

# Технология ATM

Технология ATM (Asynchronous Transfer Mode — асинхронный режим передачи) была разработана как единый универсальный транспорт для нового поколения сетей с интегрированным обслуживанием, которые называются также широкополосными сетями ISDN (Broadband ISDN, B-ISDN). По сути, ATM стала второй попыткой построения универсальной сети после неудачи ISDN. В отличие от технологии Frame Relay, которая изначально предназначалась только для передачи эластичного компьютерного трафика, цели разработчиков ATM были значительно шире.

Технология ATM должна была обеспечивать:

- ❑ передачу трафика любого типа, как компьютерного, так и мультимедийного (голос, видео, управление в реальном времени), причем для каждого вида трафика качество обслуживания должно соответствовать его потребностям;
- ❑ иерархию скоростей передачи данных, от десятков мегабит до нескольких гигабит в секунду с гарантированной пропускной способностью для критически важных приложений;
- ❑ возможность использования имеющейся инфраструктуры линий связи и физических протоколов (PDH, SDH, High-speed LAN);
- ❑ взаимодействие с унаследованными протоколами локальных и глобальных сетей (IP, SNA, Ethernet, ISDN).

Необходимо сразу подчеркнуть, что большая часть этих целей была достигнута, и с середины 90-х годов ATM является работающей технологией, обеспечивающей наиболее полную и последовательную поддержку параметров QoS для пользователей сети. Кроме того, ATM, как и любая технология на основе техники виртуальных каналов, предоставляет широкие возможности по решению задач инжиниринга трафика.

Разработку стандартов ATM осуществляет большое количество производителей телекоммуникационного оборудования и операторов связи, входящих в форум ATM, а также комитеты ITU-T и ANSI.

Несмотря на очевидные успехи технологии ATM, которая работает на многих магистралях крупнейших операторов связи, опыт эксплуатации показал и ее ограничения. Так, технологии ATM не вытеснила все остальные технологии и не стала единственной транспортной технологией телекоммуникационных сетей, хотя в середине 90-х годов казалось, что благодаря очевидным технологическим достоинствам ATM это неминуемое должно произойти. Теоретически, ATM может использоваться непосредственно прикладным уровнем протоколов, так что сеть может работать без протоколов IP и TCP/UDP. ATM обладает для этого многими качествами: поддержкой всех видов трафика, масштабируемостью и собственным сложным протоколом маршрутизации. Однако это возможно только в том случае, если сеть является технологически однородной, когда все сети всех поставщиков услуг поддерживают ATM. Очевидно, такой подход противоречит принципу составных сетей, согласно которому каждая сеть может поддерживать собственную транспортную технологию, а общий сетевой уровень объединяет эти сети в единую сеть.

Поэтому на практике протокол IP, начавший доминировать на сетевом уровне в середине 90-х годов, по-прежнему используется для объединения сетей, а ATM остается одной из технологий, на основе которой работают многие сети, образующие составную сеть. Проблемы взаимодействия ATM с IP рассматриваются в главе 22.

## Основные принципы технологии ATM

Сеть ATM имеет классическую иерархическую структуру крупной территориальной сети — конечные станции соединяются индивидуальными линиями связи с коммутаторами нижнего уровня, которые в свою очередь соединяются с коммутаторами более высоких уровней. Коммутаторы ATM с момента рождения этой технологии поддерживают как каналы PVC, так и каналы SVC. Для сетей ATM определен протокол маршрутизации **PNNI** (Private NNI — частный интерфейс NNI), с помощью которого коммутаторы могут строить таблицы маршрутизации автоматически, причем с учетом требований инжиниринга трафика. В

публичных сетях ATM обычно используются адреса в стандарте E.164, что делает простым взаимодействие этих сетей с телефонными сетями. Адреса ATM имеют иерархическую структуру, подобно телефонным номерам или IP-адресам, которая обеспечивает масштабируемость сетей ATM до любого уровня, даже общемирового.

В больших сетях применяется понятие агрегированного виртуального пути, который объединяет виртуальные каналы, имеющие в сети ATM общий маршрут между исходным и конечным узлами или общую часть маршрута между некоторыми двумя коммутаторами сети. Это свойство также обеспечивает масштабируемость сетей ATM, так как позволяет существенно сократить количество виртуальных соединений, которые поддерживает магистральный коммутатор, а, значит, повысить эффективность его работы.

Стандарт ATM не вводит свои спецификации на реализацию физического уровня. Здесь он основывается на технологии SDH/SONET, принимая ее иерархию скоростей. В соответствии с этим начальная скорость доступа пользователя сети — это скорость STM-1/OC-3 155 Мбит/с. Магистральное оборудование ATM работает и на более высоких скоростях STM-4 622 Мбит/с и STM-16 2,5 Гбит/с. Существует также оборудование ATM, которое также поддерживает скорости PDH, такие как 2 или 34/45 Мбит/с.

Однако все перечисленные пока характеристики технологии ATM не свидетельствуют о том, что это некая «особенная» технология, а скорее представляют ее как достаточно развитую, но в то же время достаточно типичную технологию глобальных сетей, основанную на технике виртуальных каналов.

Главным свойством ATM, которое отличает ее от других технологий, является комплексная поддержка параметров QoS для всех основных видов трафика.

Для достижения этого свойства разработчики ATM тщательно проанализировали все типы трафика и провели его классификацию. Мы уже познакомились с этой классификацией в главе 7, когда рассматривали требования различных приложений к QoS. Напомним, что ATM разбивает весь трафик на 5 классов, A, B, C, D и X. Первые четыре класса представляют трафик типовых приложений, которые отличаются устойчивым набором требований к задержкам и потерям пакетов, а также тем, генерируют они трафик с постоянной (CBR) или переменной (VBR) битовой скоростью. Класс X зарезервирован для уникальных приложений, набор характеристик и требований которых не относится ни к одному из первых четырех классов.

Однако на какое количество классов мы бы ни разбивали существующий трафик, принципиальная задача от этого не меняется — нужно найти решение для успешного сосуществования в одном канале и эластичных, и чувствительных к задержкам классов трафика. Требования этих классов почти всегда противоречат друг другу. Одним из таких противоречий является требование к размеру кадра.

Эластичный трафик выигрывает от увеличения размера кадра, так как при этом уменьшаются накладные расходы на служебную информацию. Мы видели на примере Ethernet, что скорость передачи пользовательской информации может изменяться почти в два раза при изменении размера поля данных от его минимальной величины в 46 байт до максимальной в 1500 байт. Конечно, размер кадра не может увеличиваться до бесконечности, так как при этом теряется сама идея коммутации пакетов. Тем не менее, для эластичного трафика при современном уровне скоростей размер кадра в несколько тысяч байтов является вполне приемлемым.

Напротив, чувствительный к задержкам трафик обслуживается лучше при использовании кадров небольшого размера в несколько десятков байтов. При применении больших кадров начинают проявляться два нежелательных эффекта:

- ожидание низкоприоритетных кадров в очередях;
- задержка пакетизации.

Рассмотрим эти эффекты на примере голосового трафика.

Мы знаем, что *время ожидания кадра в очереди* можно уменьшить, если обслуживать кадры чувствительного к задержкам трафика в приоритетной очереди. Однако если размер кадра может меняться в широком диапазоне (например, от 29 до 4500 байт, как в технологии FDDI), то даже при придании чувствительным к задержкам кадрам высшего приоритета обслуживания в коммутаторах время ожидания компьютерного пакета может все равно оказаться недопустимо высоким. Например, пакет в 4500 байт будет передаваться в выходной порт на скорости

2 Мбит/с (максимальная скорость работы порта коммутатора Frame Relay) 18 мс. При совмещении трафика за это время необходимо через тот же порт передать 144 замера голоса. Прерывать передачу пакета в сетях нежелательно, так как при распределенном характере сети накладные расходы на оповещение соседнего коммутатора о прерывании пакета, а потом — о возобновлении передачи пакета с прерванного места оказываются слишком большими.

**Задержка пакетизации** — это время, в течение которого первый замер голоса ждет момента окончательного формирования пакета и отправки его по сети. Механизм образования этой задержки иллюстрирует рис. 21.8.

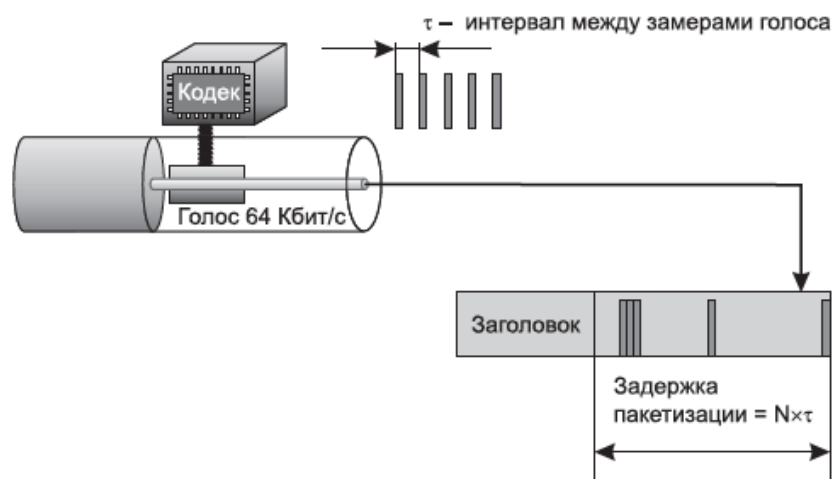


Рис. 21.8. Задержка пакетизации

Кодек делает замеры голоса через одинаковые интервалы времени. На рисунке в качестве примера он делает это с частотой 8 КГц, то есть через каждые 125 мкс. Если мы используем для передачи голоса кадры Ethernet максимального размера, то в один кадр помещается 1500 замеров голоса, так как каждый замер кодируется одним байтом данных. В результате первый замер, помещенный в кадр Ethernet, вынужден ждать отправки кадра в  $(1500 - 1) \times 125 = 187375$  мкс, или около 187 мс. Это весьма большая задержка для голосового трафика, рекомендации ИТУ-Тговорят о величине 150 мс как о максимально допустимой задержке голоса. Важно отметить, что задержка пакетизации не зависит от битовой скорости протокола, она зависит только от частоты работы кодека и размера поля данных кадра. Это отличает ее от задержки ожидания, которая снижается с возрастанием битовой скорости.

**Кадр ATM** в 53 байта с полем данных 48 байт явился результатом компромисса между требований эластичного и чувствительного к задержкам трафиков. Другими словами, можно сказать, что компромисс был достигнут между телефонистами и компьютерщиками — первые настаивали на размере поля данных в 32 байта, а вторые — в 64 байта. Небольшой и фиксированный размер кадра ATM дал ему специальное название — **ячейка**.

При размере поля данных в 48 байт одна ячейка ATM обычно переносит 48 замеров голоса, которые делаются с интервалом в 125 мкс. Поэтому первый замер должен ждать примерно 6 мс, прежде чем ячейка будет отправлена по сети. Именно по этой причине телефонисты боролись за уменьшения размера ячейки, так как 6 мс — это задержка, близкая к пределу, за которым начинаются нарушения качества передачи голоса. При выборе размера ячейки в 32 байта задержка пакетизации составила бы 4 мс, что гарантировало бы более качественную передачу голоса. А стремление компьютерных специалистов увеличить поле данных до 64 байт вполне понятно — при этом повышается полезная скорость передачи данных. Избыточность служебных данных при использовании 48-байтного поля данных составляет 10 %, а при использовании 32-байтного поля данных она сразу повышается до 16 %.

Для пакета, состоящего из 53 байт, при скорости в 155 Мбит/с время передачи кадра на выходной порт составляет менее 3 мкс. Так что эта задержка не очень существенна для трафика, пакеты которого должны передаваться каждые 125 мкс.

Чтобы пакеты содержали адрес узла назначения, и, в то же время, процент служебной информации не превышал размер поля данных пакета, в технологии ATM применен

стандартный для WAN прием — передача ячеек в соответствии с техникой виртуальных каналов. Общая длина номера виртуального канала составляет 24 бит, что вполне достаточно для обслуживания большого количества виртуальных соединений каждым портом коммутатора глобальной (может быть всемирной) сети ATM.

Нужно отметить, что использование в ATM ячеек такого небольшого размера, создающих отличные условия для качественного обслуживания чувствительного к задержкам трафика, имеет и обратную сторону. Платой за качество является высокий уровень нагрузки на ATM-коммутаторы при работе на высоких скоростях. Напомним, что объем работы, который выполняет коммутатор или маршрутизатор любой технологии, прямо пропорционален количеству обрабатываемых в единицу времени пакетов, или кадров. Очевидно, что использование ячеек размеров с полем данных 48 байт приводит к колоссальному росту объема работы для ATM-коммутатора по сравнению с, например, коммутатором Ethernet, работающим с кадрами 1500 байт. Из-за этого обстоятельства ATM-коммутаторы долго не могли превзойти границу скорости интерфейсов в 622 Мбит/с и сравнительно недавно стали поддерживать интерфейсы 2,5 Гбит/с.

Выбор для передачи данных любого типа небольшой ячейки фиксированного размера еще не решает задачу совмещения разнородного трафика в одной сети, а только создает предпосылки для ее решения. Для полного решения этой задачи технология ATM привлекает и развивает идеи *резервирования пропускной способности и качества обслуживания*, реализованные в технологии Frame Relay.

В технологии ATM для каждого класса трафика определен набор количественных параметров, которые приложение должно задать. Например, для трафика класса А необходимо указать постоянную скорость, с которой приложение будет посылать данные в сеть, а для трафика класса В — максимально возможную скорость, среднюю скорость и максимально возможную пульсацию. Для голосового трафика можно не только указать на важность синхронизации между передатчиком и приемником, но и количественно задать верхние границы задержек и вариации задержек ячеек.

В технологии ATM поддерживается следующий набор основных количественных параметров для трафика виртуального соединения:

- *пиковая скорость передачи ячеек* (Peak Cell Rate, PCR);
- *средняя скорость передачи ячеек* (Sustained Cell Rate, SCR);
- *минимальная скорость передачи ячеек* (Minimum Cell Rate, MCR);
- *максимальная величина пульсаций* (Maximum Burst Size, MBS);
- *доля потерянных ячеек* (Cell Loss Ratio, CLR);
- *задержка передачи ячеек* (Cell Transfer Delay, CTD);
- *вариация задержек ячеек* (Cell Delay Variation, CDV).

Параметры скорости измеряются в ячейках в секунду, максимальная величина пульсаций — в ячейках, а временные параметры — в секундах. Максимальная величина пульсаций определяет количество ячеек, которое приложение может передать с пиковой скоростью при заданной средней скорости. Доля потерянных ячеек является отношением потерянных ячеек к общему количеству отправленных ячеек по данному виртуальному соединению. Так как виртуальные соединения являются дуплексными, то для каждого направления соединения могут быть заданы разные значения параметров.

В технологии ATM принят не совсем традиционный подход к трактовке качества обслуживания (QoS). Обычно качество обслуживания трафика характеризуется параметрами пропускной способности (здесь это RCR, SCR, MCR, MBS), параметрами задержек пакетов (CTD и CDV), а также параметрами надежности передачи пакетов (CLR). В ATM скоростные характеристики называют *параметрами трафика* и не включают их в число параметров качества обслуживания, хотя по существу они таковыми являются. Параметрами QoS в ATM являются только CTD, CDV и CLR. Сеть старается обеспечить такой уровень обслуживания, чтобы поддерживались требуемые значения и для параметров трафика, и для задержек ячеек, и для доли потерянных ячеек.

Соглашение между приложением и сетью ATM называется трафик-контрактом. Основным его

отличием от соглашений, применяемых в сетях Frame Relay, является выбор одного из нескольких определенных классов трафика, для которого наряду с параметрами пропускной способности трафика могут указываться параметры задержек ячеек, а также параметр надежности доставки ячеек. В сети Frame Relay класс трафика один, и он характеризуется только параметрами пропускной способности.

Если для приложения не критично поддержание параметров пропускной способности и QoS, то оно может отказаться от задания этих параметров, указав в запросе на установление соединения признак обслуживания с максимальными усилиями. Такой тип трафика получил название трафика с неопределенной битовой скоростью (Unspecified Bit Rate, UBR).

После заключения трафик-контракта, который относится к определенному виртуальному соединению, в сети ATM работает несколько протоколов и служб, обеспечивающих нужное качество обслуживания. Для трафика UBR сеть выделяет ресурсы «по возможности», то есть те, которые в данный момент не заняты виртуальными соединениями, заказавшими определенные параметры качества обслуживания.

## Стек протоколов ATM

Стек протоколов ATM показан на рис. 21.9, а распределение протоколов по конечным узлам и коммутаторам ATM — на рис. 21.10.

Верхние уровни сети

Уровни адаптации ATM (AAL 1-5)	Подуровень конвергенции (CS)	Общая часть подуровня конвергенции
		Специфическая для сервиса часть
	Подуровень сегментации и реассемблирования (SAR)	
Уровень ATM (маршрутизация пакетов, мультиплексирование, управление потоком, обработка приоритетов)		
Физический уровень	Подуровень согласования передачи	
	Подуровень, зависящий от физической среды	

Рис. 21.9. Структура стека протоколов ATM

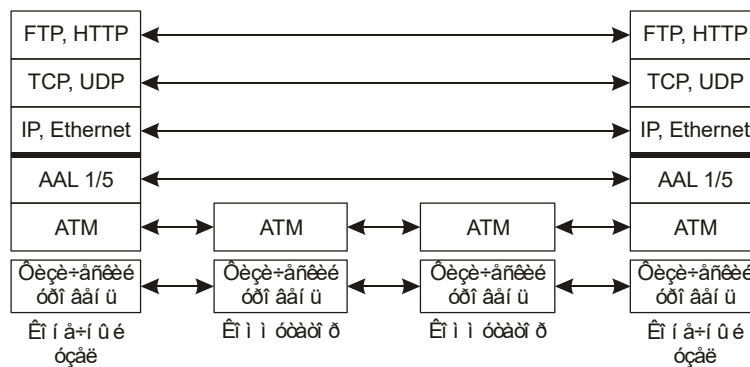


Рис. 21.10. Распределение протоколов по узлам и коммутаторам сети ATM

Стек протоколов ATM соответствует нижним уровням семиуровневой модели ISO/OSI и включает уровень адаптации ATM, собственно уровень ATM и физический уровень. Прямого соответствия между уровнями протоколов технологии ATM и уровнями модели OSI нет.

## Уровень адаптации AAL

**Уровень адаптации ATM** (ATM Adaptation Layer, AAL) представляет собой набор протоколов AAL1–AAL5, которые преобразуют сообщения протоколов верхних уровней сети ATM в ячейки ATM нужного формата. Функции этих уровней достаточно условно соответствуют функциям транспортного уровня модели OSI, например функциям протокола TCP или UDP. Протоколы AAL при передаче пользовательского трафика работают только в конечных узлах

сети, как и транспортные протоколы большинства технологий.

Каждый протокол уровня AAL обрабатывает пользовательский трафик определенного класса. На начальных этапах стандартизации каждому классу трафика соответствовал свой протокол AAL, который принимал в конечном узле пакеты от протокола верхнего уровня и заказывал с помощью соответствующего протокола нужные параметры трафика и качества обслуживания для данного виртуального канала. При развитии стандартов ATM такое однозначное соответствие между классами трафика и протоколами уровня AAL исчезло, и сегодня разрешается использовать для одного и того же класса трафика разные протоколы уровня AAL.

Уровень адаптации состоит из двух подуровней.

- ❑ **Подуровень сегментации и реасемблирования** (Segmentation And Reassembly, SAR) является нижним подуровнем AAL. Эта часть не зависит от типа протокола AAL (и, соответственно, от класса передаваемого трафика) и занимается разбиением (сегментацией) сообщения, принимаемого AAL от протокола верхнего уровня, на ячейки ATM, снабжением их соответствующим заголовком и передачей уровню ATM для отправки в сеть.
- ❑ **Подуровень конвергенции** (Convergence Sublayer, CS) — это верхний подуровень AAL. Этот подуровень зависит от класса передаваемого трафика. Протокол подуровня конвергенции решает такие задачи, как обеспечение временной синхронизации между передающим и принимающим узлами (для трафика, требующего такой синхронизации), контролем и возможным восстановлением битовых ошибок в пользовательской информации, контролем целостности передаваемого пакета компьютерного протокола (X.25, Frame Relay).

Протоколы AAL для выполнения своей работы используют служебную информацию, размещаемую в заголовках уровня AAL. После приема ячеек, пришедших по виртуальному каналу, подуровень SAR протокола AAL собирает посланное по сети исходное сообщение (в общем случае разбитое на несколько ячеек ATM) с помощью заголовков AAL, которые для коммутаторов ATM являются прозрачными, так как помещаются в 48-битном поле данных ячейки, как и полагается протоколу более высокого уровня. После сборки исходного сообщения протокол AAL проверяет служебные поля заголовка и концевого кадра AAL и на их основании принимает решение о корректности полученной информации.

Ни один из протоколов AAL при передаче пользовательских данных конечных узлов не занимается восстановлением потерянных или искаженных данных. Максимум, что делает протокол AAL, — уведомляет конечный узел о таком событии. Так сделано для ускорения работы коммутаторов сети ATM в расчете на то, что случаи потерь или искажений данных будут редкими. Восстановление потерянных данных (или игнорирование этого события) отводится протоколам верхних уровней, не входящим в стек протоколов технологии ATM.

**Протокол AAL1** обычно обслуживает трафик класса А с постоянной битовой скоростью (CBR), который характерен, например, для цифрового видео и цифровой речи, и чувствителен к временным задержкам. Этот трафик передается в сетях ATM таким образом, чтобы эмулировать обычные выделенные цифровые линии. Заголовок AAL1 занимает в поле данных ячейки ATM 1 или 2 байта, оставляя для передачи пользовательских данных соответственно 47 или 46 байт. В заголовке 1 байт отводится для нумерации ячеек, чтобы приемная сторона могла судить о том, все или не все посланные ячейки дошли до нее. При отправке голосового трафика временная отметка каждого замера известна, так как они следуют друг за другом с интервалом в 125 мкс, поэтому при потере ячейки можно скорректировать временную привязку байтов следующей ячейки, сдвинув ее на  $125 \times 46$  мкс. Потеря нескольких байтов замеров голоса не так страшна, так как на приемной стороне воспроизводящее оборудование сглаживает сигнал. В задачи протокола AAL1 входит сглаживание неравномерности поступления ячеек данных в узел назначения.

**Протокол AAL2** был разработан для передачи трафика класса В, но при развитии стандартов его исключили из стека протоколов ATM, и сегодня трафик класса В передается с помощью протокола AAL1, AAL3/4 или AAL5.

**Протокол AAL3/4** обрабатывает пульсирующий трафик — обычно характерный для трафика локальных сетей — с переменной битовой скоростью (VBR). Этот трафик обрабатывается так, чтобы не допустить потерь ячеек, но ячейки могут задерживаться коммутатором. Протокол AAL3/4 выполняет сложную процедуру контроля ошибок при передаче ячеек, нумеруя каждую

составляющую часть исходного сообщения и снабжая каждую ячейку контрольной суммой. Правда, при искажениях или потерях ячеек этот уровень не занимается их восстановлением, а просто отбрасывает все сообщение — то есть все оставшиеся ячейки, так как для компьютерного трафика или компрессированного голоса потеря части данных является фатальной ошибкой. Протокол AAL3/4 представляет собой результат слияния протоколов AAL3 и AAL4, которые обеспечивали поддержку трафика компьютерных сетей соответственно с установлением соединения и без установления соединения. Однако ввиду большой близости используемых форматов служебных заголовков и логики работы протоколы AAL3 и AAL4 были впоследствии объединены.

**Протокол AAL5** является упрощенным вариантом протокола AAL4 и работает быстрее, так как вычисляет контрольную сумму не для каждой ячейки сообщения, а для всего исходного сообщения и помещает ее в последнюю ячейку сообщения. Первоначально протокол AAL5 разрабатывался для передачи кадров сетей Frame Relay, но теперь он чаще всего применяется для передачи любого компьютерного трафика. Протокол AAL5 может поддерживать различные параметры качества обслуживания, кроме тех, которые связаны с синхронизацией передающей и принимающей сторон. Поэтому он обычно используется для поддержки всех классов трафика, относящегося к передаче компьютерных данных, то есть классов C и D. Некоторые производители оборудования с помощью протокола AAL5 обслуживают трафик CBR, оставляя задачу синхронизации трафика протоколам верхнего уровня. Протокол AAL5 работает не только в конечных узлах, но и в коммутаторах сети ATM. Однако там он выполняет служебные функции, не связанные с передачей пользовательских данных. В коммутаторах ATM протокол AAL5 поддерживает служебные протоколы более высоких уровней, занимающиеся установлением коммутируемых виртуальных соединений.

Существует определенный интерфейс между приложением, которому требуется передать трафик через сеть ATM, и уровнем адаптации AAL. С помощью этого интерфейса приложение (протокол компьютерной сети, модуль оцифровывания голоса) заказывает требуемую услугу, определяя тип трафика, его параметры, а также параметры QoS. Технология ATM допускает два варианта определения параметров QoS: первый — непосредственное задание их каждым приложением, второй — назначение их по умолчанию в зависимости от типа трафика. Последний способ упрощает задачу разработчика приложения, так как в этом случае выбор максимальных значений задержек доставки ячеек и вариации задержек перекладывается на плечи администратора сети.

Самостоятельно обеспечить требуемые параметры трафика и QoS протоколы AAL не могут. Для выполнения соглашений трафик-контракта требуется согласованная работа коммутаторов сети вдоль всего виртуального соединения. Эта работа выполняется протоколом ATM, обеспечивающим передачу ячеек различных виртуальных соединений с заданным уровнем качества обслуживания.

## Протокол ATM

**Протокол ATM** занимает в стеке протоколов ATM примерно то же место, что протокол IP в стеке TCP/IP или протокол LAP-F в стеке протоколов технологии Frame Relay. Протокол ATM передает ячейки через коммутаторы при установленном и настроенном виртуальном соединении, то есть на основании готовых таблиц коммутации портов.

Протокол ATM выполняет коммутацию по **номеру виртуального соединения**, который в технологии ATM разбит на две части:

- **идентификатор виртуального пути** (Virtual Path Identifier, VPI);
- **идентификатор виртуального канала** (Virtual Channel Identifier, VCI).

Помимо решения этой основной задачи протокол ATM выполняет ряд функций по контролю за соблюдением трафик-контракта со стороны пользователя сети, маркировке ячеек-нарушителей, отбрасыванию ячеек-нарушителей при перегрузке сети, а также управлению потоком ячеек для повышения производительности сети (естественно, при соблюдении условий трафик-контракта для всех виртуальных соединений).

Формат ячеек протокола ATM представлен на рис. 21.11.

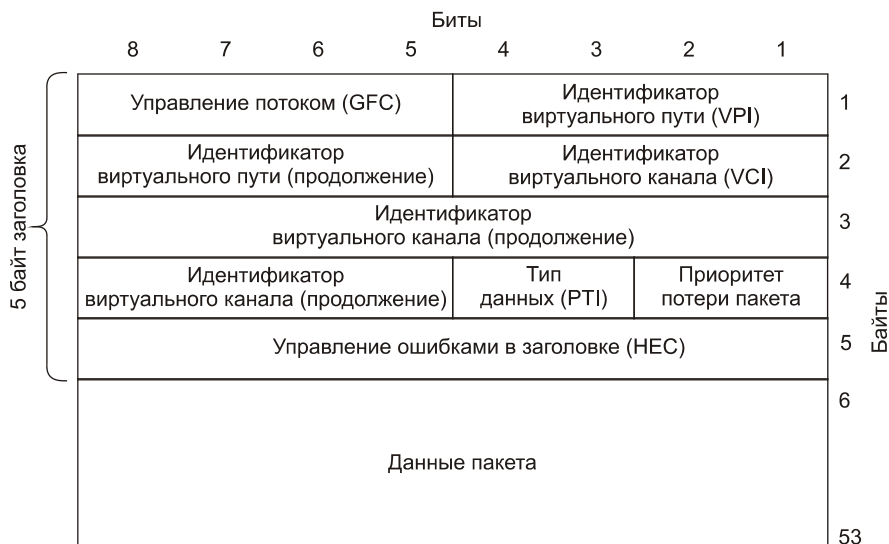


Рис. 21.11. Формат АТМ-ячейки

Поле *обобщенного управления потоком* (Generic Flow Control, GFC) используется только при взаимодействии конечного узла и первого коммутатора сети. В настоящее время его точные функции не определены.

Поля *идентификатора виртуального пути* (Virtual Path Identifier, VPI) и *идентификатора виртуального канала* (Virtual Channel Identifier, VCI) занимают соответственно 1 и 2 байта. Эти поля задают номер виртуального соединения, разделенный на старшую (VPI) и младшую (VCI) части.

Поле *идентификатора типа данных* (Payload Type Identifier, PTI) состоит из 3 бит и задает тип данных, переносимых ячейкой, — пользовательские или управляющие (например, управляющие установлением виртуального соединения). Кроме того, один бит этого поля используется для указания о перегрузке в сети. Это бит EFCI (Explicit Forward Congestion Identifier — прямой явный идентификатор перегрузки), который играет ту же роль, что и бит FECN в технологии Frame Relay, то есть передает информацию о перегрузке по направлению потока данных.

Поле *приоритета потери кадра* (Cell Loss Priority, CLP) играет в данной технологии ту же роль, что и поле DE в технологии Frame Relay — в нем коммутаторы АТМ отмечают ячейки, которые нарушают соглашения о параметрах качества обслуживания, чтобы удалить их при перегрузках сети. Таким образом, ячейки с полем CLP = 0 являются для сети высокоприоритетными, а ячейки с полем CLP = 1 — низкоприоритетными.

Поле *управления ошибками в заголовке* (Header Error Control, HEC) содержит контрольную сумму, вычисленную для заголовка ячейки. Контрольная сумма вычисляется с помощью техники корректирующих кодов Хэмминга, поэтому она позволяет не только обнаруживать ошибки, но и исправлять все одиночные ошибки, а также некоторые двойные. Кроме того, поле HEC обеспечивает не только обнаружение и исправление ошибок в заголовке, но и нахождение границы начала кадра в потоке байтов кадров SDH, которые являются предпочтительным физическим уровнем технологии АТМ, или же в потоке битов физического уровня, основанного на ячейках. Указателей, позволяющих в поле данных кадра STS-n (STM-n) технологии SONET/SDH обнаруживать границы ячеек АТМ (подобных тем указателям, которые используются для определения, например, границ виртуальных контейнеров подканалов T1/E1), не существует. Поэтому коммутатор АТМ вычисляет контрольную сумму для последовательности из пяти байтов, находящихся в поле данных кадра STM-n, и, если вычисленная контрольная сумма говорит о корректности заголовка ячейки АТМ, первый байт становится границей ячейки. Если же это не так, то происходит сдвиг на один байт и операция продолжается. Таким образом, технология АТМ выделяет асинхронный поток ячеек АТМ в синхронных кадрах SDH или потоке битов физического уровня, основанного на ячейках.

Рассмотрим методы коммутации ячеек АТМ на основе пары чисел VPI/VCI. Коммутаторы АТМ могут работать в двух режимах.

- **Коммутация виртуального пути.** В этом режиме коммутатор выполняет продвижение ячейки только на основании значения поля VPI, а значение поля VCI он игнорирует. Обычно так работают магистральные коммутаторы территориальных сетей. Они доставляют ячейки из одной пользовательской сети в другую на основании *только старшей части номера виртуального канала*, что соответствует идее агрегирования адресов. В результате один виртуальный путь соответствует целому набору виртуальных каналов, коммутируемых как единое целое.
- **Коммутация виртуального канала.** После доставки ячейки в локальную сеть ATM ее коммутаторы начинают коммутировать ячейки с учетом как поля VPI, так и поля VCI, но при этом им хватает для коммутации только младшей части номера виртуального соединения, так что фактически они работают с VCI, оставляя VPI без изменения. Этот режим и называется режимом коммутации виртуального канала.

Для создания коммутируемого виртуального канала в технологии ATM используются протоколы, не показанные на рис. 20.11. Подход здесь аналогичен подходу в сети ISDN — для установления соединения разработан отдельный **протокол Q.2931**, который весьма условно можно отнести к сетевому уровню. Этот протокол во многом похож на протоколы Q.931 и Q.933 (даже номером), но в него внесены, естественно, изменения, связанные с наличием нескольких классов трафика и дополнительных параметров качества обслуживания. Протокол Q.2931 опирается на достаточно сложный протокол канального уровня SSCOP, который обеспечивает надежную передачу пакетов Q.2931 в своих кадрах. В свою очередь, протокол SSCOP работает поверх протокола AAL5, который необходим для разбиения кадров SSCOP на ATM-ячейки и сборки этих ячеек в кадры при доставке кадра SSCOP в коммутатор назначения.

#### ПРИМЕЧАНИЕ

Протокол Q.2931 появился в стеке протоколов технологии ATM после принятия версии интерфейса UNI 3.1, а до этого в версии UNI 3.0 вместо него использовался протокол Q.93B. Из-за несовместимости протоколов Q.2931 и Q.93B версии пользовательского интерфейса UNI 3.0 и UNI 3.1 также несовместимы. Версия UNI 4.0 обратно совместима с UNI 3.1, так как основана на тех же служебных протоколах, что и версия UNI 3.1.

Виртуальные соединения, образованные с помощью протокола Q.2931, бывают симплексными (однонаправленными) и дуплексными.

Протокол Q.2931 позволяет также устанавливать двухточечные виртуальные соединения и виртуальные соединения с одним отправителем и несколькими получателями. Первый случай поддерживается во всех технологиях, основанных на виртуальных каналах, а второй характерен для технологии ATM и является аналогом групповой рассылки с одним ведущим (передающим) узлом. При установлении соединения с одним отправителем и несколькими получателями ведущим считается узел, который является инициатором этого соединения. Сначала этот узел устанавливает виртуальное соединение всего с одним узлом, а затем путем специального вызова добавляет к соединению по одному новому члену. Ведущий узел становится вершиной дерева соединения, а остальные узлы — листьями этого дерева. Сообщения, которые посылает ведущий узел, принимают все листья соединения, но сообщения, которые посылает какой-либо лист (если соединение дуплексное), принимает только ведущий узел.

Пакеты протокола Q.2931, предназначенные для установления коммутируемого виртуального канала, имеют те же названия и то же назначение, что и пакеты протокола Q.933, рассмотренные при изучении технологии Frame Relay, но структура их полей, естественно, другая.

Адрес конечного узла в коммутаторах ATM 20-байтный.

- При работе в публичных сетях используется адрес стандарта E.164. Адрес имеет гибкий формат и может делиться на части для обеспечения иерархической маршрутизации между сетями и подсетями. Он поддерживает больше уровней иерархии, чем IPv4-адрес, и похож в этом отношении на IPv6-адрес.
- Последние 6 байт адреса занимает поле идентификатора конечной системы (End System Identifier, ESI), которое имеет смысл MAC-адреса узла ATM, причем формат его также соответствует формату MAC-адреса.

ESI-адрес присваивается конечному узлу на предприятии-изготовителе в соответствии с

правилами IEEE, то есть 3 первых байта содержат код предприятия, а остальные три — порядковый номер, за уникальность которого отвечает данное предприятие.

При работе в частных сетях ATM обычно применяются адреса близкого к E.164 формату с небольшим изменением.

Конечный узел при подключении к коммутатору ATM выполняет так называемую процедуру регистрации. При этом конечный узел сообщает коммутатору свой ESI-адрес, а коммутатор сообщает конечному узлу старшую часть адреса, то есть номер сети, в которой работает узел.

Кроме адресной части пакет CALL SETUP протокола Q.2931, с помощью которого конечный узел запрашивает установление виртуального соединения, включает также части, описывающие параметры трафика и требования QoS. При поступлении такого пакета коммутатор должен проанализировать эти параметры и решить, достаточно ли у него свободных ресурсов для обслуживания нового виртуального соединения. Если да, то новое виртуальное соединение принимается, и коммутатор передает пакет CALL SETUP дальше в соответствии с адресом назначения и таблицей маршрутизации, а если нет, запрос отвергается.

## Категории услуг протокола ATM и управление трафиком

Для поддержания требуемого качества обслуживания различных виртуальных соединений и рационального использования ресурсов в сети на уровне протокола ATM реализовано несколько служб, предоставляющих для обслуживания пользовательского трафика услуги различных категорий. Эти службы являются внутренними службами сети ATM, они предназначены для поддержания пользовательского трафика различных классов совместно с протоколами AAL. Но в отличие от протоколов AAL, которые работают в конечных узлах сети, данные службы распределены по всем коммутаторам сети. Услуги этих служб разбиты на категории, которые в общем соответствуют классам трафика, поступающим на вход уровня AAL конечного узла. Услуги уровня ATM заказываются конечным узлом через интерфейс UNI с помощью протокола Q.2931 при установлении виртуального соединения. Как и при обращении к уровню AAL, при заказе услуги необходимо указать категорию услуги, а также параметры трафика и параметры QoS. Эти параметры берутся из аналогичных параметров уровня AAL или же определяются по умолчанию в зависимости от категории услуги.

Всего на уровне протокола ATM определено пять категорий услуг:

- ❑ **CBR** (Constant Bit Rate) — услуги для трафика с постоянной битовой скоростью;
- ❑ **rtVBR** (real-time Variable Bit Rate) — услуги для трафика с переменной битовой скоростью, требующего соблюдения средней скорости передачи данных и синхронизации источника и приемника;
- ❑ **nrtVBR** (non real-time Variable Bit Rate) — услуги для трафика с переменной битовой скоростью, требующего соблюдения средней скорости передачи данных и не требующего синхронизации источника и приемника;
- ❑ **ABR** (Available Bit Rate) — услуги для трафика с переменной битовой скоростью, требующего соблюдения некоторой минимальной скорости передачи данных и не требующего синхронизации источника и приемника;
- ❑ **UBR** (Unspecified Bit Rate) — услуги для трафика, не предъявляющего требований к скорости передачи данных и синхронизации источника и приемника.

Названия большинства категорий услуг совпадают с названием типов пользовательского трафика, для обслуживания которого они разработаны, но необходимо понимать, что сами службы уровня ATM и их услуги — это внутренние механизмы сети ATM, которые экранируются от приложения уровнем AAL.

*Услуги категории CBR* предназначены для поддержания трафика синхронных приложений — голосового, эмуляции цифровых выделенных каналов и т. п. Когда приложение устанавливает соединение категории CBR, оно заказывает пиковую скорость передачи ячеек (PCR), являющуюся максимальной скоростью, которую может поддерживать соединение без риска потерять ячейку, а также параметры QoS: величины максимальной задержки ячеек (CTD), вариации задержек ячеек (CDV) и максимальной доли потерянных ячеек CLR.

Затем данные передаются по этому соединению с запрошенной скоростью — не с большей и, в

большинстве случаев, не с меньшей, хотя уменьшение скорости приложением возможно, например, при передаче компрессированного голоса при выполнении услуги категории CBR. Любые ячейки, передаваемые станцией с большей скоростью, контролируются первым коммутатором сети и помечаются признаком  $CLP = 1$ . При перегрузках сети они могут просто отбрасываться сетью. Ячейки, которые запаздывают и не укладываются в интервал, оговоренный параметром вариации задержки (CDV), также считаются мало значащими для приложения и отмечаются признаком низкого приоритета  $CLP = 1$ .

Для соединений CBR нет ограничений на некоторую дискретность заказа скорости PCR, как, например, в каналах T1/E1, где скорость должна быть кратна 64 Кбит/с.

По сравнению со службой CBR, *службы VBR* требуют более сложной процедуры заказа соединения между сетью и приложением. В дополнение к пиковой скорости PCR приложение VBR заказывает еще и два других параметра: длительно поддерживаемую скорость — SCR, которая представляет собой среднюю скорость передачи данных, разрешенную приложению, а также максимальный размер пульсаций — MBS. Максимальный размер пульсаций измеряется в количестве ATM-ячеек. Пользователь может превышать скорость вплоть до величины PCR, но только на короткие периоды времени, в течение которых передается объем данных, не превышающий MBS. Этот период времени называется *терпимостью к пульсациям* (Burst Tolerance, BT). Сеть вычисляет этот период как производный от трех заданных значений PCR, SCR и MBS.

Если скорость PCR наблюдается в течение периода времени, большего чем BT, то ячейки помечаются как «нарушители» — устанавливается признак  $CLP = 1$ .

Для *услуг категории rtVBR* задаются и контролируются те же параметры QoS, что и для услуг категории CBR, а услуги категории nrtVBR ограничиваются поддержанием параметров трафика. Сеть также поддерживает для обеих категорий услуг VBR определенный максимальный уровень доли потерянных ячеек CLR, который либо задается явно при установлении соединения, либо назначается по умолчанию в зависимости от класса трафика.

Для контроля параметров трафика и QoS в технологии ATM применяется так называемый *обобщенный алгоритм контроля скорости ячеек*, который может проверять соблюдение пользователем и сетью таких параметров, как PCR, CDV, SCR, BT, CTD и CDV. Он работает по модифицированному алгоритму дырявого ведра, применяемому в технологии Frame Relay.

Для многих приложений, которые могут быть чрезвычайно «взрывными» в отношении интенсивности трафика, невозможно точно предсказать параметры трафика, оговариваемые при установлении соединения. Например, обработка транзакций или трафик двух взаимодействующих локальных сетей непредсказуемы по своей природе — изменения интенсивности трафика слишком велики, чтобы заключить с сетью какое-либо разумное соглашение.

В отличие от CBR и обеих служб VBR, *служба UBR* не поддерживает ни параметры трафика, ни параметры качества обслуживания. Служба UBR предлагает только доставку с максимальными усилиями (по возможности) без каких-либо гарантий. Разработанная специально для превышения полосы пропускания служба UBR представляет собой частичное решение для тех непредсказуемых «взрывных» приложений, которые не готовы согласиться с фиксацией параметров трафика.

Главными недостатками услуг UBR являются отсутствие механизмов управления потоком данных и неспособность принимать во внимание другие типы трафика. Несмотря на перегрузку сети, соединения UBR будут продолжать передачу данных. Коммутаторы сети могут буферизовать некоторые ячейки поступающего трафика, но в некоторый момент буферы переполняются, и ячейки теряются. А так как для соединений UBR не оговаривается никаких параметров трафика и QoS, то их ячейки отбрасываются в первую очередь.

*Служба ABR* подобно службе UBR предоставляет возможность превышения полосы пропускания, но благодаря технике управления трафиком при перегрузке сети она дает некоторые гарантии сохранности ячеек.

ABR — это первый тип служб уровня ATM, который действительно обеспечивает надежный транспорт для пульсирующего трафика за счет того, что может находить неиспользуемые интервалы в общем трафике сети и заполнять их своими ячейками, если другим категориям служб эти интервалы не нужны.

Как и в службах CBR и VBR, при установлении соединения категории ABR оговаривается значение пиковой скорости (PCR). Однако соглашение о пределах изменения задержек передачи ячеек или о параметрах пульсаций не заключается. Вместо этого сеть и конечный узел заключают соглашение о требуемой минимальной скорости передачи (MCR). Это гарантирует приложению, работающему в конечном узле, небольшую пропускную способность, обычно минимально необходимую для того, чтобы приложение работало. Конечный узел соглашается не передавать данные со скоростью, выше пиковой, то есть PCR, а сеть соглашается всегда обеспечивать минимальную скорость передачи ячеек.

Если при установлении соединения ABR не задаются значения максимальной и минимальной скоростей, то по умолчанию считается, что PCR совпадает со скоростью линии доступа станции к сети, а скорость MCR считается равной нулю.

Трафик соединения категории ABR получает гарантированное качество услуг в отношении доли потерянных ячеек и пропускной способности. Что касается задержек передачи ячеек, то хотя сеть и старается свести их к минимуму, но гарантий по этому параметру не дает. Следовательно, служба ABR предназначена не для приложений реального времени, а для приложений, в которых поток данных не очень чувствителен к задержкам в передаче.

При передаче трафика CBR, VBR и UBR явное управление перегрузками в сети отсутствует. Вместо этого используется механизм отбрасывания ячеек-нарушителей, а узлы, пользующиеся услугами CBR и VBR, стараются не нарушать условия контракта под угрозой потери ячеек, поэтому они обычно не задействуют дополнительную пропускную способность, даже если она в данный момент доступна в сети.

Служба ABR позволяет воспользоваться резервами пропускной способности сети, так как сообщает конечному узлу о наличии в данный момент избыточной пропускной способности с помощью механизма обратной связи. Этот же механизм может помочь службе ABR снизить скорость передачи данных конечным узлом в сеть (вплоть до минимального значения MCR), если сеть испытывает перегрузку.

Узел, пользующийся услугами ABR, должен периодически посылать в сеть наряду с ячейками данных специальные **служебные ячейки администрирования ресурсов** (Resource Management, RM). RM-ячейки, которые узел отправляет вдоль потока данных, называются ячейками прямого администрирования ресурсов (Forward Resource Management, FRM), а ячейки, которые идут в обратном по отношению к потоку данных направлении, называются ячейками обратного администрирования ресурсов (Backward Resource Management, BRM).

Существует несколько **петель обратной связи**. Самая простая петля обратной связи — между конечными станциями. При ее наличии коммутатор сети извещает конечную станцию о перегрузке с помощью специального флага в поле прямого управления перегрузками (флаг EFCI) ячейки данных, переносимой протоколом ATM. Затем конечная станция посылает через сеть сообщение, содержащееся в специальной ячейке BRM, говоря станции-отправителю о необходимости снизить скорость посылки ячеек в сеть.

В этом способе конечная станция несет основную ответственность за управление потоком, а коммутаторы играют пассивную роль в петле обратной связи, только уведомляя станцию-отправитель о перегрузке.

Такой простой способ имеет несколько очевидных недостатков. Конечная станция не узнает из BRM-сообщения, насколько нужно снизить скорость передачи данных в сеть. Поэтому она просто снизит скорость до величины MCR, хотя, возможно, это и не обязательно. Кроме того, при большой протяженности сети коммутаторы вынуждены продолжать буферизовать данные все время, пока уведомление о перегрузке будет путешествовать по сети, а для глобальных сетей это время может быть достаточно большим, и буферы могут переполниться, так что требуемый эффект достигнут не будет.

Разработаны и более сложные схемы управления потоком, в которых коммутаторы играют более активную роль, а узел-отправитель точнее узнает о возможной в данный момент скорости отправки данных в сеть.

В первой схеме узел-источник посылает в FRM-ячейке явное значение скорости передачи данных в сеть, которую он хотел бы поддерживать в данное время. Каждый коммутатор, через который проходит по виртуальному пути это сообщение, может снизить запрашиваемую скорость до некоторой величины, которую он способен поддерживать в соответствии с

имеющимися у него свободными ресурсами (или оставить запрашиваемую скорость без изменения). Узел назначения, получив FRM-ячейку, отправляет ее обратно в виде BRM-ячейки, причем он тоже может снизить запрашиваемую скорость. Получив ответ в BRM-ячейке, узел-источник точно узнает, какая скорость отправки ячеек в сеть для него доступна.

Во второй схеме каждый коммутатор сети может работать и как источник, и как приемник. Как узел-источник он может сам генерировать FRM-ячейки и отправлять их по имеющимся виртуальным каналам. Как узел-приемник он может отправлять на основе получаемых FRM-ячеек BRM-ячейки в обратном направлении. Такая схема является более быстродействующей и полезной в протяженных территориальных сетях.

Как видно из описания, служба ABR предназначена не только для прямого поддержания требований к обслуживанию конкретного виртуального соединения, но и для более рационального распределения ресурсов сети между ее абонентами, что в конечном итоге также приводит к повышению качества обслуживания всех абонентов сети.

Коммутаторы сети ATM используют различные механизмы для поддержания требуемого качества обслуживания. Помимо описанных в стандартах комитета ITU-T и форума ATM механизмов заключения соглашения на основе параметров трафика и QoS с последующим отбрасыванием ячеек, не удовлетворяющих условиям соглашения, практически все производители оборудования ATM реализуют в своих коммутаторах несколько очередей ячеек, обслуживаемых с разными приоритетами.

Стратегия приоритетного обслуживания трафика основана на категориях услуг каждого виртуального соединения. До принятия спецификации ABR в большинстве коммутаторов ATM была реализована простая одноуровневая схема обслуживания, которая давала трафику CBR первый приоритет, трафику VBR — второй, а трафику UBR — третий. При такой схеме комбинация услуг CBR и VBR может потенциально заморозить трафик, обслуживаемый другим классом служб. Такая схема не будет правильно работать с трафиком ABR, так как не обеспечит его требования к минимальной скорости передачи ячеек. Для обеспечения этого требования должна быть выделена некоторая гарантированная полоса пропускания.

Чтобы поддерживать службу ABR, коммутаторы ATM должны реализовать двухуровневую схему обслуживания, которая бы удовлетворяла требованиям CBR, VBR и ABR. По этой схеме коммутатор предоставляет некоторую часть своей пропускной способности каждому классу служб. Трафик CBR получает часть пропускной способности, необходимую для поддержания пиковой скорости (PCR), трафик VBR получает часть пропускной способности, необходимую для поддержания средней скорости (SCR), а трафик ABR получает часть пропускной способности, достаточную для обеспечения требования минимальной скорости ячеек (MCR). Это гарантирует, что каждое соединение будет работать без потерь ячеек и не будет доставлять ABR-ячейки за счет трафика CBR или VBR. На втором уровне этого алгоритма трафик CBR и VBR может забрать всю оставшуюся пропускную способность сети, если это необходимо, так как соединения ABR уже получили свою минимальную пропускную способность, которая им гарантировалась.

Отдельной задачей, которую нужно решать для поддержки корректной работы описанных служб и, соответственно, обеспечения заданного уровня QoS для всех классов трафика, является оптимизация работы сети ATM методами инжиниринга трафика (TE). Использование в сетях ATM (как и в сетях Frame Relay) техники виртуальных каналов создает хорошие предпосылки для решения задачи TE. В технологии Frame Relay каких-либо автоматизированных процедур для динамического выбора маршрутов следования виртуальных путей в целях обеспечения сбалансированной загрузки ресурсов сети пока не существует. Вся работа по оптимизации маршрутов должна выполняться предварительно, с помощью каких-либо внешних программных систем моделирования или оптимизации сети, а затем вручную реализовываться за счет установления постоянных виртуальных каналов в соответствии с выбранными маршрутами.

В сетях ATM в выборе маршрута для виртуальных каналов и путей может участвовать протокол маршрутизации PNNI, который для прокладки новых виртуальных каналов учитывает не только номинальную, но и доступную в данный момент пропускную способность.