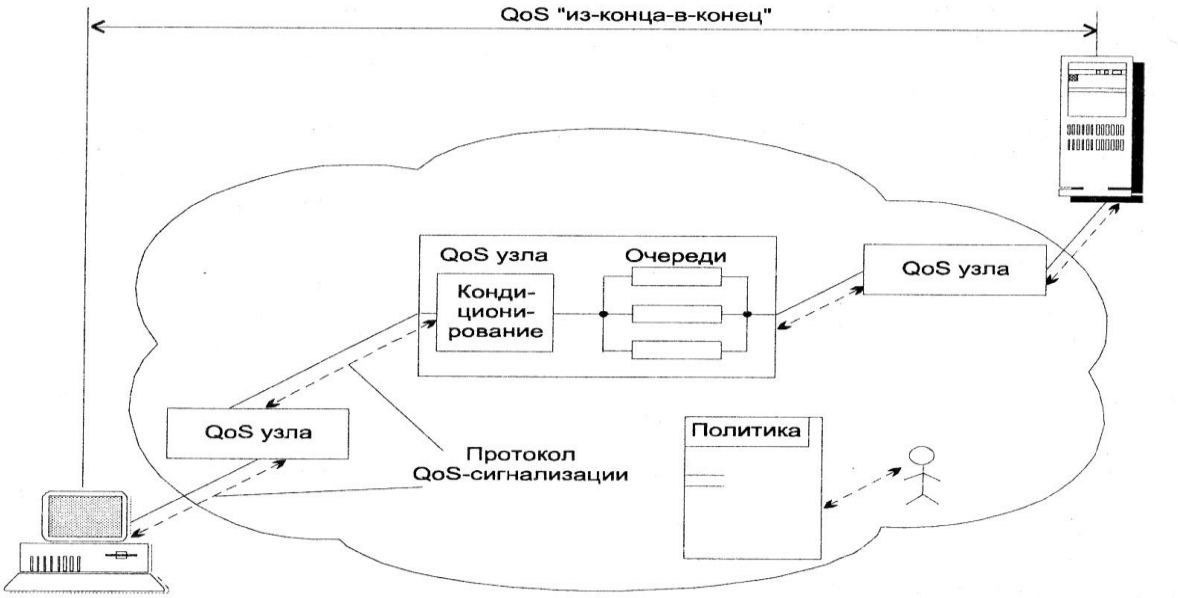


***Рис. 8.1.*** *Базовая архитектура средств QoS*

Очереди возникают в те периоды времени, когда скорость поступления трафика становится больше скорости его продвижения. Механизмы обслуживания очередей рассчитаны на работу как раз в такие периоды времени и нужны для того, чтобы потоки как можно меньше страдали от существования подобных периодов. Задержки от ожидания в очередях должны укладываться в параметры потока. Механизмы кондиционирования трафика решают задачу создания условий качественного обслуживания трафика другим способом - за счет уменьшения скорости поступления потока в данный узел настолько, чтобы

она всегда оставалась меньше, чем скорость продвижения этого потока. Механизм кондиционирования трафика обычно включает выполнение нескольких функций:

- *Классификация трафика*. Функция выделяет из общей последовательности пакетов пакеты одного потока, имеющего общие требования к качеству обслуживания. Классификация выполняется на основе различных признаков пакета - адресов источника и назначения, TCP/UDP портов, значения приоритета, значения метки потока (в версии IPv6).

- *Профилирование трафика на основе правил политики*. Для каждого входного потока имеется соответствующий ему набор параметров QoS, который называется профилем трафика. Профилирование трафика подразумевает проверку соответствия каждого входного потока параметрам его профиля. В случае нарушения параметров (например, превышения длительности пульсации или средней скорости) происходит отбрасывание или маркировка пакетов этого потока.

*Отбрасывание* некоторых пакетов снижает интенсивность потока и приводит его параметры в соответствие указанным в профиле. *Маркировка* пакетов без отбрасывания нужна для того, чтобы пакеты все же были обслужены данным узлом (или последующими по потоку), но с качеством обслуживания, отличным от указанного в профиле.

Для проверки соответствия входного трафика заданному профилю механизм кондиционирования выполняет измерения параметров потока (по алгоритмам: "дырявого ведра" , GCRA и т.д.).

- *Формирование трафика*. С помощью данной функции стремятся сгладить пульсации трафика, чтобы пакеты выходили из устройства более равномерно, чем входили. Сглаживание пульсаций уменьшает очереди в сетевых устройствах, которые будут обрабатывать трафик далее по потоку. Его также целесообразно использовать для восстановления временных соотношений трафика приложений, работающих с равномерными потоками, например, голосовых приложений. Механизмы кондиционирования трафика могут поддерживаться каждым узлом сети или быть реализованы только в пограничных устройствах.

**Протоколы сигнализации QoS** нужны для того, чтобы механизмы QoS отдельных узлов могли обмениваться служебной информацией, способствующей координации усилий по обеспечению параметров качества обслуживания на всем пути следования потока, то есть "из-конца-в-конец". Например, с помощью средств сигнализации приложение может зарезервировать себе вдоль всего маршрута следования требуемую среднюю пропускную способность (для сетей IP это протокол RSVP).

Одно средств сигнализации - маркировка пакета признаком, несущим информацию о требуемом для пакета качестве обслуживания. Наиболее часто в этих целях используется поле приоритета (в пакете IPv4 - первые три бита поля *Type Of Service,* TOS). Продвигаясь от устройства к устройству, пакет переносит вдоль пути следования свои требования к качеству обелуживания, правда, в достаточно обобщенной форме - так как поле приоритета имеет всего несколько возможных значений, то и качество об служивания будет предоставляться дифференцированно по нескольким агрегированным потокам сети.

Инициировать работу протокола сигнализации может и конечный узел и промежуточное устройство.

**Централизованные функции политики, управления и учета QoS** не являются необходимым элементом архитектуры службы QoS, но очень желательны в крупных сетях. Это средства, с помощью которых администратор может задавать рациональный уровень качества обслуживания для отдельных пользователей и приложений или же для их групп. Функции политики позволяют создавать правила, по которым сетевые устройства могут, на основании набора признаков, распознавать отдельные типы трафика и применять к ним определенные функции QoS. Возможны два типа архитектуры QoS:

1. Правила политики мотут конфигурироваться и храниться отдельно в каждом сетевом устройстве, но это требует значительных усилий, порождает большое количество ошибок и приводит к несогласованной работе сетевых устройств.

2. Единые правила политики для всех устройств сети хранятся на сервере политики (или нескольких серверах, реплицирующих базу данных политики). Администратор конфигурирует правила политики в одной точке, что снижает затраты его труда и количество ошибок. Затем с помощью специального протокола эти правила распространяются по всем сетевым устройствам, поддерживающим качество обслуживания, а сетевые устройства применяют политику для кондиционирования трафика и управления очередями в соответствии с нужными параметрами.

Службы QoS, в которых работают централизованные системы поддержки политики, называются службами QoS, основанными на политике. Правила политики полезны не только для управления QoS, но и для координации сетевых устройств при выполнении других функций, например, функций защиты трафика. Поэтому централизованная система политики сети обычно базируется на общей справочной службе сети *(Directory Service),* традиционно хранящей все учетные данные о пользователях, а также самые разнообразные данные о сети: данные о политике QoS, политике безопасности и т. п.

Описанной модели службы QoS соответствует большинство конкретных протоколов поддержки QoS, таких как RSVP, DiffServ сетей TCP/IP, протоколов служб CBR, VBR и ABR сетей ATM.

### 8.2. Алгоритмы управления очередями QoS

Основу средств поддержки QoS в сетевых элементах составляют очереди и алгоритмы обработки этих очередей. Эти механизмы используются в любом сетевом устройстве, которое работает на основе механизма коммутации пакетов - маршрутизатор, коммутатор локальной или глобальной сети, конечный узел. Очередь нужна для обработки периодов временных перегрузок, когда сетевое устройство не может передать пакеты на выходной интерфейс в том темпе, в котором они поступают для выполнения такого продвижения. Если

причиной перегрузки является процессорный блок сетевого устройства, то для хранения необработанных пакетов используется входная очередь. В том же случае, когда причина перегрузки заключается в ограниченной скорости выходного интерфейса, пакеты хранятся в выходной очереди.

Главным фактором по степени влияния на возникновение очередей является коэффициент нагрузки устройства - отношение средней интенсивности входного трафика устройства к средней интенсивности продвижения пакетов на выходной интерфейс.

Если коэффициент нагрузки больше единицы, значит, интенсивность входого трафика постоянно выше, чем интенсивность продвижения пакетов на выходной интерфейс. Поэтому очередь в устройстве существует всегда, и ее длина стремилась бы к бесконечности, если бы не конечный размер буфера, отведенного под хранение пакетов, стоящих в очереди. Но и в том случае, когда коэффициент нагрузки меньше единицы, очередь тоже может существовать, более того - иметь достаточно значительную среднюю длину.

Это происходит тогда, когда имеется некоторая вариация интервалов поступления пакетов в устройство - чем больше эта вариация, тем больше средняя длина очереди. Вариация интервалов поступления пакетов является вторым по величине фактором, влияющим на поведение очередей, после коэффициента нагрузки. При пульсирующем характере многих типов трафика компьютерных сетей, когда коэффициент пульсаций равен 100:1 или более, очереди могут быть значительными. Если же эта вариация отсутствует, то есть пакеты прибывают строго через определенные промежутки времени, как это происходит у трафика типа равномерного потока, то очередь при коэффициенте загрузки, меньшем 1, не возникает. Влияние пульсаций трафика на появление задержек обслуживания хорошо известно пользователям разделяемых сегментов Ethernet. Даже при значениях коэффициента нагрузки сегмента 0,5 задержки доступа к сети бывают значительными, что заставляет использовать эти сети с коэффициентом нагрузки сегмента не более 0,3.

Последствием возникновения очередей является ухудшение качества обслуживания трафика. Образуются задержки передачи пакетов, носящие к тому же непостоянный характер, а это значит, что растут вариации задержек. Кроме того, при длительных пульсациях очереди могут возрастать настолько, что пакеты не помещаются в буферную память сетевых устройств и теряются.

Оценка возможной длины очередей в сетевых устройствах была бы очень полезной, так как помогла бы оценить параметры качества обслуживания при известных характеристиках трафика. Однако поведение очередей представляет собой вероятностный процесс, на который влияет много факторов, особенно при сложных алгоритмах обработки очередей, использующих приоритеты или взвешенное обслуживание разных потоков. Хотя для анализа очередей и разработана спеииальная область прикладной математики - теория массового обслуживания - она может дать количественные оценки только для очень простых ситуаций, не соответствующих реальным условиям работы сетевых устройств. Поэтому служба QoS использует для поддержки гарантированного уровня QoS достаточно сложную модель, решающую задачу комплексно. Это делается с помощью следующих методов:

- за счет предварительного резервирования полосы пропускания для трафика с известными параметрами (например, значениями средней интенсивности и величины пульсации);
- принудительного профилирования входного трафика, что поддерживает коэффициент нагрузки устройства на нужном уровне;
- использования сложных алгоритмов управления очередями.

Чаще всего в маршрутизаторах и коммутаторах применяются следующие алгоритмы обработки очередей:

- традиционный алгоритм FIFO;
- приоритетное обслуживание *(Priority Queuing),* которое также называют "подавляющим";
- настраиваемые очереди *(Custom Queuing);*
- взвешенное справедливое обслуживание ( *Weighted Fair Queuing*, WFQ).

Каждый алгоритм разрабатывался для решения определенных задач и специфическим образом воздействует на качество обслуживания различных типов трафика в сети. Возможно и комбинированное применение этих алгоритмов.

**Традиционный алгоритм FIF0**

Принцип алгоритма в том, что в случае перегрузки пакеты помещаются в очередь, а при исчезновении перегрузки передаются на выход в том порядке, в котором поступили. Во всех устройствах с коммутацией пакетов это алгоритм обработки очередей по умолчанию. Его достоинством является простота реализации и отсутствие потребности в конфигурировании. Однако он имеет и коренной недостаток - невозможность дифференцированной обработки пакетов различных потоков. Все пакеты стоят в общей очереди на равных основаниях - и пакеты чувствительного к задержкам голосового трафика, и пакеты нечувствительного к задержкам, но очень интенсивного трафика резервного копирования, длительные пульсации которого могут надолго задержать голосовой пакет.

Очереди FIFO необходимы для нормальной работы сетевых устройств, но они недостаточны для поддержки дифференцированного качества обслуживания.

**Приоритетное обслуживание**

Алгоритмы приоритетной обработки очередей популярны в операционных системах. Применяются эти алгоритмы и для обеспечения преимущественной обработки одного класса трафика по сравнению с другими. Механизм приоритетной обработки трафика основан на разделении всего сетевого трафика на небольшое количество классов, а затем назначении каждому классу некоторого признака - приоритета. Разделение на классы (классификация) может производиться разнообразными способами.

Выбранный способ классификации не связан непосредственно с работой алгоритма обслуживания на основе приоритетов. Классификация трафика представляет собой отдельную задачу. Пакеты могут разбиваться на приоритетные классы в соответствии:

- с типом сетевого протокола - например, IP, IPX (способ подходит только для устройств, работающих на втором уровне),
- на основании адресов назначения и источника,
- номера TCP/UDP порта,
- любых других комбинаций признаков, которые содержатся в пакетах.

Правила классификации пакетов на приоритетные классы представляют собой часть политики управления сетью. Точка классификации трафика может размещаться как в самом устройстве, как это показано на рисунке 8.2., так и вне его. Более масштабируемое решение - размещение функций классификации трафика в одном или нескольких устройствах, расположенных на границе сети (например, в коммутаторах корпоративной сети, к которым подключаются компьютеры пользователей, или во входных маршрутизаторах сети провайдера).

Этот вариант классификации требует наличия в пакете специального поля. В поле можно запомнить назначенное значение приоритета, чтобы им могли воспользоваться остальные сетевые устройства, обрабатывающие трафик после классифицирующего устройства. Такое поле имеется в заголовке многих протоколов. Так, в пакете IP для этой цели есть трехбитное подполе IP Precedence в поле TOS. В тех же случаях, когда специального поля приоритета в заголовке нет, разрабатывается дополнителъный протокол, который вводит новый заголовок с таким полем. Так произошло с протоколом Ethernet - для него (и других протоколов семейства 802) пришлось разработать спецификации IEEE 802.1Q/p, которые вводят дополнительное трехбитное поле приоритета.

Приоритеты могут назначаться не только коммутатором или маршрутизатором, но и приложением в узле-отправителе. Необходимо также учитывать, что каждое сетевое устройство может не согласиться с назначенным в другой точке сети приоритетом данному пакету. В этом случае оно перепишет значение приоритета в соответствии с локальной политикой, хранящейся непосредственно в данном устройстве. Поэтому необходимо использовать цент-

рализованные способы применения политики в сети, обеспечивающие скоординированную работу устройств.

Независимо от выбранного способа классификации трафика, в сетевом устройстве имеется несколько очередей в соответствии с количеством приоритетных классов. Поступивший пакет в периоды перегрузок помещается в очередь, соответствующую его приоритетному классу. Приоритеты очередей имеют абсолютный характер предпочтения при обработке - так, пока из более приоритетной очереди не будут выбраны все имеющиеся в ней пакеты, устройство не переходит к обработке следующей, менее приоритетной очереди.

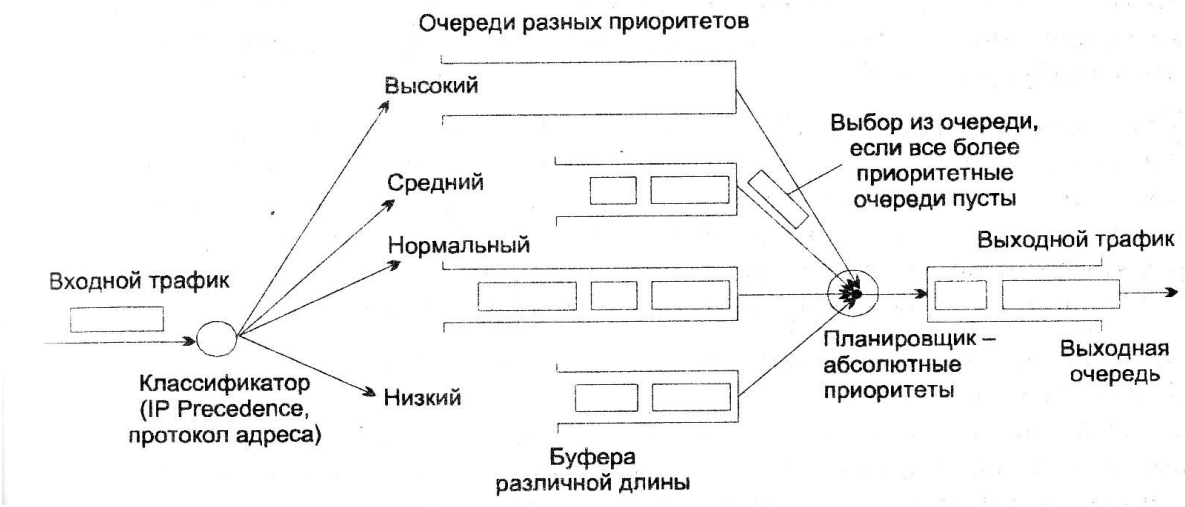


***Рис. 8.2.*** *Приоритетное управление очередями*

Конечный размер буферной памяти сетевого устройства требует, чтобы очереди ожидающих обслуживание пакетов имели некоторую предельную длину. Обычно по умолчанию всем приоритетным очередям отводятся одинаковые буферы, но многие устройства разрешают администратору назначать каждой очереди буфер индивидуального размера.

В общем случае, чем выше значимость трафика для предприятия, чем больше его интенсивность и пульсации, тем больший размер буфера требуется этому трафику. На примере, приведенном на рис. 8.2, для трафика высшего и нормального приоритета выбраны большие значения буферов, а для остальных двух классов – меньшие. Мотивы принятого решения для высшего приоритета очевидны, а трафик нормального приоритета имеет высокую интенсивность и значительный коэффициент пульсаций.

Приоритетное обслуживание очередей обеспечивает высокое качество обслуживания для пакетов из самой приоритетной очереди. Если средняя интенсивность их поступления в устройство не превосходит пропускной способности выходного интерфейса (и производительности внутренних продвигающих блоков самого устройства), то пакеты высшего приоритета всегда получают ту пропускную способность, которая им нужна. Уровень задержек высокоприоритетных пакетов также минимален. Однако он не нулевой, и зависит в основном от характеристик потока этих пакетов – чем выше пульсации потока и его интенсивность, тем выше уровень задержек. Трафик всех остальных приоритетных классов почти прозрачен для пакетов высшего приоритета. Слово "почти" относится к ситуации, когда высокоприоритетный пакет должен ждать завершения обслуживания низкоприоритетного пакета, так как его приход совпал по времени с началом продвижения низкоприоритетного пакета на выходной интерфейс.

Что же касается остальных классов приоритетов, то качество их обслуживания будет ниже, чем у пакетов самого высокого приоритета, причем уровень снижения заранее предсказать весьма трудно. Это снижение может быть довольно существенно, если у высокоприоритетного трафика возникает иногда потребность в передаче данных с большой интенсивностью. Если коэффициент нагрузки выходного интерфейса только трафиком высшего приоритетного класса приближается в какой-то период времени к единице, то трафик остальных классов просто на это время замораживается.

Поэтому приоритетное обслуживание обычно применяется в том случае, когда в сети есть один класс трафика, чувствительный к задержкам, но его интенсивность невелика, так что его обслуживание не слишком ущемляет обслуживание остального трафика. Например, голосовой трафик чувствителен к задержкам, но его интенсивность обычно не превышает 8-16 Кбит/с, так что при назначении ему высшего приоритета остальные классы трафика не будут страдать. Однако в сети могут наблюдаться и другие ситуации, например, при существовании видеотрафика, также требующего приоритетного обслуживания, но имеющего гораздо более высокую интенсивность. Для таких

случаев разработаны алгоритмы обслуживания очередей, дающие низкоприоритетному трафику некоторые гарантии, даже в периоды повышiения интенсивности высокоприоритетного трафика.

**Взвешенные настраиваемые очереди**

Алгоритм взвешенных очередей разработан для того, чтобы можно было предоставить всем классам трафика определенный минимум пропускной способности или гарантировать некоторые требования к задержкам. Под весом данного класса понимается процент предоставляемой классу трафика пропускной способности от полной пропускной способности выходного интерфейса. Алгоритм, в котором вес классов трафика может назначаться администратором, называется "настраиваемой очередью" *(Custom Queuing)*. В случае, когда веса назначаются автоматически на основании некоторой адаптивной стратегии, реализуется так называемый алгоритм *"взвешенного справедливого обслуживания"* *(Weighted Fair Queuing,* WFQ).

Как при взвешенном, так и при приоритетном обслуживании, трафик делится на несколько классов, и для каждого класса ведется отдельная очередь пакетов. Но с каждой очередью связывается не ее приоритет, а процент пропускной способности выходного интерфейса, гарантируемый данному классу трафика при перегрузках этого интерфейса. В примере, приведенном на рисунке 8.3, устройство поддерживает 5 очередей для 5 классов трафика.

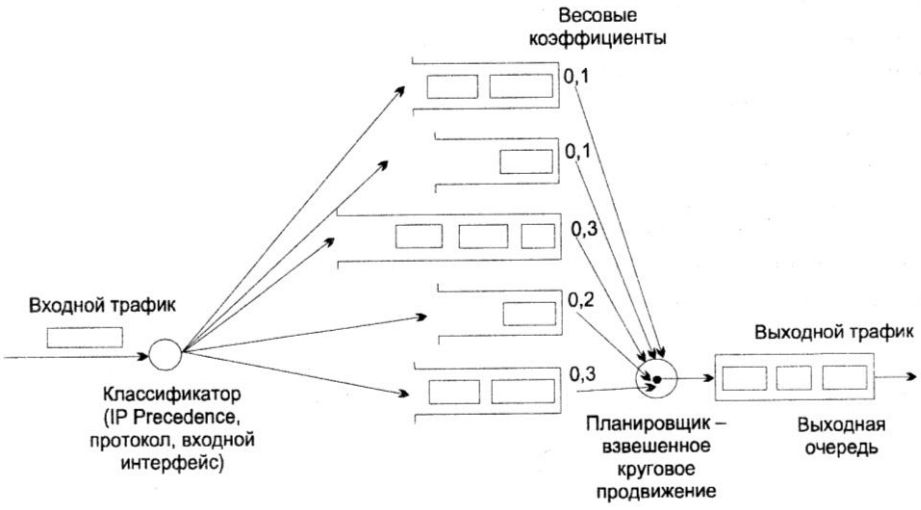


**Рис. 8.3.** Взвешенные настраиваемые очереди

Достигается поставленная цель тем, что очереди обслуживаются последовательно и циклически, и в каждом цикле обслуживания из каждой очереди выбирается такое число байт, которое соответствует весу данной очереди. Например, если цикл просмотра очередей в рассматриваемом примере равен 1 секунде, а скорость выходного интерфейса равна 100 Мбит/с, то при перегрузках в каждом цикле из первой очереди выбирается 10 Мбит данных, из второй тоже 10 Мбит, из третьей - 30 Мбит, из четвертой - 20 Мбит, а из пятой - 30 Мбит. В результате каждому классу трафика достается гарантированный минимум пропускной способности, что во многих случаях является хорошим результатом.

Оценить уровень задержек сложнее, чем уровень пропускной способности. Чтобы он был соизмерим со временами передачи пакетов, цикл работы арбитра очереди должен быть, естественно, меньше, чем 1 секунда - значение, выбранное для иллюстрации метода и упрощения расчетов. При таком времени цикла задержка может составить 1 секунду и больше, так как арбитр возвращается к каждой очереди не чаще, чем раз в секунду, кроме того, в очереди может находиться более одного пакета. Время цикла в 100 мкс более подходит приведенному примеру. С одной стороны, оно обеспечивает обслуживание очереди каждого класса каждые 100 мкс. С другой стороны, этого времени достаточно, чтобы выбрать из каждой очереди в среднем по несколько пакетов (учитывая, что размер пакета в сетях Ethernet колеблется от 64 до 1518 байт).

На уровень задержек и вариации задержек пакетов для некоторого класса трафика при взвешенном обслуживании в значительной степени влияет коэффициент нагрузки трафика данного класса. В этом случае коэффициент подсчитывается как отношение интенсивности входного трафика класса к пропускной способности, выделенной этому классу его весом. Качественное поведение очереди и, соответственно, задержек здесь выглядит примерно так же, как и в случае очереди FIFO

- чем меньше коэффициент нагрузки, тем меньше средняя длина очереди и тем меньше задержки.

Точные значения параметров QoS для алгоритма взвешенного обслуживания предсказать трудно. На них оказывают существенное влияние все динамически изменяющиеся параметры нагрузки сетевого устройства - интенсивность пакетов всех классов и вариации промежутков времени между прибытием пакетов. В общем случае, взвешенное обслуживание приводит к большим задержкам и их вариациям, чем приоритетное обслуживание для самого приоритетного класса, даже при значительном превышении выделенной пропускной способности над интенсивностью входного потока данного класса. Но для более низких приоритетных классов это соотношение может оказаться и несправедливым, поэтому для создания более благоприятных условий обслуживания всех классов трафика взвешенное обслуживание часто оказывается более приемлемым.

Как и для приоритетного обслуживания, администратор может назначать разным классам очередей буферы различных размеров. Уменьшение размеров буферов для очередей приведет к увеличению потерь пакетов при перегрузках, но зато уменьшаются времена ожидания для тех пакетов, которые не были отброшены и попали в очередь.

**Взвешенное справедливое обслуживание (WFQ)**

Взвешенное справедливое обслуживание является комбинированным механизмом обслуживания очередей, сочетающим приоритетное обслуживание со взвешенным. Существует большое количество различных реализаций WFQ производителями сетевого оборудования.

Наиболее распространенная схема предусматривает существование одной особой очереди, которая обслуживается по приоритетной схеме, то есть в первую очередь и до тех пор, пока все заявки из нее не будут выбраны. Эта очередь предназначена дли системных сообщений, сообщений управления сетью и, возможно, пакетов наиболее критических и требовательных при-

ложений. Во всяком случае, предполагается, что ее трафик имеет невысокую интенсивность, так что значительная часть пропускной способности выходного интерфейса остается другим классам трафика.

Остальные очереди маршрутизатор просматривает последовательно, по алгоритму взвешенного обслуживания (рис. 8.4). Администратор может задать вес каждого класса трафика, то есть количество байт из каждой очереди, выбираемых перед переходом к следующей. Возможен и вариант работы по умолчанию, когда всем остальным классам трафика достается равное количество пропускной способности выходного интерфейса, оставшейся от трафика приоритетного класса.

Производители оборудования дополняют механизм WFQ некоторыми полезными режимами работы. Например, в маршрутизаторах компании Cisco сушествует несколько разновидностей WFQ:

- основанный на потоках *(F1ow-based)* режим WFQ (FWFQ);
- основанный на классах *(C1ass-based)* режим WFQ (CWFQ).

Для основанного на потоках варианта FWFQ в маршрутизаторе создается столько очередей, сколько потоков существует в трафике. При применении алгоритма FWFQ под потоком понимаются пакеты, имеющие определенные значения IP-адресов назначения и источника и/или портов TCP/UDP источника и назначения (типа протоколов транспортного уровня), значения поля TOS. То есть поток – это последовательность пакетов от одного приложения с определенными параметрами качества обслуживания, заданными в поле TOS. (Иногда такой поток называют микропотоком - *microflow,* оставляя термин "поток" для агрегированных микропотоков, представляюших, например, совокупность всех микропотоков от приложений одного пользователя или определенной подсети.)

Каждому потоку соответствует отдельная выходная очередь. Во время периодов перегрузок механизм WFQ назначает каждой выходной очереди равные значения пропускной способности порта. Поэтому иногда алгоритм FWFQ называют просто FQ *(Fair Queuing)* - справедливое обслуживание.

Основанный на классах вариант CWFQ в маршрутизаторах Cisco имеет два подварианта:

- классы трафика определяются на основании так называемых QoS-групп, которые соответствуют набору признаков из списка управления доступом (ACL), например, номеру входного интерфейса или номеру хоста или подсети;
- классы трафика определяются значениями полей TOS.

Для варианта QoS-групп администратор задает веса пропускной способности, выделяемой каждой QoS-группе, а также (опционально) максимальную длину очереди.


**Рис. 8.4.** Взвешенное справедливое обслуживание

Для очередей, основанных на классах QoS, пакеты, которые не назначены ни в одну группу, принадлежат к группе 0. При назначении весов WFQ нужно принимать во внимание следующее:

- один процент имющейся пропускной способности автоматически назначается QoS-группе 0;
- общий вес всех остальных групп не может превосходить 99%;
- любая остающаяся после назначения весов пропускная способность выделяется QoS-группе 0.

Для варианта, использующего для классификации значение TOS, существуют веса классов по умолчанию. Они назначаются, если администратор явно не задал их с помощью команды *weight*. Для классификации используется 2 младших

бита трехразрядного подполя IP Precedence из поля TOS, так что в этом варианте существует всего 4 класса трафика. По умолчанию, классу 0 выделяется 10% выходной пропускной способности, классу 1-20%, классу 2-30% и классу 3-40%. Чем выше класс, тем выше значимость трафика, поэтому выделение ему по умолчанию большей доли пропускной способности создает для него более привилегированные условия продвижения. Используя команду weight, администратор может изменить значения весов TOS-классов, используемых по умолчанию.

Администратор в маршрутизаторах Cisco должен явно задать режим работы механизма WFQ: fair-queue для поддержки потоков, fair-queue qos-group для поддержки qos-групп или fair-queue tos для поддержки классов TOS.

Во многих сетевых устройствах механизм WFQ является одним из основных для поддержки качества обслуживания, в том числе и по различным протоколам, использующим методы сигнализации для координированного поведения всех устройств сети, например, протокола RSVP.

### 8.3. Механизмы профилирования и формирования трафика

Случайное раннее обнаружение - RED

Техника, названная "случайным ранним обнаружением" *(Random Ear1y Detection,* RED) представляет собой механизм профилирования трафика, разработанный сообществом Internet для предотвращения перегрузок на магистрали Internet. Основным назначением RED является предотвращение серьезных перегрузок сети. RED работает совместно с протоколом надежного транспорта TCP и использует алгоритм реакции TCP на потерю пакетов, когда источник трафика замедляет передачу пакетов в сеть. Это свойство и использует RED как неявную обратную связь для уведомления о том, что источник слишком интенсивно генерирует данные.

В алгоритме RED используются два конфигурируемых порога уровня перегрузки. Когда уровень перегрузки ниже первого порога, пакеты не отбрасываются. Когда уровень перегрузки находится между двумя порогами, пакеты

отбрасываются с линейно возрастаюшей вероятностью из диапазона от 0 до конфигурируемой величины, достигаемой как раз при достижении второго порога. Когда же перегрузка превышает второй порог, пакеты начинают отбрасываться со 100%-й вероятностью.

В качестве показателя перегрузки используется вычисляемое среднее значение длины очереди пакетов, принадлежащей к определенной сессии TCP. Использование усредненного, а не мгновенного значения очереди позволяет отделить кратковременные перегрузки, которые могут быть нормально обработаны устройством и сетью, от длительных перегрузок, которые грозят затопить сеть.

Алгоритм "дырявого ведра"

Алгоритм "дырявого ведра" *(Leaky Bucket)* разработан для профилирования пульсирующего трафика. Алгоритм позволяет проверить соблюдение трафиком оговоренных значений средней скорости и пульсации. У алгоритма имеется несколько настраиваемых значений:

- T - период усреднения скорости;
- CIR *(Committed Information Rate)* - средняя скорость, которую трафик не должен превышать (скорость, согласованная с сетью);
- Bc - объем пульсации, соответствующий средней скорости CIR и периоду T: Bc = CIR x T;
- Be - допустимое превышение объема пульсации.

В алгоритме предполагается, что трафик контролируется каждые T секунд. На каждом из этих интервалов времени трафик должен иметь среднюю скорость не более оговоренной скорости CIR. Скорость контролируется на основе подсчета объема данных, поступивших за период T. Если этот объем меньше или равен Bc, то фактическая скорость трафика была меньше Bc/T, то есть меньше CIR. Превышение объемом пульсации оговоренного значения Bc на величину Be считается "мягким" нарушением - пакеты-нарушители должны быть помечены признаком DE *(Discard Eligibility),* но не отброшены. При превышении объема пульсации величины Bc + Be кадры отбрасываются.

Алгоритм использует счетчик С поступивших от пользователей байт. Каждые Т секунд этот счетчик уменьшается на величину Вс (или же сбрасывается в 0, если значение счетчика меньше, чем Вс), что часто иллюстрируется ведром, из которого дискретно, каждые Т секунд, вытекает объем, равный минимальному из чисел В или С (рис. 8.5.). Все кадры, данные которых не увеличили значение счетчика свыше порога Вс, пропускаются в сеть со значением признака DE = 0. Кадры, данные которых привели к значению счетчика, большему Вс, но меньшему Вс + Ве, также передаются в сеть, но с признаком DE = 1. И, наконец, кадры, которые привели к значению счетчика, большему Вс + Ве, отбрасываются коммутатором.

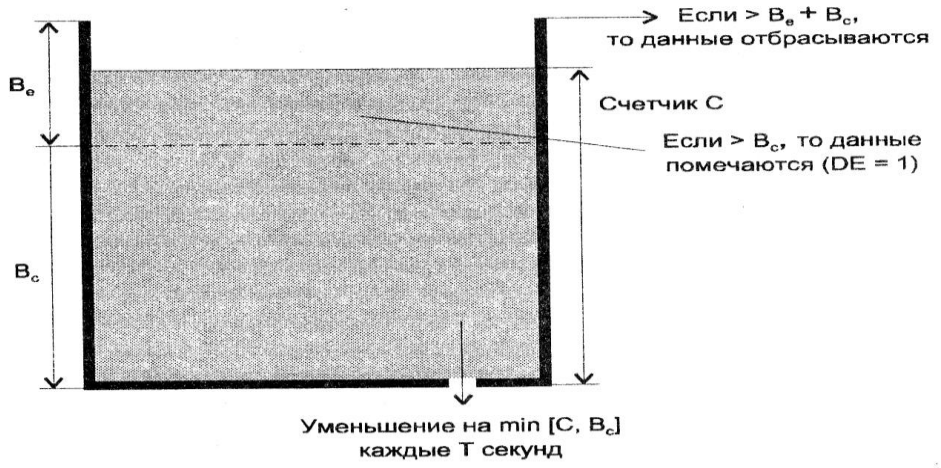


***Рис. 8.5.*** *Алгоритм “дырявого ведра”*

Алгоритм допускает различные модификации, например, использование большего количества порогов объема пульсации, использование вместо объема пульсации значения максимальной скорости и т. п. Одна из модификаций алгоритма "дырявого ведра" под названием *Generic Ce!! Rate Algorithm* (GCRA) применяется в сетях ATM для контроля нескольких параметров: пиковой скорости, средней скорости, вариации интервала прибытия ячеек и объема пульсации.

Алгоритм "ведро токенов"

Этот алгоритм применяется не для профилирования, а для формирования графика. Его цель - уменьшение неравномерности

продвижения пакетов, когда из-за значительной пульсации они сбиваются в плотные группы (рис. 8.6.).

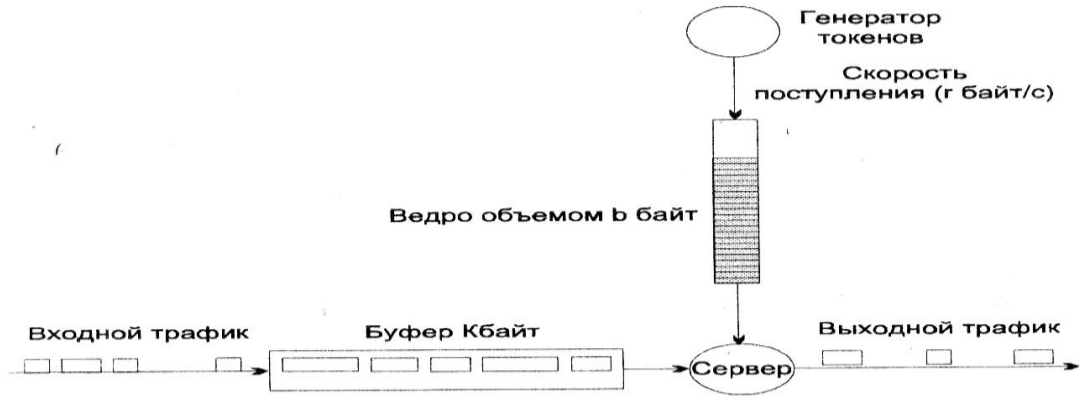


***Рис. 8.6.*** *Алгоритм "ведро токенов"*

Под токеном в данном случае понимается некий абстрактный объект, носитель "порции" информации, используемый для построения модели обслуживания трафика. Генератор токенов периодически направляет очередной токен в "ведро" с ограниченным объемом в b байт. Все токены имеют одинаковый объем m байт, а генерация токенов происходит с такой скоростью, что "ведро" заполняется со скоростью r байт в секунду. Скорость r является желательной средней скоростью для формируемого трафика. Пакеты поступают в систему и попадают в очередь объемом К байт. Из очереди пакет продвигается сервером только в том случае, если к этому моменту "ведро" заполнено токенами до уровня не ниже М байт, где М - объем пакета. Если это условие выполняется и пакет продвигается на выход, то из ведра удаляются токены общим объемом в М байт (с точностью до m байт). Если же ведро заполнено недостаточно, то пакет из очереди не выбирается, ожидая поступления нужного числа токенов. Таким образом, достигается "улучшение" трафика: если в результате пульсации в систему приходит большая пачка пакетов, то из очереди пакеты выходят равномерно, в темпе, задаваемом генератором токенов. Поток токенов представляет собой идеальный трафик, к форме которого стараются привести входной трафик.

## 8.4. Общая характеристика протоколов QoS IP

Протоколы и механизмы поддержки качества обслуживания в сетях IP делятся на две категории в зависимости от уровня гарантий предоставляемого сервиса:

- протоколы поддержки "твердого" качества обслуживания, обеспечивающего гарантированное обслуживание потоков данных на основе резервирования ресурсов;
- протоколы поддержки "мягкого" качества обслуживания, которые не могут дать количественных гарантий предоставляемого обслуживания, но за счет приоритезированной и взвешенной обработки очередей выполняют предпочтительное распределение ресурсов сети между классами трафика.

Протоколы первой категории разрабатываются рабочей группой IETF по интегрированным сервисам - IntServ. Базовая модель такого сервиса предполагает интегрированное взаимодействие всех устройств сети по обеспечению требуемого качества обслуживания вдоль всего пути потока. Сетевые ресурсы распределяются в соответствии с QoS-запросами приложений и подчиняются политике управления полосой пропускания. В целом модель IntServ соответствует обобщенной модели службы QoS, описанной выше. Наиболее детально проработан протокол сигнализации модели RSVP, с помощью которого конечные узлы выполняют резервирование ресурсов.

Однако ввиду значительной сложности поддержки интегрированных сервисов в масштабах такой сети, как Internet, получило развитие и другое направление, которое в данный момент ведет рабочая группа по дифференцированным сервисам - *Differentiated Services,* DiffServ. Эта группа занимается протоколами второй категории, которые не обеспечивают "настоящего" гарантированного качества обслуживания, но зато гораздо проще в реализации.

Сервисы DiffServ опираются на ту же обобщенную модель QoS, что и сервисы IntServ, однако не прибегают к резервированию ресурсов сетевых устройств. Вместо этого они пользуются сигнализацией потребностей потоков в каждом отдельном пакете - поле IP Precedence переносит код, который

интерпретируется каждым маршрутизатором сети для приоритетного или взвешенного обслуживания данного потока по отношению к другим.

Дифференцированные сервисы пока отстают от интегрированных по степени зрелости протоколов. Интегрированные сервисы применяются в основном к отдельным потокам приложений (микропотокам), а дифференцированные - к небольшому числу агрегированных потоков. Поэтому реализация дифференцированных сервисов связана с гораздо более низким уровнем накладных расходов в маршрутизаторах. Маршрутизаторам не требуется запоминать большое количество переменных состояния каждого микропотока сети, а нужно только помнить параметры политики обработки для небольшого количества агрегированных потоков и применять их независимо к каждому пакету. Этот подход близок к традиционному стилю работы маршрутизаторов, когда каждый пакет обрабатывается независимо от состояния сети и конечных узлов.

Областью преимущественного применения дифференцированных сервисов специалисты видят магистрали сетей. Это связано с тем, что через магистраль обычно проходит настолько большое количество микропотоков различных приложений, что гарантированная поддержка каждого из них в соответствии с принципами интегрированного обслуживания требует чрезмерно больших накладных расходов в маршрутизаторах магистрали. Применение здесь дифференцированных сервисов гораздо экономичней, так как они работают с агрегированными потоками, образованными из микропотоков со схожими требованиями к QoS.

Кроме протоколов категорий IntServ и DiffServ, в IP-сетях на сегодня существует еще несколько протоколов, связанных с поддержкой качества обслуживания. Прежде всего, это рассмотренный выше протокол MPLS, который позволяет направлять поток по маршруту, обеспечивающему необходимое качество обслуживания.

Имеется также протокол согласованной поддержки качества обслуживания в подсетях IP, называемый *Subnet Bandwidth Management* (SBN). Он осуществляет связь средств поддержки QoS на канальном уровне для разделяемых и

коммутируемых сетей IEEE 802 (спецификаций 802.1Q/p) с параметрами качества обслуживания на уровне IP.

К средствам поддержки QoS в IP-сетях относятся также механизмы обслуживания и формирования очередей, такие как WFQ, CQ, RED, работающие в отдельных узлах. Эти механизмы работают в маршрутизаторах как необходимые элементы для реализации интегрированных и дифференцироианных сервисов. Однако администратор сети может их использовать и для улучшения качества обслуживания в тех маршрутизаторах, где возникают заторы трафика, существенно влияющие на работу сети в целом. В таблице 8.1 приведено сравнение протоколов QoS для IP-сетей по следующим параметрам:

- поддерживаемый уровень QoS - высший или низший;
- место, где применяется сервис и управление - в приложении (App) или в маршрутизаторах сети (N et).

Из таблицы видно, что существуют различные варианты применения одного и того же протокола, отличающиеся точкой применения (в узле, на входе в определенную часть сети и т. п.), а также его согласованностью с другим протоколом. Например, дифференцированные сервисы будут обеспечивать более высокий уровень QoS, если их работа в точке входа в магистраль будет согласована с требованиями по резервированию ресурсов потоков, втекающих в магистраль и обслуживающихся на периферии сети протоколом RSVP.

**Резервирование пропускной способности с помощью RSVP**

*ReSerVation Protocol* (RSVP) - это протокол сигнализации, который обеспечивает резервирование ресурсов и управление ими с целью предоставления интегрированных сервисов, предназначенных ддя эмуляции выделенных каналов в IP-сетях. Протокол RSVP представляет собой наиболее сложную QoS-технологию, как для приложений (хостов), так и для сетевых элементов (маршрутизаторов и коммутаторов). Он в наибольшей степени отличается от IP-сервиса "с максимальными усилиями" и обеспечивает на сегодня наивысший уровень качества обслуживания в терминах гарантий сервиса, диф-

ференцированного распределения ресурсов и уровня детализации обратной связи для QoS-приложений и пользователей.

**Таблица 8.1.**

Сравнение протоколов QoS

| Уровень QoS | Сеть | Приложение | Описание |
|---|---|---|---|
| Высший | X | | Избыточное выделение ресурсов из конца в конец (например, частная сеть с низкой загрузкой) |
| | X | X | Гарантированный сервис RSVP (обеспечивает обратную связь с приложени ем |
| | X | X | Дифференцированные сервисы (DiffServ), примененные на входе в магистраль сети и соответствующие уровню сервиса резервирования RSVP для этого потока |
| | X | X | Приоритезация, использующая SBM |
| | X | | DiffServ или SBM, примененные к отдельному потоку приложением-источником |
| | X | | DiffServ, примененный на входе в магистраль сети |
| | | | Алгоритмы обслуживания очередей, применяемые сетевыми элементами, например, WFQ, CQ, RED |
| Низший | | | Сервис "с максимальными усилиями" |

Протокол RSVP разрабатывался в расчете на то, что в Intenet будет широко использоваться групповое вещание, при котором у одного источника существует несколько приемников информации. При этом приемники информации могут отличаться своими возможностями, так что и потребности в пропускной способности соединения с источником у них могут быть разные. Для учета разнородности нескольких приемников в протоколе RSVP параметры резервирования определяет не источник, а приемник информации. Инициатором работы протокола остается источник, но он только рассылает всем приемникам уведомление о том, что вскоре начнет передавать данные и приемникам необходимо зарезервировать ресурсы сети для качественного приема этих данных. В частном случае, когда приемник один, то есть когда в пакете указан не групповой, а уникальный адрес назначения, работа протокола RSVP не изменяется. Резервирование необходимых ресурсов сети выполнятся с помощью такой последовательности шагов (рис. 8.7.):

1. Источник данных посылает по уникальному или групповому адресам получателя специальное сообщение *Pass,* в котором он указывает рекомендуемые параметры для качественного приема своего трафика: верхней и нижней границ пропускной способности, задержки и вариации задержки. Эти параметры трафика содержатся в так называемой *спецификации трафика (TSpec).* Сообщение *Pass* передается маршрутизаторами сети в направлении к источнику (или источникам) с использованием таблиц маршрутизации, полученных с помощью любого протокола маршрутизации.

2. Каждый поддерживающий RSVP маршрутизатор, получив сообщение *Pass,* фиксирует "состояние пути", которое включает предыдущий адрес источника сообщения *Pass,* то есть следующий шаг вверх по пути, ведущему к источнику. Тем самым в сети образуется фиксированный маршрут передачи сообщений в рамках сессии RSVP. Под сессией понимается поток пакетов, имеющих одинаковые значения IP-адреса назначения, идентификатора транспортного протокола, указанного в поле *Protocol,* а также номера порта UDP/TCP.

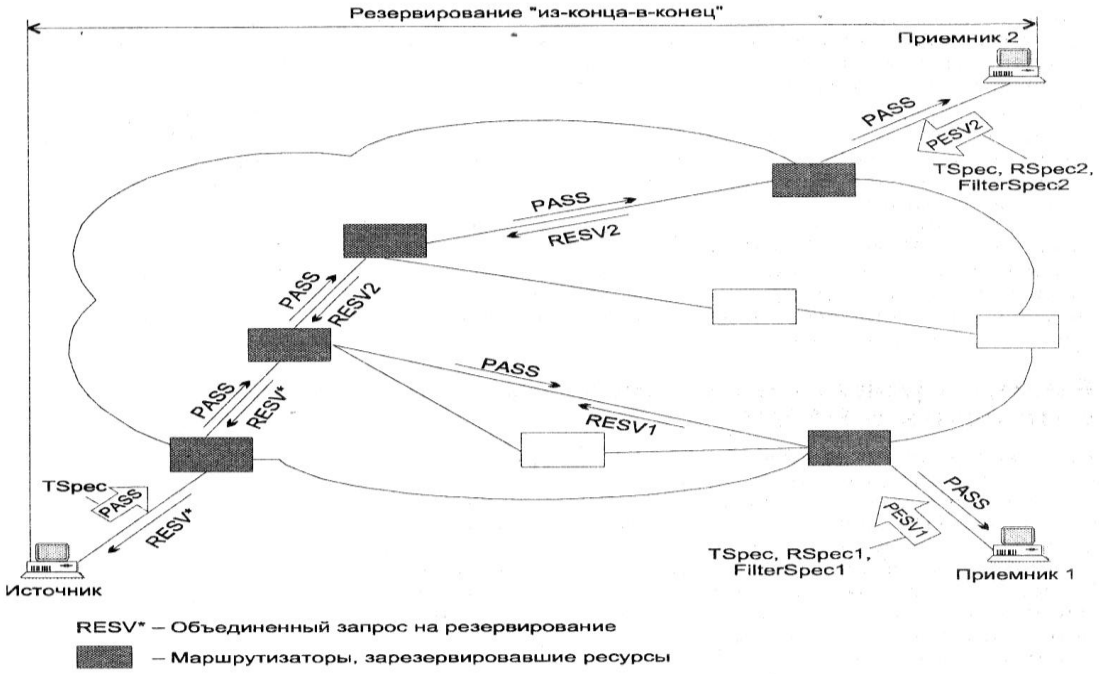


***Рис. 8.7.*** *Резервирование ресурсов по протоколу RSVP*

Адрес источника не фигурирует в определении потока, что позволяет в рамках сессии передавать потоки данных к определенному приложению определенного узла от разных узлов и приложений. При использовании группового адреса номер порта не используется, так как получающее данные приложение определяется отдельным групповым адресом. (Для идентификации потоков в протоколе IPv6 введено такое понятие, как "метка потока" *(flow 1abe1).* Метка потока - это признак, размещаемый в основном заголовке IP-пакета и указывающий принадлежность данного пакета к последовательности пакетов - потоку, для которого требуется обеспечить определенные параметры обслуживания, отличные от принятых по умолчанию.)

3. После получения сообщения *Pass* приемник отправляет маршрутизатору, от которого он получил это сообщение, запрос резервирования ресурсов - сообщение *Resv (reservation reguest).* 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

(например, по типу транспортного протокола и номеру порта). Вместе *RSpec и filterspec* представляют собой *дескриптор потока,* который маршрутизатор использует для идентификации каждого резервирования ресурсов. Фильтр может определить поток пакетов с любой степенью детализации и на основании любой информации, в том числе и прикладного уровня. Например, спецификация фильтра может быть использована для выбора различных микропотоков в иерархически кодированном потоке видеоданных за счет полей в заголовке прикладного уровня. Запрашиваемые параметры QoS в спецификации *RSpec* могут отличаться от указанных в спецификации *TSpec.* Например, если приемник решил принимать не все посылаемые источником пакеты, а только их часть (что нужно задать с помощью спецификации фильтра), то ему нужна, соответственно, меньшая пропускная способность.

4. Когда каждый поддерживающий RSVP маршрутизатор вдоль восходящего пути получает сообщения *Resv,* то он использует два процесса, с помощью которых определяет приемлемость указанных в запросе параметров резервирования. С помощью процесса управления доступом *(admission control)* проверяется, имеются ли у маршрутизатора ресурсы, необходимые для поддержания запрашиваемого уровня QoS, а с помощью процесса управления политикой *(policy control)* - имеет ли пользователь право на резервирование ресурсов. Если запрос не может быть удовлетворен (из-за недостатка ресурсов или ошибки авторизации), маршрутизатор возвращает сообщение об ошибке отправителю. Если запрос принимается, то маршрутизатор посылает сообщение *Resv* вверх по пути следующему маршрутизатору, а данные о требуемом уровне QoS передаются механизмам маршрутизатора, ответственным за управлением трафиком. Протокол RSVP не описывает процессы управления доступом и политики - это составляет предмет отдельных спецификаций (процесс управления доступом описан и общей спецификации интегрированных сервисов, а детали процесса управления политикой находятся еще в стадии разработки).

5. Прием запроса резервирования маршрутизатором означает также передачу параметров QoS на отработку в соответствующие блоки маршрутизатора. Конкретный способ отработки параметров QoS маршрутизатором в протоколе RSVP не описывается. В том случае, когда технология второго уровня, обслуживающая выходной интерфейс, не поддерживает управление качеством обслуживания, то отработка выполняется механизмом управления очередями, таким как WFQ или CQ. Если же эта технология поддерживает QoS, то параметры спецификации *RSpec* отображаются на параметры QoS данной технологии, например ATM.

6. Когда последний маршрутизатор получает сообщение *Resv* и принимает запрос, то он посылает подтверждающее 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. Если для всех резервируемых потоков запрашивается одинаковая пропускная способность, то она требуется и для общего потока. Если запрашиваются различные величины пропускной способности, то для общего потока выбирается максимальная.

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

8. Резервирование можно отменить прямо или косвенно. Прямая отмена выполняется по инициативе источника или приемника с помощью соответствующих сообщений протокола RSVP. Неявная отмена происходит по тайм-ауту: состояние резервирования имеет срок жизни, как, например, и динамические записи в таблицах маршрутизации, и приемник по протоколу RSVP должен периодически подтверждать резервирование. Если же сообщения подтверждения перестают поступать, то резервирование отменяется по истечении его срока жизни.

Сообщения PATH и RESV передаются маршрутизаторами, не поддерживающими RSVP, прозрачно. Следовательно:

- RSVP - это не транспортный, а управляющий протокол. Он не переносит данные, а работает параллельно с потоками данных TCP или UDP.
- Резервирование в каждом маршрутизаторе является 'мягким", что означает, что приемник должен периодически его обновлять.
- Приложениям требуются API для задания требований к потоку, инициирования требований на резервирование и получение уведомлений об успехе или неудаче резервирования во время начального запроса или в течение сессии.
- Резервирование основано на приемнике для того, чтобы эффективно поддерживать большие мультикастовые гетерогенные группы приемников.
- Групповое резервирование "сливается" в точках репликации трафика на восходящем пути, что требует разработки сложных алгоритмов, которые еще не очень хорошо поняты.
- RSVP-трафик может проходить через не поддерживающие RSVP маршрутизаторы. Это создает слабые звенья в цепи QoS, в которых уровень обслуживания снижен до уровня обслуживания "с максимальными усилиями".

В заключение отметим, что существует две разновидности протокола RSVP. Native RSVP имеет номер протокола 46 и инкапсулируется непосредственно в пакет IP. Вторая разновидность - RSVP, основанный на UUP. Механизм SBM использует только Native RSVP. Как уже было упомянуто, протокол RSVP реализует высший уровень QoS, имеющийся для RSVP. Поэтому возникает вопрос, зачем нужны другие протоколы, если он такой хороший? Причина в том, что за это качество приходится расплачиваться сложностями и большими накладными расходами, что неприемлемо для многих приложений и некоторых областей сети. Отсюда появляется потребность в более простых, хотя и менее качественных методах, одним из которых является DiffServ.

**Дифференцированное обслуживание DiffServ**

Дифференцированные сервисы представляют собой простой и весьма грубый метод классификации требований к качеству обслуживания небольшого

числа агрегированных потоков сети. Основная идея, положенная в основу этой группы сервисов QoS, состоит в том, чтобы все маршрутизаторы сети единообразно понимали закодированные в 5 битах поля TOS протокола IPv4 (или поля Traffic C1ass протокола IPv6) требования к качеству обслуживания. Дифференцированные сервисы не используют резервирования ресурсов в маршрутизаторах, поэтому не могут дать гарантий на предоставляемое качество обслуживания, реализуя только "мягкую" поддержку QoS. Каждый маршрутизатор работает независимо от остальных и старается обеспечить предпочтительное качество обслуживания потоков в соответствии с правилами пошагового поведения *(Per-Hop Behavior*, PHB), которые должны определяться отдельными стандартами IETF. Пять бит, отводимые для кодирования требований к качеству обслуживания, определяют, что в сети могут существовать не более 32 агрегированных потоков, отличающихся предоставляемым качеством обслуживания.

Обобщенно поле, переносящее коды качества обслуживания в протоколах IP разных версий, называется DS-байтом. Упомянутые ранее 5 бит составляют старшую часть этого поля, а оставшиеся 3 бита пока не используются. Для протокола IPv4 DS-байт является байтом TOS, а для IPv6 – байтом Traffic C1ass. Как показано на рисунке 8.7, хотя поле DS использует поле TOS, оно не полностью сохраняет значения бит этого поля, как это было определено ранее в соответствующихх RFC (791, 1122, 1349). Однако значения старших трех бит IP Precedence (0-2) сохраняются - и в поле DS-байт эти три бита по умолчанию определяют приоритетность трафика, причем значению 111 этих бит соответствует самый приоритетный класс трафика, а значению 000 - самый низкоприоритетный (RFC 2474). Оставшиеся два бита поля DS-байт определяют предпочтительность отбрасывания пакетов при перегрузках.

В целом, модель дифференцированных сервисов соответствует обобщенной модели QoS, если считать, что перенос поля DS-байт через сеть является протоколом сигнализации. Маршрутизатор, поддерживающий дифференцированные сервисы и процессы классификации, маркирования, измерения rt кондиционирования трафика.

Хоти маркировкой пакетов может заниматься каждый маршрутизатор сети, модель дифференцированных сервисов считает основным вариантом маркировку пакетов во входной точке сети, поддерживающей протокол DiffServ и находящейся под административным контролем одной организации. Такая сеть называется доменом DiffServ. При выходе пакетов из домена DiffServ маркировка снимается, так что другой домен может назначить ее заново.

Протокол DiffServ подразумевает существование соглашения о качестве обслуживания (SLA) между доменами с обшей границей. Соглашение SLA устанавливает критерии политики и определяет профиль трафика. Ожидается, что трафик будет формироваться и сглаживаться в выходных точках домена в соответствии с SLA, а во входной точке домена кондиционироваться в соответствии с правилами политики. Любой трафик "вне профиля" (то есть выходящий за верхние границы полосы пропускания, указанные в SLA) не получает гарантий обслуживания (или же оплачивается повышенной стоимостью в соответствии со SLA). Критерии политики могут включать время дня, адреса источника и назначения, транспортный протокол, номера портов. В общем случае любое содержимое пакета может использоваться для применения политики.

На сегодняшний день разработаны два стандарта пошагового продвижения пакетов PHB, которые представляют два различных сервиса:

- "быстрое продвижение" *(Expedited Forwarding,* EF). Характеризуется одним значением кода (10111) и представляет собой высший уровень качества обслуживания, минимизирующий задержки и вариации задержек. Любой трафик, интенсивность которого превышает профиль, отбрасывается;
- "гарантированная доставка" *(Assured Forwarding,* AF). Имеются четыре класса трафика и три уровня отбрасывания пакетов в каждом классе, всего 12 различных типов трафика. Каждому классу трафика выделяется определенный минимум пропускной способности и размер буфера для хранения его очереди. Трафик, который превышает профиль, доставляется с меньшей степенью

вероятности, чем трафик, удовлетворяющий условию профиля. Это означает, что он может быть понижен в качестве, но не обязательно отброшен.

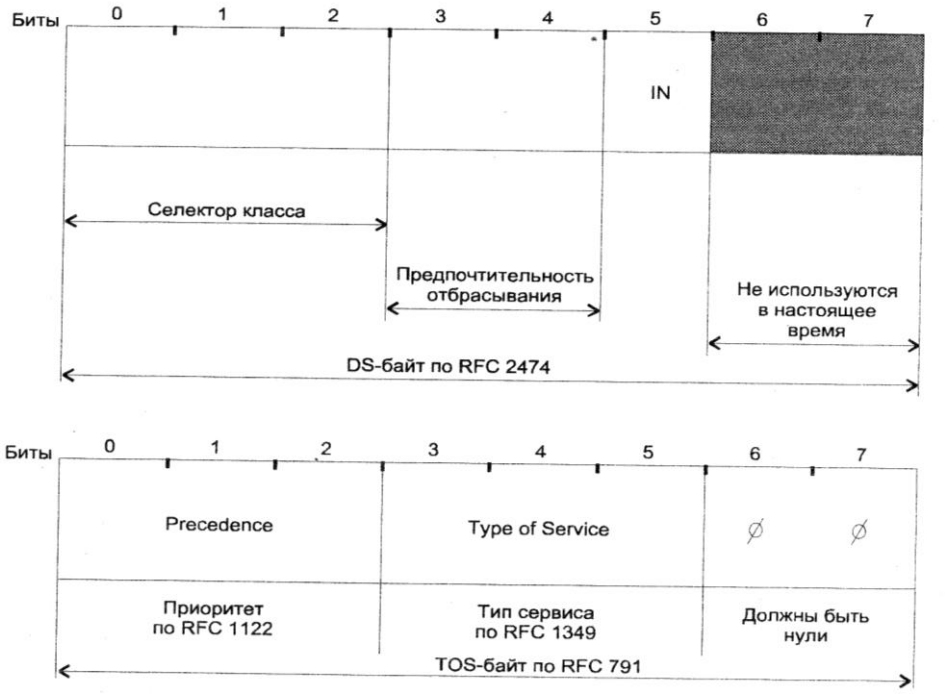


***Рис. 8.7.*** *Соответствие полей байта DS полям байта TOS*

Основное назначение сервиса EF - предоставление качества обслуживания, сопоставимого с качеством выделенных каналов. Поэтому этот сервис называется также сервисом "виртуальных выделенных каналов". Кроме того, он известен под названием Premium Service, что подчеркивает высшее качество обслуживания в IP-сетях с дифференцированным обслуживанием, которое он предоставляет.

Если сервис EF реализуется механизмом, который позволяет неограниченное вытеснение другого трафика (например, приоритетной очередью), то реализация должна включать некоторые средства ограничения влияния трафика EF на другие классы трафика, например, с помощью ограничения скорости EF-трафика с помощью алгоритма "ведра токенов". Максимальная скорость трафика EF и, возможно, размер пульсации должны устанавливаться сетевым администратором. Спецификация RFC 2598 не оговаривает точного механизма обслуживания

очереди, который должен работать в узле для реализации этого сервиса. Наряду с приоритетным обслуживанием им может быть и механизм взвешенного обслуживания, например WFQ, сконфигурированный в узле так, чтобы трафику EF предоставлялся минимум пропускной способности, превышающий максимальную пропускную способность трафика, оговоренную в профиле.

Сервис AF описан в спецификации RFC 2597, его четыре группы ориентированы на гарантированную доставку, но без минимизации уровня задержек пакетов, как это оговорено для более качественного сервиса EF. Гарантированная доставка выполняется в том случае, когда входная скорость трафика не превышает отведенной данному классу минимальной пропускной способности. Классы сервиса AH хорошо сочетаются с одним классом сервиса EF - трафик EF может обслуживаться по приоритетной схеме, но с ограничением интенсивности входного потока. Оставшаяся пропускная способность распределяется между классами трафика AF в соответствии с алгоритмом взвешенного обслуживания, который обеспечит необходимую пропускную способность, но не минимизацию задержек.

Простота приоритезации трафика с помощью DiffServ определяет его гибкость и мощь. Когда DiffServ использует параметры RSVP или специфические типы приложений для идентификации и классификации трафика CBR, то возможно организовать агрегированные потоки, направленные в каналы фиксированной пропускной способности. В результате можно эффективно разделять ресурсы и одновременно поддерживать гарантированный сервис.

**Виртуальные каналы MPLS**

Технология MPLS *(Mu1ti-Protocol Label Switching)* подобна в некотором отношении DiffServ - например, она также помечает трафик во входных точках сети и снимает отметки в выходных. Но в отличие от DiffServ, использующего маркировку для задания приоритета маршрутизатору, 20-битные метки MPLS предназначены для определения маршрутизатора

следующего шага, то есть для продвижения пакетов. Протокол MPLS не управляется приложениями, поэтому API MPLS не существует. В отличие от любого протокола QoS, MPLS работает только в маршрутизаторах. Технология MPLS протокольно независима и может использоваться с любым сетевым протоколом: IP, IPX, ATM, PPP или frame relay, а также непосредственно над канальным уровнем.

Протокол MPLS является во многом протоколом "конструирования трафика", а не протоколом QoS. Маршрутизация MPLS используется для образования виртуальных каналов в IP-сетях (или IPX-сетях, или сетях с другим сетевым протоколом), причем предполагается, что для этих каналов маршрутизаторы сети выделяют определенные ресурсы. При этом потоку трафика, следующему вдоль виртуального канала, гарантируются параметры QoS, такие как пропускная способность или максимальный уровень задержек (как это делается для виртуальных каналов в технологиях ATM или frame relay). Однако сам способ резервирования и поддержки качества обслуживания остается за пределами протокола MPLS, он только создает виртуальный канал и может переносить в поле метки требования QoS. Резервирование пропускной способности для виртуального канала MPLS может выполнять как администратор, так и другой протокол, например RSVP.

Протокол MPLS упрощает процесс маршрутизации, уменьшая накладные расходы и увеличивая производительность. Схема работы MPLS-маршрутизатора, называемого *Label Switching Router* (LSR), состоит в следующем:

- первый маршрутизатор в сети MPLS принимает решение о продвижении пакета на основании его IP-адреса назначения или на основании любой другой информации заголовка, как определено локальной политикой, в том числе и параметров QoS. Затем маршрутизатор определяет соответствующее значение метки, идентифицирующей определенный класс эквивалентности продвижения пакетов, а затем присоединяет метку к пакету и передает ее следующему маршрутизатору;

• на следующем шаге маршрутизатор использует значение метки как индекс в таблице продвижения (не нужно ее путать с таблицей маршрутизации, в которой указываются не метки, а IP-адреса), которая определяет следующий шаг и новое значение метки. Затем LSR присоединяет к пакету новую метку и продвигает его следующему маршрутизатору.

Как видно из описания, метки имеют локальное значение, подобно идентификаторам виртуальных каналов frame relay или ATM. Маршрут, которому следует помеченный пакет, называется "коммутируемый метками путь" *(label switched path).* Основная идея MPLS состоит в том, что маршрутизаторы, определяющие на основе метки следующий шаг, выполняют меньший объем работы и действуют подобно простым коммутаторам. Сетевые администраторы имеют большую степень контроля для точного конструирования трафика, используя механизм меток и политику для их назначения. Например, администратор может проложить в сети постоянный виртуальный канал, по которому будут направляться пакеты определенного потока.

Процесс назначения меток в действительности несколько более сложен, чем он описан выше, так как метки могут образовывать стек (маршруты внутри маршрутов), а помеченные пакеты имеют поле TTL, как показано на рисунке 8.8. Поле TTL работает точно так же, как поле TTL в заголовке IP: каждый маршрутизатор уменьшает его на 1, пока оно не достигнет 0. Разница состоит в том, что при достижении нулевого значения действия LSR-маршрутизатора зависят от метки (то есть пакет не всегда отбрасывается). Процесс обработки меток представляет собой сравнительно простой аспект MPLS.

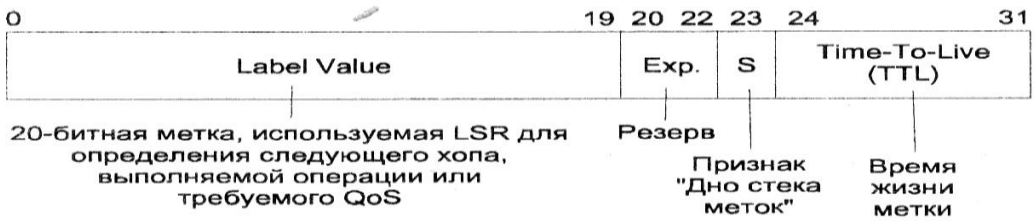


***Рис. 8.8.*** *Стек меток протокола MPLS*

Более сложным аспектом MPLS является распределение и управление метками в LSR-маршрутизаторах, чтобы они согласованно воспринимали значения

меток. Для этих целей специально разработан протокол распределения меток. Он называется MPLS LDP *(Labe1 Distribution Protocol).* Он не является единственным: есть предложение использовать в этих целях протоколы RSVP, BGP, PIM. Так что можно ожидать сосуществования нескольких протоколов распределения меток.

**Комбинирование протоколов QoS IP**

В реальном мире маловероятно, чтобы все QoS-протоколы применялись независимо, фактически они и разработаны для совместного использования с другими технологиями QoS, чтобы обеспечить поддержку качества обслуживания "сверху донизу" и "из-конца-в-конец" (рис. 8.9.).

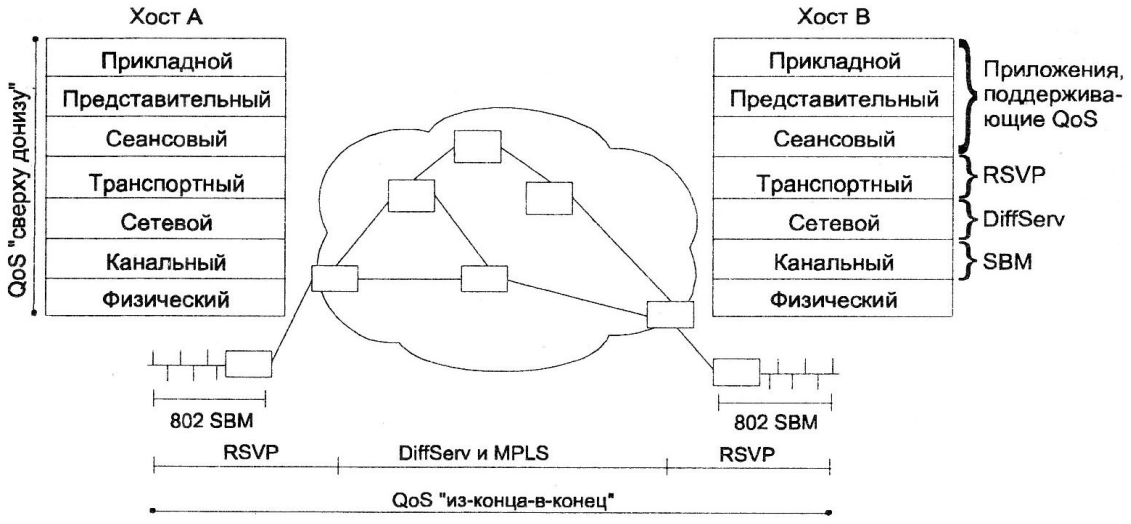


***Рис. 8.9.*** *Области применения различных протоколов QoS*

Хотя большая часть спецификаций, предназначенных для "склеивания" элементов QoS вместе, пока не стандартизована, проводится работа по определению различных архитектур, способных обеспечить поддержку QoS "из-конца-в-конец". Рисунок 8.9. представляет собой высокоуровневый взгляд на то, как эти элементы должны работать совместно. А на рисунке 8.10. дана более детальная иллюстрация той же самой идеи.

Взаимодействие протоколов RSVP и DiffServ

На рисунке 8.10. приведена полная картина того, как QoS-технологии могут работать вместе для обеспечения QoS "из-конца-в-конец". Брокер пропускной

способности представляет собой элемент новой концепции управления качеством обслуживания на основе централизованной политики. Другим элементом этой концепции является протокол COPS, с помощью которого брокеры пропускной способности распределяют данные политики по маршрутизаторам сети. Протокол RSVP выделяет ресурсы для сетевого трафика на основе переговорной процедуры, в то время как DiffServ просто помечает и приоритезирует трафик. RSVP предъявляет к маршрутизаторам гораздо более жесткие требования по сравнению с DiffServ, так что он может плохо влиять на магистральные маршрутизаторы. Именно поэтому практический опыт показывает, что на магистрали применение протокола DiffServ является более предпочтительным, чем применение RSVP.

DiffServ хорошо дополняет RSVP, и их комбинация способна обеспечить QoS "из-конца-в-конец". Конечные узлы могут использовать запросы RSVP с высокой детализацией требований (например, к пропускной способности, порогу вариации задержки и т. п.) Затем пограничные маршрутизаторы магистрали могут отобразить параметры резервирования RSVP на классы обслуживания с помощью значения DS-байта. Значение DS-байта может также установить узел-отправитель. В точке выхода магистрали снова может быть осуществлено резервирование RSVP на пути до узла назначения. В сущности, точки входа выполняют формирование трафика ддя клиентов, чтобы гарантировать соглашение SLA.

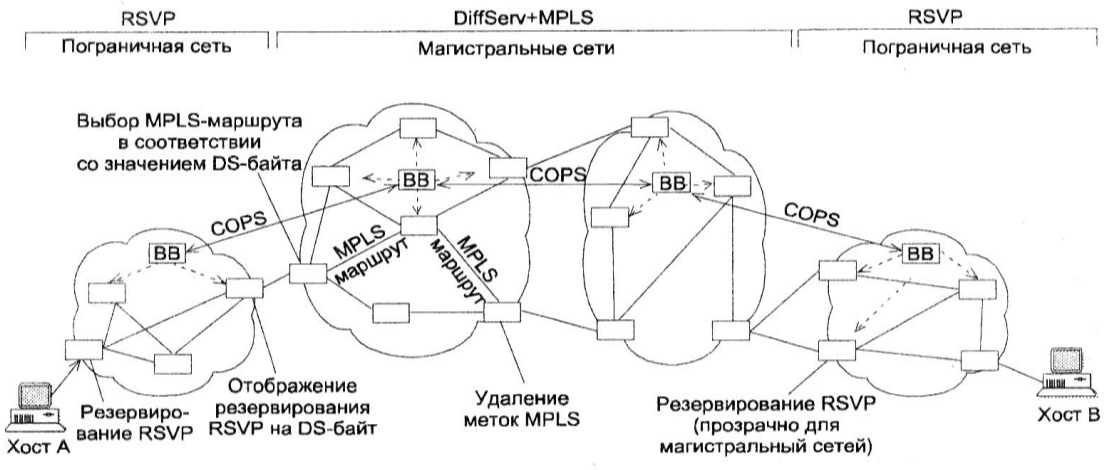


***Рис. 8.10.*** *Модель совместного функционирования протоколов QoS*

Архитектура, представленная на рисунке 8.10, - RSVP на границе сети и Diff'Seiv на магистрали - имеет мощную поддержку в виде активной деятельности рабочей группы IETF DiffServ.

RSVP DCLASS-объект

С помощью объекта DCLASS значения резервирований RSVP могут быть отображены на установки DiffServ. Объект DCLASS содержится в сообщении резервирования RSVP, который, в свою очередь, содержит предпочтительное значение кода DiffSem (этот объект является аналогом объекта TCLASS, используемого протоколом SBM). Объект DCLASS может быть добавлен либо отправителем, либо любым RSVP-маршрутизатором вдоль маршрута.

Недостатком назначения DCLASS получателем или промежуточным узлом является то, что эти сетевые элементы могут не знать наилучших значений для использования в других частях сети. Резервирования RSVP могут быть отвергнуты, если установки DCLASS окажутся неподходящими. Кроме того, любой промежуточный сетевой элемент способен просто проигнорировать "предложение" DCLASS и помечать пакеты в соответствии со своей локальной политикой.

RSVP для наборов потоков

Как уже было сказано, классифицируя потоки трафика, протоколы DiffServ и MPLS создают эффективные "каналы" для этих наборов. Для того чтобы обеспечить эти каналы качеством обслуживания, лучшим, чем сервис "с максимальными усилиями", трафик этих виртуальных каналов не должен превосходить пропускную способность сети. Проблема состоит в том, что ни DiffServ, ни MPLS не обладают механизмами определения потребностей в пропускной способности потоков и назначения необходимых ресурсов для выделенного использования. Такие механизмы есть только у протоколов RSVP.

Следовательно, хотя RSVP был разработан для распределения пропускной способности индивидуальных потоков, при таком распределении также очень важно учитывать потребности агрегированного трафика. Эти потребности создают достаточно сложную проблему для сетевых инженеров, использующих

DiffServ или MPLS, состоящую в необходимости предугадывания требований агрегированных потоков к пропускной способности, на основании которых можно было бы сделать соответствующий запрос резервирования ресурсов.

Ключевой проблемой для RSVP версии 1, как это указано в руководстве по его применению, является отсутствие механизмов агрегирования индивидуальных сессий резервирования в общий класс. Такое резервирование необходимо для масштабируемости. Поэтому в дополнение к проблеме использования RSVP для агрегированных потоков QoS существует проблема использования RSVP для агрегирования резервированных сессий RSVP.

MPLS для RSVP

Существует предложение использовать в протоколе RSVP объект "точный маршрут" *(explicit route)* для направления переключаемых на основе меток потоков RSVP по предопределенным маршрутам. Эти потоки используют виртуальные каналы, установленные в MPLS-маршрутизаторах (LSR), как показано на рис. 8.10. Даже без использования объекта *explicit route* при резервировании RSVP можно сделать так, чтобы MPLS назначал метки в соответствии со спецификацией потока RSVP. В любом случае эффект проявляется в значительном упрощении поддержки RSVP в MPLS-маршрутизаторах. Имея указания в виде меток MPLS, маршрутизаторы, поддерживающие LSR, могут не заниматься поддержкой сосгояния RSVP.

MPLS для DiffServ

Отображение трафика DiffSem на "каналы" MPLS представляет собой несложную задачу, поскольку оба этих протокола основаны на классификации трафика. Для поддержания пошаговой модели DiffServ оператору сети MPLS нужно назначить метки и набор совокупных ресурсов продвижения для каждого класса продвижения DiffServ в каждом MPLS-маршрутизаторе. Кроме того, при реализации LSR может понадобиться связать пакет с определенным уровнем предпочтения отбрасывания пакета (его значение может храниться в поле "Experimental" заголовка MPLS).

**Поддержка группового вещания в протоколах QoS**

Групповая доставка IP - не опция, а необходимое свойство, если Internet собирается быть масштабируемой сетью. Для QoS поддержка широковещательной рассылки "один-ко-многим" аудио- и видеоинформации через Internet является естественным дополнительным свойством, поэтому при разработке протоколов QoS поддержка групповой рассылки всегда была фундаментальным требованием. Хотя определенные попытки в этом направлении всегда делались при начальной разработке протоколов QoS, полная поддержка QoS групповой доставки пока еще не стандартизована. Она затрагивает много вопросов, которые рассматриваются ниже.

Поддержка групповой доставки в RSVP

При разработке протокола RSVP и интегрированных сервисов поддержка групповой рассылки IP учитывалась в том, что резервирование выполняется приемником. Одна из проблем поддержки групповой рассылки состоит в том, что приемники, принадлежащие к одной группе, могут иметь существенно отличающиеся требования к пропускной способности нисходящих к ним потоков данных. Эта неоднородность приемников данных может приводить к большим вариациям запросов на резервирование, относящимся к пути, по которому данные будут проходить. Следовательно, существенно, чтобы каждому приемнику разрешалось задавать различные параметры резервирования в соответствии с его потребностями.

Другим аспектом проблемы неоднородности приемников является возможность задавать спецификации условий фильтрации. При этом допускается построение иерархии потоков данных. Иерархические потоки данных конструируются так, что когда доступно меньше пропускной способнсти, приемники все еще получают пригодные для использования данные, хотя и с меньшей точностью. Спецификации фильтров в этом случае резервируют пропускную способность для части потока, которую низкоскоростной приемник может принять.