



УДК 683. 519

**АНАЛИЗ МЕТОДОВ УПРАВЛЕНИЯ ТРАФИКОМ ABR В  
СЕТЯХ АТМ С ПРИМЕНЕНИЕМ ИМИТАЦИОННОГО  
МОДЕЛИРОВАНИЯ**

**Ю.П. ЗАЙЧЕНКО, МУХАММЕД-АЛИ АЗЗАМ ХАМУДИ**

Предложен новый метод управления трафиком ABR в сетях АТМ на основе построения модели, прогнозирующей загрузку буферов коммутаторов сети АТМ. Разработана имитационная модель корпоративной сети АТМ университета. Выполнено моделирование ее работы с различными методами управления трафиком, включая предложенный. Проведен анализ их эффективности.

**ВВЕДЕНИЕ**

Одной из проблем проектирования компьютерных сетей с технологией АТМ является анализ и выбор систем управления трафиками категорий CBR, VBR и ABR. Разработка и внедрение системы управления трафиками позволяет обеспечить необходимые показатели качества передачи, наилучшим образом управлять имеющимися ресурсами при установлении новых соединений, что, в конечном счете, влияет на эффективность функционирования сети, ее производительность и качество обслуживания абонентов. Современная компьютерная сеть АТМ — очень сложная система, в которой одновременно передаются разные виды информации (аудио, видео и данные с различными показателями качества (QoS)). На функционирование системы непосредственно влияет организация управления трафиками CBR, VBR и ABR, управление работой коммутаторов, в частности способ организации буферов и метод управления очередями. Поскольку построение аналитических моделей для комплексного исследования влияния этих факторов на QoS невозможно, единственный способ решения указанной проблемы — разработка имитационной модели.

Цель данной работы — описание детализированной имитационной модели (ИМ) корпоративной сети АТМ, исследование и анализ на основе ИМ методов управления потоками и способов управления буферами коммутаторов АТМ, включая предложенный новый метод управления на основе прогнозирования.

В работе описана ИМ сети АТМ. Рассматривается корпоративная компьютерная сеть АТМ, состоящая из хост-ЭВМ, конечного оборудования

пользователей — АП, узлов связи (УС) — коммутаторов ATM и каналов связи (КС) соответствующей пропускной способности.

Конечное оборудование и хост-ЭВМ являются генераторами входных пакетов-трафиков разных категорий сервиса передачи информации:

- с постоянной скоростью (аудио и видеоинформация) — CBR;
- с переменной скоростью (сжатых видео и аудио) — VBR;
- с доступной информацией данных — ABR.

Каждый УС<sub>*i*</sub> характеризуется векторами  $H_i^{CBR}$ ,  $H_i^{VBR}$ , и  $H_i^{ABR}$ , которые отвечают категориям трафика CBR, VBR и ABR.

Особенности технологии ATM — введение следующих показателей качества соответствующих категорий: CTD — средняя задержка ячеек, CDV — вариация средней задержки, CLR — вероятность потери ячеек.

Каждой категории сервиса соответствуют свои значения *QoS*. Для передачи трафика CBR используется постоянная полоса в канале, независимо от того, идет передача информации или нет. Остаток полосы выделяется для смеси трафиков VBR и ABR. Трафик VBR является более приоритетным и обслуживается в первую очередь, а остаток полосы канала занимает трафик ABR.

## 1. УПРАВЛЕНИЕ ТРАФИКОМ

Важная функция коммутаторов ATM и систем управления сетью — управление трафиком. Управление трафиком ABR осуществляется на основе механизма обратной связи, с помощью которого контролируется скорость передачи источника в соответствии с изменением характеристик сети. Механизм обратной связи реализуется специальными ячейками, которые называются RM-ячейками или ячейками управления ресурсами.

Существует несколько методов и подходов к управлению трафиком ABR. Наиболее известны подходы rate-based и credit-based [3].

### 1.1.1. Кредитная схема

Эта схема предусматривает управление потоком на каждом участке каждого виртуального соединения. На принимающих портах коммутаторов ATM под конкретное виртуальное соединение резервируются определенные объемы буферной памяти. Таким образом источник трафика как бы получает кредит на передачу определенного числа ячеек, которую он может осуществить, не дожидаясь какой-либо управляющей информации. Число ячеек определяется объемами выделенных буферов. При возникновении перегрузки и переполнении буферной памяти одного из коммутаторов он посылает управляющее сообщение на предыдущий коммутатор с требованием прекратить передачу данных по «перегруженному» соединению. Тот прекращает передачу и начинает накапливать поступающие данные в своем буфере. Это продолжается до тех пор, пока не исчезнет перегрузка на первом участке, и коммутатор не пришлет разрешение на возобновление передачи. Отметим, что при такой схеме управления перегрузками для каждого виртуального соединения должен быть выделен индивидуальный буфер.

Кредитная схема управления перегрузками может предотвратить потерю ячеек, что очень важно для работы некоторых приложений. Она позволяет максимально эффективно использовать полосу пропускания канала и, кроме того, дает возможность проходящим по одному физическому каналу виртуальным соединениям работать на разных скоростях.

Рассмотренная схема не получила широкого распространения, так как ее применение требует серьезной доработки оборудования АТМ для поддержки индивидуальных буферов и сложных алгоритмов динамического расчета буферного пространства при установлении коммутируемых виртуальных соединений (SVC). Кредитная схема не была одобрена Форумом АТМ и не вошла в стандарты по управлению трафиком.

### 1.1.2. Скоростные схемы

Существует несколько скоростных схем управления перегрузками, и все они предусматривают использование обратной связи для информирования источника о том, с какой скоростью в данный момент он может передавать ячейки по каждому виртуальному соединению [3, 5].

#### **FECN и BECN**

FECN (Forward Explicit Congestion Notification) — одна из скоростных схем управления перегрузками, использующая отрицательную обратную связь. Когда коммутатор АТМ испытывает перегрузку, он выставляет в проходящих через него ячейках бит EFCI (находится в поле PFI заголовка ячейки), информируя тем самым приемник о перегрузке по конкретному виртуальному соединению. Получив сообщение о перегрузке, приемник направляет эту информацию источнику трафика.

В свою очередь, источник принимает решение о снижении скорости передачи по «перегруженному» виртуальному соединению.

Рассмотрим принцип работы схемы FECN. Испытывающий перегрузку коммутатор устанавливает во всех ячейках, проходящих по виртуальному соединению VC1, биты EFCI, равные единице, и тем самым информирует приемник о перегрузке по этому соединению. Получив данную информацию, приемник, чтобы проинформировать источник трафика о перегрузке, высылает ему специализированные ячейки управления ресурсами RM (Resource Management). Получив ячейки RM, источник начинает снижать скорость передачи по соединению VC1.

По аналогичному принципу работает и схема BECN (Backward Explicit Congestion Notification), но в ней предусмотрено, что информацию о перегрузке направляет источнику трафика непосредственно сам испытывающий ее коммутатор (рис.1). Очевидным преимуществом схемы BECN является более быстрая и, следовательно, более эффективная реакция на перегрузку. Но, с другой стороны, при использовании этой схемы каждый транзитный коммутатор АТМ должен уметь генерировать ячейки RM и вставлять их в проходящий поток данных.

В схемах FECN и BECN коммутатор АТМ считается перегруженным, если очередь ячеек на обслуживание (коммутацию) превысит определенное пороговое значение. Получив информацию о перегрузке, источник должен начать снижение скорости передачи трафика, направляемого через перегру-

женный участок сети. Это снижение происходит до тех пор, пока продолжает поступать информация о перегрузке. Как только появление ячеек RM с информацией о перегрузке прекратится, и они не будут поступать в течение определенного интервала времени, источник может начать увеличение скорости передачи вплоть до ее максимального значения (PCR).

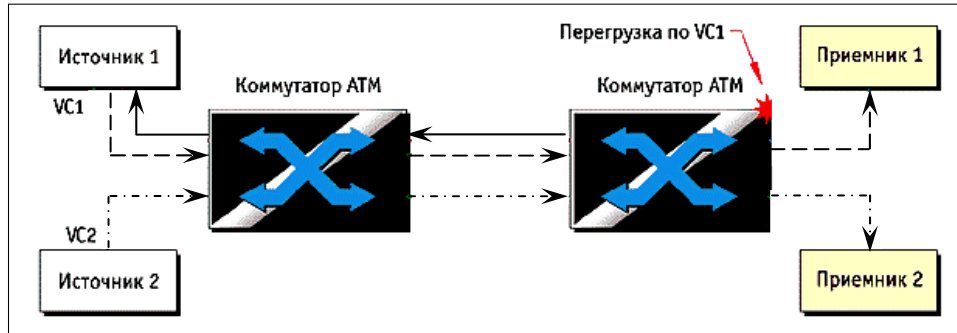


Рис. 1. Принципы работы схемы BECN: — ячейки RM; ---- поток данных с EFCI=1; ..... с EFCI=0

Если ячейки RM, направляющиеся к источнику трафика с информацией о перегрузке, сами попадают в перегруженный поток, то, вполне возможно, они никогда не смогут доставить эту информацию. Кроме того, чрезмерно перегруженный коммутатор ATM может быть не в состоянии не только обслуживать транзитный трафик, но и генерировать ячейки RM. В обеих ситуациях сеть будет лишена информации о перегрузках. Это может усугубить положение дел в сети, ведь при отсутствии ячеек RM источник трафика начнет увеличивать скорость передачи, приближая ее к максимальной (PCR).

Таким образом, при использовании схем FECN и BECN появляется потенциальная возможность возникновения аварийной ситуации. Поэтому Форум ATM разработал более устойчивые схемы, например, схему, основанную на алгоритме пропорционального управления скоростью (Proportional Rate Control Algorithm — PRCA) [3,6,8].

## 1.2. Алгоритм EPRCA

В отличие от предыдущих схем алгоритм EPRCA основан на принципе положительной обратной связи, что позволяет избежать описанных выше проблем. При использовании EPRCA источник трафика увеличивает скорость передачи только тогда, когда получает разрешение на это от приемника. В противном случае, т. е. при отсутствии такого разрешения, источник должен последовательно снижать скорость пропорционально числу передаваемых ячеек.

Алгоритм EPRCA (Enhanced PRCA) работает следующим образом.

Источник трафика посылает все информационные ячейки с битом EFCI, равным нулю. Через каждые  $N$  таких ячеек источник посылает ячейку RM, содержащую значения желаемой (обычно равна PCR) и текущей (ACR) скоростей передачи (рис. 2). Испытывающий перегрузку коммутатор ATM подсчитывает собственное значение скорости ACR ( $ACR^*$ ), которое зависит от того, какой объем трафика он может обработать в данный момент, и вставляет это значение в проходящую через него ячейку RM.

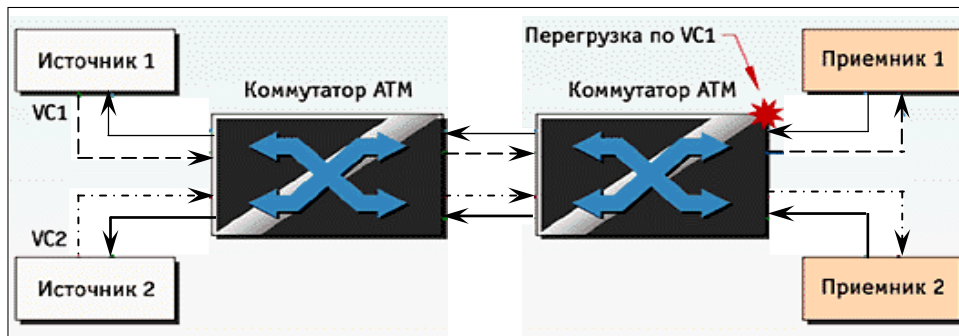


Рис. 2. Принципы работы схемы ERPCA: - - - - ячейки RM (PCR, ACR);  
 ——— данные и ячейки RM (PCR, ACR); ······ ячейки RM (PCR, ACR\*);  
 ——— данные и ячейки RM (PCR, ACR\*)

Получив эту ячейку, приемник отправляет ее обратно источнику, а тот корректирует свою скорость в соответствии с новым значением скорости ACR\*. Таким образом, проходя по кругу, ячейка RM «собирает» информацию о наличии ресурсов со всех промежуточных коммутаторов и возвращается обратно к источнику со значением наименьшей доступной скорости для данного виртуального канала.

Алгоритм EPRCA, как и алгоритм PRCA, обязывает источник трафика снижать скорость передачи при отсутствии ячеек RM, что избавляет сеть от возможных перегрузок. Он не имеет недостатков PRCA, так как скорость передачи трафика от источника не зависит от числа ячеек с EFCI=1, а задается в явном виде при передаче управляющей ячейки RM. Алгоритм EPRCA включен в принятый Форумом ATM стандарт ATM Traffic Management [3, 8, 9].

### 1.3. Метод управления соединениями класса ABR на основе прогнозирующей модели

Под управлением трафиком ABR будем подразумевать управление темпом загрузки–выгрузки буферов коммутаторов ATM посредством изменения скоростей передачи информации, составляющей трафик ABR.

Для пояснения введем некоторые определения, сформируем условие применимости и цели управления предлагаемого метода.

Под соединением будем понимать виртуальный канал, установленный между узлом-отправителем и узлом-адресатом. Под трактом — участок сети между ее коммутаторами. Под интегральным потоком (в отличие от просто потока) — поток, включающий в себя трафики ABR и VBR от нескольких источников, обслуживаемых одним коммутатором.

Цель управления состоит в распределении скоростей передачи источниками трафика ABR таким образом, чтобы для каждого  $j$ -го источника обеспечить скорость  $V^j$ , удовлетворяющую соотношению  $V_{\min}^j \leq V^j \leq V_{\max}^j$ , где  $V_{\min}^j$  и  $V_{\max}^j$  — соответственно минимальная и максимальная скорости передачи информации, которые сеть гарантирует  $j$ -му источнику.

При этом суммарная скорость всех потоков, проходящих через тракт, не должна превышать пропускной способности тракта. Распределение ско-

ростей должно быть справедливым для всех потоков, а производительность сети должна быть максимальной.

Будем считать, что коммутаторы представляют собой устройства, имеющие возможность хранить в дополнительной области памяти статистические значения определенных далее параметров, а также рассчитывать прогнозируемые значения с помощью рекуррентного МНК.

К числу переменных, характеризующих состояние сети и необходимых для решения задачи прогнозирования и управления применительно к  $j$ -му коммутатору, относятся:

- скорость каждого  $i$ -го входного потока информации сервисов VBR и ABR, обрабатываемого соответствующим коммутатором в текущий момент времени  $V_i^t$ ;

- суммарная скорость потоков информации сервисов VBR и ABR, обрабатываемых коммутатором в текущий момент времени  $V^t = \sum_{i=1}^n V_i^t$ ;

- приращение суммарной скорости потоков информации сервисов VBR и ABR за текущий интервал времени  $\Delta V^t = V^t - V^{t-1}$ ;

- загрузка буфера коммутатора на текущий момент времени  $h^t$ , %;

- приращение загрузки буфера коммутатора за текущий интервал времени.

На основе этих статистических данных, собранных за определенный период времени каждым коммутатором, синтезируются прогнозирующие модели рекуррентным МНК, где прогнозируемыми переменными служат приращения загрузки буферов коммутаторов (%) на момент времени, когда управляющая ячейка прибывает к источнику. Если для какого-либо буфера спрогнозированные приращения загрузки сигнализируют об их перегрузке в этот момент времени, то рассчитывается величина снижения скорости суммарных трафиков ABR применительно к данному коммутатору с использованием тех же прогнозирующих моделей.

Прогноз для каждого коммутатора осуществляется по модели, имеющей структуру

$$\Delta h^{t+1} = a_0 + a_1 V^t + a_2 \Delta V^t + a_3 \Delta h^t. \quad (1)$$

Выбор линейного вида зависимости в  $\Delta h = f(\Delta v, v)$  в модели (1) определяется самой «физикой» процесса поступления ячеек в буфер. Число этих ячеек пропорционально скорости передачи  $V$ , поэтому изменение размера очереди  $\Delta h$  (в первом приближении) пропорционально изменению текущей скорости  $\Delta V$ , а величина снижения суммарных трафиков ABR в случае прогнозирования перегрузки буфера коммутатора за интервал времени рассчитывается по формуле

$$\Delta V^t = \frac{a_0}{a_2} + \frac{1}{a_2} \Delta h^{t+1} - \frac{a_1}{a_2} V^t - \frac{a_3}{a_2} \Delta h^t. \quad (2)$$

Если  $\Delta V^t > 0$ , то нам необходимо повысить суммарные скорости трафиков ABR, если  $\Delta V^t < 0$  — то понизить.

Выбирая  $\Delta h^{t+1}$ , можно вычислить снижение суммарной скорости трафиков ABR, которое приведет к снижению загрузки буфера коммутатора за интервал времени на эту же величину  $\Delta h^{t+1}$ .

Величину  $\Delta h^{t+1}$  будем выбирать таким образом: понизим (условно) все скорости потоков сервиса ABR до скоростей  $V_{i,\min}$  и вычислим понижение скоростей  $\Delta w_i^t = V_i^t - V_{i,\min}$ . Найдем суммарное понижение скорости

$$\Delta w^t = \sum_{i=1}^n \Delta w_i^t.$$

Вычислим значение  $\Delta h^{t+1}$  по выражению (1), при этом вместо  $\Delta V^t$  подставим  $\Delta w^t$ , вместо  $\Delta h^t$  подставим 0, а значение  $V^t$  оставим без изменений.

Выберем значение  $\Delta h^{t+1}$  так, чтобы оно было меньше вычисленного по выражению (1).

Подставив выбранное значение  $\Delta h^{t+1}$  в выражение (2), определим  $\Delta V^t$  — величину необходимого понижения суммарной скорости потоков сервиса ABR.

Справедливо понизим все скорости потоков сервиса ABR.

1. Вычислим  $d = \frac{\Delta V^t}{m}$ , где  $m$  — количество соединений ABR, проходящих через узел.
2. Будем пробовать понижать скорости всех соединений ABR на величину  $d$ . Если для некоторых потоков это не удастся из-за того, что пониженная таким образом скорость станет меньше  $V_{i,\min}$ , то скорости таких соединений следует понизить только до  $V_{i,\min}$ , а значения, на которые не удалось понизить скорость, — суммировать во временную переменную  $\Delta V^*$ .
3. Если  $\Delta V^* = 0$ , или все скорости соединений уже понижены до  $V_{i,\min}$ , то работа алгоритма понижения скоростей заканчивается, в противном случае  $\Delta V^*$  переобозначим в  $\Delta V^t$  и переходим к п. 1.

Напомним, что все наши рассуждения относятся к одному конкретному тракту, коммутатору и его буферу, а также, разумеется, к соответствующим потокам.

Поскольку в сети в общем случае несколько коммутаторов, то алгоритм управления сводится к корректировке скорости, указанной в обратной RM-ячейке, каждым коммутатором, который принадлежит участку сети между источником и адресатом. Значение скорости прогнозируется на время  $\Delta t$ , необходимое для преодоления обратной RM-ячейкой участка пути от текущего коммутатора до источника. При прохождении ячейки через коммутаторы выбирается минимальная из всех спрогнозированных скоростей.

Процесс управления скоростями передачи информации источниками приводит к загрузке и выгрузке буферов коммутаторов. Алгоритм приращенния суммарной скорости ABR-соединений аналогичен алгоритму понижения.

1. Вычислим  $V^t$  — суммарную скорость трафиков ABR и VBR.
2. Вычислим  $R = P - V^t$  — разность между пропускной способностью тракта передачи информации  $P$  и значением  $V$ .

3. Вычислим  $d = \frac{R}{m}$ , где  $m$  — количество потоков ABR.

4. Будем пробовать повышать скорости потоков ABR на величину  $d$ . Если для некоторых потоков это не удастся, потому что такая скорость станет большей  $V_{i,\max}$ , то скорости таких соединений повысить только до  $V_{i,\max}$ , а значения, на которые не удалось повысить скорость, — суммировать во временную переменную  $R^*$ .

5. Если  $R = 0$ , или все скорости соединений уже повышены до  $V_{i,\max}$ , то алгоритм повышения скоростей останавливается, в противном случае  $R^*$  переобозначим в  $R$  и переходим к п. 2.

Управление перегрузкой в сети ATM по комбинированному принципу обеспечивает в большинстве ситуаций управление по предварению и только в случае ошибок прогноза — по отклонению, что в конечном итоге приводит к более высокому качеству управления.

## 2. ОПИСАНИЕ ИМИТАЦИОННОЙ МОДЕЛИ

### 2.1. Входные данные

Топология сети:

- $n_u$  — число узлов;  $n_k$  — каналов,  $n_{\text{CBR}}$  — генераторов CBR-трафиков;  $n_{\text{VBR}}$  — генераторов VBR-трафиков.

Параметры объектов, участвующих в имитационной модели:

- для узла:  
идентификатор, объем входящего и исходящего буферов;
- для канала:  
идентификатор, идентификаторы двух узлов, связанных данным каналом, пропускная способность канала;
- для соединений класса CBR:  
идентификатор, идентификаторы узла-отправителя и узла-адресата, скорость передачи информации (яч/с).
- для соединений класса VBR:  
идентификатор, идентификаторы узла-отправителя и узла-адресата, минимальная скорость передачи информации (яч/с), средняя скорость передачи (яч/с), максимальная скорость передачи (яч/с).

Параметры сети:



максимально допустимое количество соединений, проходящих через коммутатор (максимальное количество строчек в таблице маршрутизации для узла);  
длина пакета (яч).

Параметры для соединений ABR:

максимальный объем данных для передачи (пакетов);  
максимальная скорость передачи (яч/с);  
средняя скорость передачи (яч/с);  
интенсивность попыток генерации новых ABR-соединений (1/1000 с);  
величина тайм-аута (если соединение не получило первую обратную управляющую ячейку, оно будет удалено из сети).

Для метода прогнозирования переполнений задаются такие параметры:  
объем выборки;  
порог (в процентах от объема буфера), при превышении спрогнозированным значением которого коммутатором будут генерироваться out-of-rate управляющие ячейки.

## 2.2. Имитационный алгоритм

В качестве метода имитации избран итерационный метод с учетом особых состояний. Его идея состоит в том, что при создании каналов и генераторов трафиков определяется время, когда они могут работать: для канала — время передачи ячейки из одного узла в другой, а для источника трафика — передачи ячейки в буфер узла (источника). Такой подход дает возможность учесть ситуации, когда несколько каналов или источников передают данные одновременно [1, 2].

В процессе имитации каждую сотую долю секунды (момент  $\Delta t$ ) срабатывает таймер, определяющий состояние сети. Соответственно, источники трафиков, которым разрешено в этот момент передавать, генерируют исходные ячейки.

За базовую принята модель, в которой каналы используют интервал времени, равный 1/100 с. Канал передает ячейки во входной буфер коммутатора, и он после этого выполняет обработку ячеек. Ячейки, не поместившиеся во входной буфер, отбрасываются.

### Описание $k$ -й итерации

*Шаг 1.* Срабатывает таймер. Текущее время системы —  $t_{\text{тек}}$ .

*Шаг 2.* Из множества  $U_{\text{CBR}}$  источников CBR-трафиков выбираются те  $k$ , время передачи ячеек  $t$  которых равняется  $t_{\text{тек}}$ . Каждый источник генерирует ячейки в 1/10 заданной скорости и передает их в буфер узла-источника.

*Шаг 3.* Выполняется аналогично шагу 2, но для VBR-трафиков. В отличие от CBR-трафика количество сгенерированных ячеек выбирается случайно по равномерному закону распределения в пределах минимальной и максимальной заявленных скоростей. Поскольку при установлении трафик-контракта между пользователем категории VBR и сетью определяются параметры трафика (минимальная скорость MCR и пиковая скорость передачи

ячеек PCR, которую пользователь не может превышать), то в текущий момент времени пользователь может передавать с любой скоростью  $V$  в интервале  $[MCR, PCR]$ , а так как поведение пользователя априорно неизвестно, то принимается равномерное распределение выбора пользователем скорости из данного интервала (в соответствии с известным принципом индетерминизма Лапласа).

*Шаг 4.* Аналогично выбирается множество источников ABR-трафиков. Количество сгенерированных ячеек зависит от текущей разрешенной скорости.

*Шаг 5.* Из  $n_k$  каналов выбираются те  $k$  каналов, время освобождения которых  $t_1$  или  $t_2$  равняется  $t_{\text{тек}}$ .

*Шаг 6.* Для каждого из  $k$  каналов по каждому свободному направлению выполняется следующее:

6.1. Из буфера исходного узла выбираются те ячейки, которые по таблице маршрутизации должны быть переданы по данному каналу, после чего они перемещаются в буфер узла-адресата. Если ячейка является обратной RM-ячейкой, то выполняется корректировка скорости, с которой соединению разрешено передавать в данный момент в зависимости от выбранного метода управления.

Проверяется, подлежит ли ячейка сбрасыванию (имеется ли запись в таблице сбрасываемых ячеек узла-адресата о ячейках с такими параметрами VCI, VPI и ID пакета) и переполнен ли буфер на узле-адресате.

6.1.1. Когда одно из условий выполняется, ячейка отбрасывается и счетчик отброшенных ячеек на узле-адресате увеличивается на 1. Если ячейка отброшена из-за переполненности буфера, а не в ходе проверки необходимости сброса остатков пакета, то по заголовку ячейки выясняются ее параметры VCI, VPI и номер пакета, а затем заносятся в таблицу сбрасываемых ячеек. Таким образом данные о пакете, остаток которого необходимо сбросить, сохраняются в специальной таблице коммутатора, на котором состоялся сброс. Поскольку прием всех поступивших ячеек и пакетов подтверждается получателем соответствующей квитанцией, то в случае сброса и потери ячейки средствами протокола TCP обеспечивается ее повторная передача. Но это уже не влияет на значение показателя CLR (интенсивности потери ячеек), оцениваемого с помощью имитационного моделирования.

6.1.2. В противном случае проверяется, нет ли на узле-адресате записи о том, что ячейки предыдущего пакета с такими же индикаторами VCI и VPI должны быть сброшены. При положительном результате эта запись из таблицы удаляется.

*Шаг 7.* Для каждого узла, входной буфер которого является непустым, выполняются такие операции:

7.1. Анализируются все ячейки во входном буфере. Если, согласно таблице маршрутизации, ячейка прибыла к конечному адресату, то она остается во входном буфере. В противном случае изменяются поля VCI и VPI в соответствии с локальной таблицей маршрутизации узла, и ячейка направляется в исходящий буфер.

7.2. Ячейки, адресованные этому узлу и оставшиеся во входящем буфере, обрабатываются следующим образом:

7.2.1. Если ячейка является CBR или VBR-ячейкой, то переходим к шагу 7.2.6.

7.2.2. Иначе проверяется, не является ли эта ячейка RM. Если так, то немедленно генерируется обратная ячейка, которая помещается в исходящий буфер данного узла, и переходим на шаг 7.2.6.

7.2.3. В противном случае проверяется, не является ли эта ячейка обратной RM-ячейкой. В этом случае соответствующий источник ABR-трафика вносит коррективы относительно дальнейшей скорости передачи в соответствии с содержанием поля ER в этой ячейке.

7.2.4. Если ячейка оказалась обычной (т. е. не RM), то проверяются номера пакета и ячейки в пакете у последней принятой этим соединением ячейки. Если предыдущий пакет не был получен полностью, то источнику отсылается служебная ячейка, несущая в себе информацию о том, что весь пакет необходимо отправить заново.

7.2.5. Если ячейка оказалась первой обратной RM-ячейкой, то в зависимости от указанных в ней допустимых параметров скорости ожидающему разрешению на передачу ячеек соединению класса ABR устанавливается разрешение или запрет на начало передачи пакетов, и происходит перевод соединения в активное состояние. В случае запрета увеличивается индикатор числа отказанных соединений ABR, статус текущего соединения изменяется на RTD-ready to die.

7.2.6. Если получена последняя ячейка пакета, то индикатор полученных пакетов увеличивается на 1.

7.2.7. Если получены все пакеты, предусмотренные соединением к передаче, то увеличивается счетчик успешно завершенных соединений, а статус этого соединения изменяется на RTD.

7.2.8. Из заголовка полученной ячейки считывается поле «время создания» и сравнивается с текущим состоянием системы. Считаём время передачи ячейки:

$$T_{\text{пер}} = T_{\text{тек}} - T_{\text{созд}}, \quad (3)$$

где  $T_{\text{пер}}$  — промежуток времени за который ячейка была передана адресату;  $T_{\text{тек}}$  — текущий момент времени, в котором находится система;  $T_{\text{созд}}$  — момент времени, в котором система находилась при создании ячейки.

7.2.9. Таким образом получаем время передачи данной ячейки. Далее рассчитываем показатель средней задержки для текущего соединения

$$T_{\text{ср}}^{\text{н}} = \frac{T_{\text{ср}} * n_{\text{яч}} + T_{\text{пер}}}{n_{\text{яч}} + 1}, \quad (4)$$

где  $T_{\text{ср}}^{\text{н}}$  — новое значение показателя средней задержки ячеек текущего соединения;  $T_{\text{ср}}$  — показатель средней задержки ячеек текущего соединения;  $n_{\text{яч}}$  — количество успешно переданных ячеек;  $T_{\text{пер}}$  — время передачи данной ячейки. После этого количество принятых ячеек увеличивается на 1.

*Шаг 8.* Выбираются все соединения класса ABR, статус которых равняется RTD, после чего они все уничтожаются в сети. Под уничтожением под-

разумеается удаление всех оставшихся служебных ячеек, принадлежащих этому виртуальному соединению, а также информации о нем из таблиц коммутаторов, участвовавших в передаче информации.

Важной функцией управления трафиком в сети ATM является управление обслуживанием ячеек в буферах коммутаторов. Для этого используется дисциплина приоритетного обслуживания с соответствующими приоритетами. Наивысшим приоритетом пользуются ячейки CBR, затем ячейки трафика VBR. Наименьший приоритет имеют ячейки трафика ABR. Для обслуживания ячеек в памяти коммутатора ATM создаются три очереди: для категорий сервиса CBR, VBR и ABR. Внутри очереди используется дисциплина FIFO. Возможны следующие варианты организации очереди:

- 1) отдельные очереди для трех категорий сервиса;
- 2) для каждого виртуального соединения;
- 3) общая очередь.

### Алгоритм размещения ячейки в буфер

Определяется тип буфера: распределенный или общий.

При общем режиме буферизации проверяется, есть ли свободное место в буфере, и если так, то в его конец добавляется ячейка.

При динамическом режиме буферизации с перераспределением рассматривается, есть ли в буфере ABR свободное место. В случае утвердительного ответа в объеме буфера для ABR трафиков выделяется место для сохранения ячейки с высшим приоритетом (CBR или VBR). После освобождения эта часть буфера возвращается в буфер для ABR-соединений.

В случае использования динамического режима буферизации коммутатор отправляет ячейки согласно приоритетам по классам трафика.

## 2.3. Реализация алгоритмов управления

### 2.3.1. Rate-based

Управление осуществляется редактированием соответствующего поля обратных RM-ячеек, которые проходят весь путь от адресата виртуального соединения к источнику. На каждом коммутаторе, входящем в тракт соответствующего виртуального соединения, рассчитывается допустимая скорость, после чего выбирается минимальное значение между рассчитанным значением и прописанным в обратной RM-ячейке.

$$V_j = \frac{1}{n_{m,i}} \left( S_i (1 - K_{\text{исп CBR+ABR}}^i) \right), \quad (5)$$

где  $V_j$  — скорость, с которой разрешено передавать  $j$ -му ABR-трафику;  $S_i$  — пропускная способность текущего канала;  $K_{\text{исп CBR+ABR}}^i$  — коэффициент использования  $i$ -го канала CBR и VBR трафиками;  $n_{m,i}$  — количество виртуальных соединений класса ABR, проходящих через коммутатор  $m$  и канал  $i$  в сторону передачи данных текущего виртуального соединения.

### 2.3.2. Управление на основе прогнозирования переполнения буферов коммутаторов

В отличие от предыдущего метода, где управление производилось с помощью обратных RM-ячеек, которые отсылаются адресатом в ответ на пришедшую прямую RM-ячейку (соответственно, компоненты сети никак не могут быть инициаторами регулировки скорости соединений), в этом методе коммутаторы могут самостоятельно генерировать обратные RM-ячейки, в случае, если коммутатором будет спрогнозировано переполнение буфера.

Также отличием от предыдущих двух методов является регулирование скоростей передачи не по текущим значениям, а с упреждением, используя для прогнозирования МНК.

Как указывалось в разделе 2.2, прогнозирующая модель, используемая в этом методе, выглядит таким образом:

$$\Delta h^{t+1} = a_0 + a_1 V^t + a_2 \Delta V^t + a_3 \Delta h^t. \quad (6)$$

Соответственно, расчет величины изменения суммарных трафиков ABR, проходящих через узел, осуществляется по формуле (2).

Время  $t$ , на которое делается прогноз, вычисляется, исходя из длины пути от коммутатора до адресата управляющей ячейки.

В случае, когда спрогнозированное коммутатором значение превышает заданный порог, коммутатор самостоятельно генерирует управляющие ячейки, передающие скорости, с которыми соединениям разрешено передавать данные.

## 3. РЕЗУЛЬТАТЫ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ

Цель данной работы — не только реализация имитационной модели, но и проведение сравнительного анализа реализованных методов управления, способов организации буферов, а также анализ различных топологий сети.

Моделируемая сеть имеет топологию, идентичную топологии корпоративной сети НГУУ «КПИ» (рис. 3). Номера узлов соответствуют учебным корпусам и общежитиям университета.

В табл. 1 приведены результаты имитационного моделирования. На их основе делался анализ перечисленных выше методов. Для каждого метода управления трафиком ABR проводились испытания на каждом из типов буферизации с варьированием интенсивностью ABR-соединений (максимальным количеством виртуальных соединений, которым разрешено проходить через один коммутатор).

### 3.1. Анализ методов управления соединениями класса ABR

Исходя из представленных в работе [11] данных, анализировать здесь метод управления соединениями Credit-based не имеет смысла. Напомним, что проведенные ранее исследования показали: метод Credit-based намного уступает методу Rate-based по таким ключевым показателям, как средняя задержка и процент отброшенных ячеек. Поэтому проведем сравнительный анализ метода Rate-based и метода прогнозирования переполнения буферов коммутаторов.

**Т а б л и ц а 1. Итоговые результаты типичной моделирования**

№ п/п	Название параметра	Rate-based						Прогнозируемые переопределения буфера. Итоговая установка содержания, с <sup>-1</sup>					
		Интенсивность установления соединений, с <sup>-1</sup>			Раздельный			Общий буфер			Раздельный		
		10	15	45	10	15	45	10	15	45	10	15	45
1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	Количество соединений в сети	12	9	16	11	11	13	7	12	18	10	16	19
2	Средняя задержка сети, 1/100 с	181,1	131,8	188,8	272,3	247,5	252,5	406,9	466,7	310,3	252,7	202,1	237,5
3	Средняя задержка сети по соединениям класса CBR, 1/100 с	134,5	8,53	63,05	2,60	2,91	4,13	123,1	109,0	124,3	4,13	4,60	3,95
4	VBR, 1/100 с	78,96	131,2	171,1	7,5	14,32	8,64	324,4	325,4	260,7	7,49	9,50	8,197
5	ABR, 1/100 с	296,3	178,6	238,2	577,3	451,5	537,8	595,7	780,8	451,0	576,9	391,1	439,6
6	Общий процент отброшенных ячеек	0,57	8,96	3,26	1,81	12,06	7,98	41,81	40,27	39,67	0,33	2,56	0,65
7	Процент служебных ячеек в сети	5,44	9,34	9,32	11,78	9,73	10,43	10,36	15,26	16,11	9,25	17,12	19,22
8	Процент отброшенных ячеек по отношению к трафикам ABR	1,3	16,62	5,6	3,64	20,14	15,76	61,6	60,21	56,24	0,74	4,89	5,36
9	Процент служебных ячеек по отношению к соединениям класса ABR	11,60	17,27	16,12	21,56	16,53	20,09	20,51	23,04	27,3	18,77	28,78	31,62
10	Отказано в создании соединений	81	72	44	91	62	53	89	71	36	94	52	39
11	Успешно закончили работу соединения ABR	13	4	10	5	14	11	3	7	11	7	8	3
12	Удалены по тайм-ауту ABR	19	40	55	18	38	48	26	35	60	16	49	67
13	Успешно передано пакетов	3381	2937	2734	2931	2631	2582	2673	2604	2554	3014	2887	2802

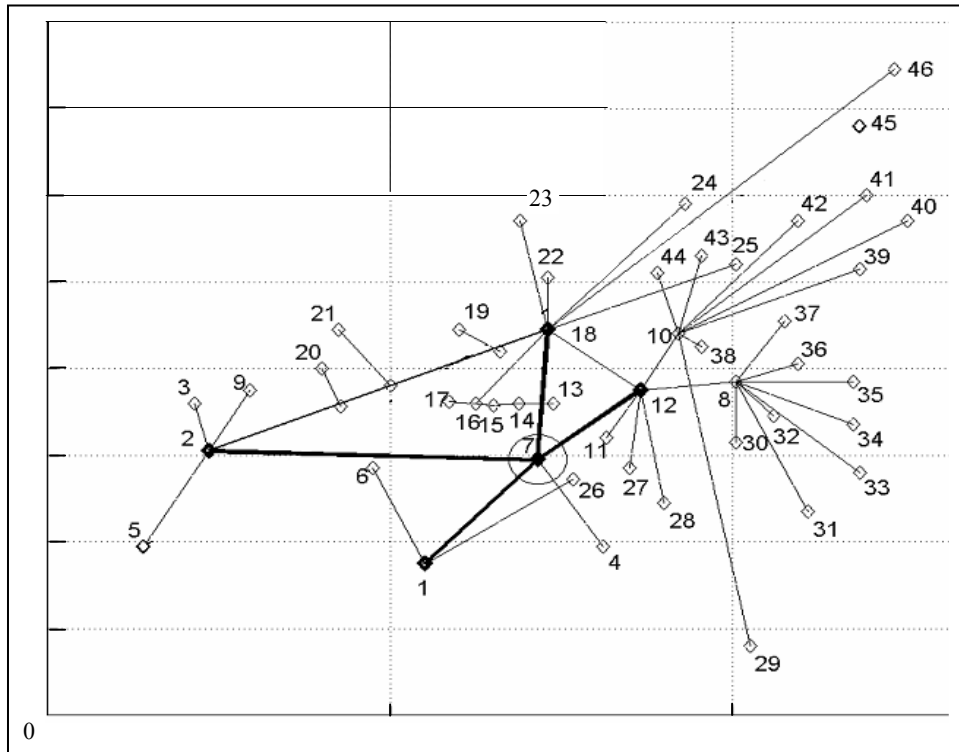


Рис. 3. Структура моделируемой сети

Построим графики, отображающие эффективность использования каналов — количество успешно переданных пакетов (рис. 4).

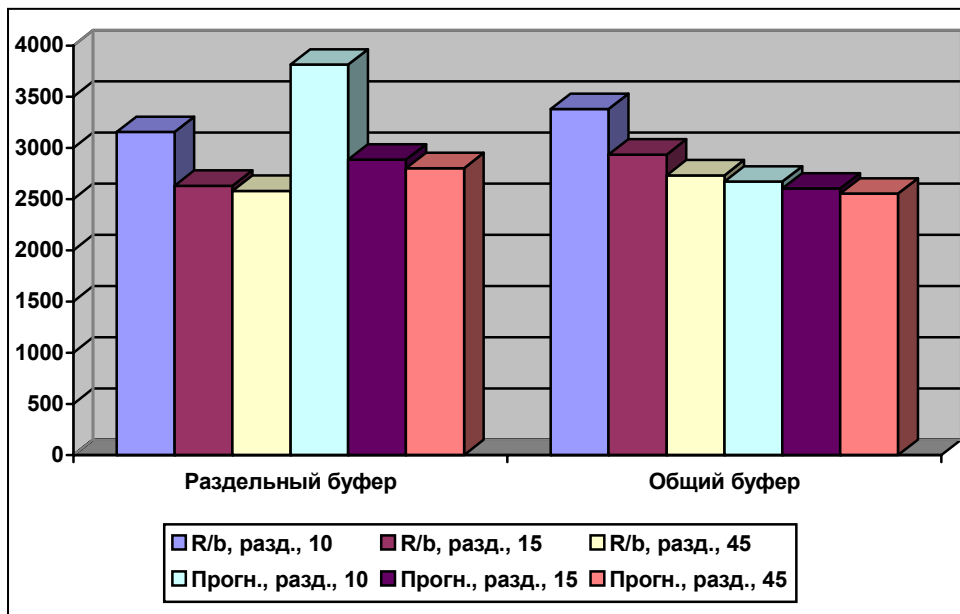


Рис. 4. График количества успешно переданных пакетов: R/b — базовый метод Rate-based; прогн. — метод с прогнозированием; разд. — раздельный буфер

Как видно из рис.4, при использовании более эффективного метода буферизации с раздельными буферами эффективность метода прогнозирования выше существующего метода Rate-based.

Построим диаграмму, совмещающую данные о проценте отброшенных ячеек в сети (рис. 5).

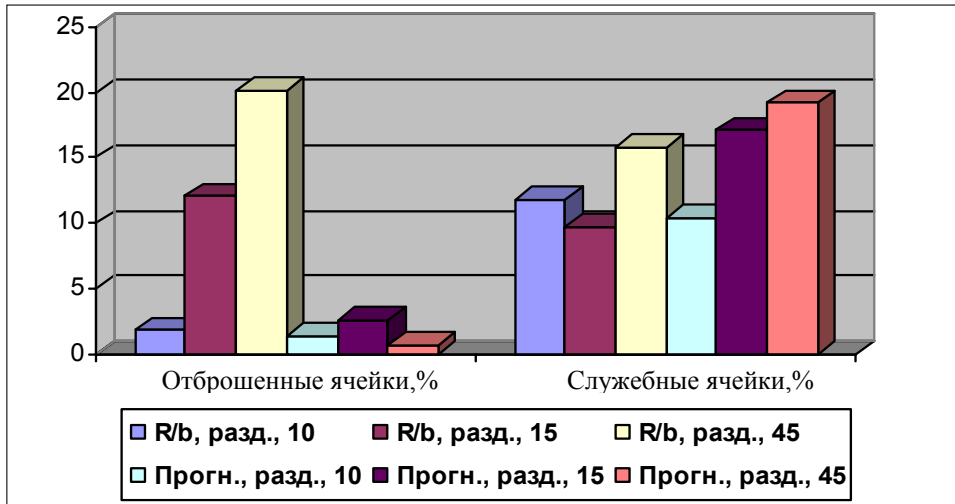


Рис. 5. Графики отброшенных и служебных ячеек в сети при использовании раздельных буферов, %: R/b — базовый метод Rate-based; прогн. — метод с прогнозированием; разд. — раздельный буфер

Как следует из рис. 5, метод управления с прогнозированием переполнения буферов коммутатора намного эффективнее справляется с увеличением интенсивности соединений класса ABR чем используемый Rate-based. Косвенным доказательством, подтверждающим это, является показатель процента служебных ячеек в сети, представленный в правой части диаграммы на рис. 5, где в среднем этот показатель выше, чем при использовании метода Rate-based. Рассмотрим показатель средней задержки в сети (рис. 6).

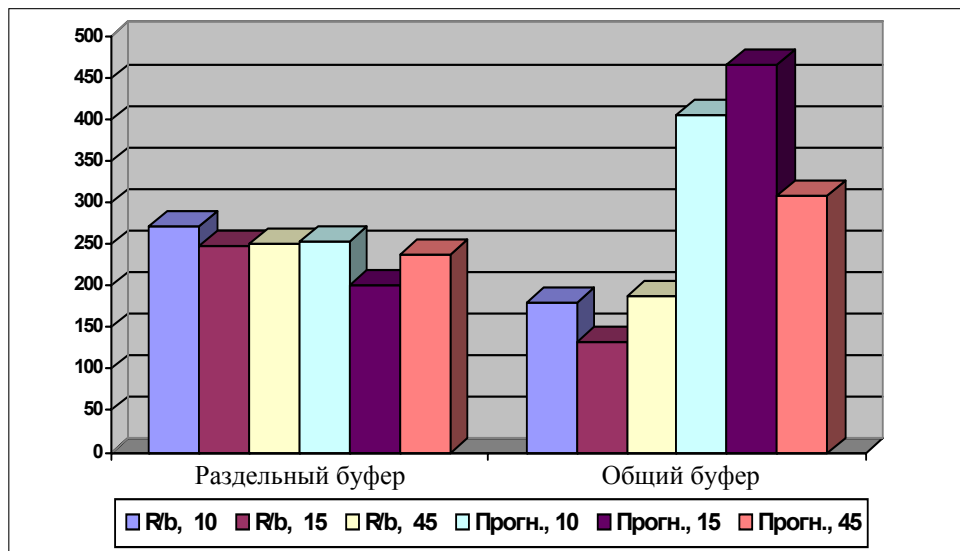


Рис. 6. Графики средней задержки ячеек в сети, мс



Здесь и далее величины задержек приводятся в мс.

Как следует из диаграмм рис. 6, метод управления с прогнозированием оказывается предпочтительнее метода для отдельного буфера по классам соединений.

Исходя из сказанного выше, имеет смысл рассмотреть детальнее метод управления соединениями класса ABR на основе прогнозирования переполнения буферов коммутаторов.

Рассмотрим графики средней задержки сети для разных интенсивностей соединений ABR (рис. 7, 8).

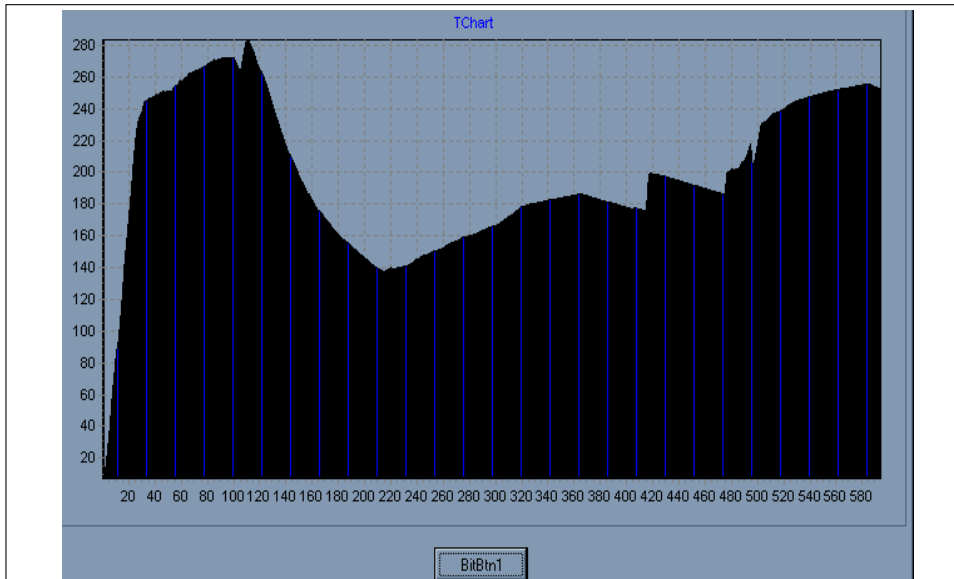


Рис. 7. График мгновенных значений средней задержки в сети. Интенсивность соединений ABR — 10 ед/с

Очевидно, что интенсивность соединений ABR практически не влияет на среднюю задержку в сети. Метод одинаково эффективно справляется как с интенсивностью в 10 соединений, так и в 45.

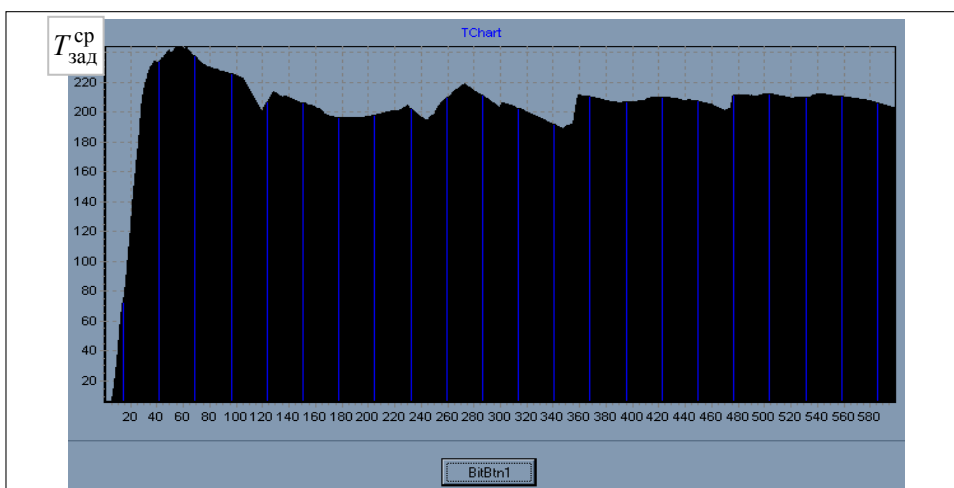


Рис. 8. График мгновенных значений средней задержки в сети. Интенсивность соединений ABR — 45 ед/с

Очевидно, что интенсивность соединений ABR практически не влияет на среднюю задержку в сети. Метод одинаково эффективно справляется как с интенсивностью в 10 соединений, так и в 45.

Посмотрим, каким образом влияет на эффективность метода варьирование значения порога заполненности буфера коммутатора. Результаты измерений для значений порога 50, 80 и 90% приведены в табл.2.

**Таблица 2.** Результаты экспериментов при варьировании порогов заполнения буфера

№ п/п	Название параметра	Метод прогнозирования		
		Порог, %		
		50	80	90
1	2	3	4	5
1	Количество трафиков в сети	11	10	11
2	Средняя задержка сети, 1/100 с	115,11	211,0	275,8
3	Средняя задержка сети по соединениям класса CBR, 1/100 с	4,40	4,13	2,63
4	VBR, 1/100 с	8,14	7,49	10,16
5	ABR, 1/100 с	266,44	438,78	576,12
6	Общий процент отброшенных ячеек	0,21	0,33	1,57
7	Процент служебных ячеек в сети	11,77	9,25	9,92
8	Процент отброшенных ячеек по отношению к трафикам ABR	0,48	0,74	3,14
9	Процент служебных ячеек по отношению к трафикам ABR	23,90	18,77	18,30
10	Отказано в создании соединений ABR	86	94	91
11	Успешно закончили работу соединения ABR	7	7	7
12	Удалены по тайм-ауту соединения ABR	21	16	16
13	Успешно передано пакетов	2541	3014	3605

По полученным данным построим совместный график трех ключевых показателей качества: средняя задержка в сети, количество успешно переданных пакетов и процент отброшенных ячеек. Для наглядности сведем полученные результаты в один график (рис. 9).

Очевидно, что все показатели повышаются с приближением порога к 100% объема буферов коммутаторов. При этом, если зависимости средней задержки и количества успешно переданных пакетов от значения порога заполненности буфера имеют линейный характер, то процент отброшенных ячеек больше похож на экспоненциальную зависимость.

## ВЫВОДЫ

Метод управления на основе прогнозирования переполнения буферов коммутаторов более эффективный по сравнению с методом Rate-based по таким показателям, как процент отброшенных ячеек и количество успешно переданных пакетов. Этот метод позволяет также более гибко настраивать сеть на достижение оптимальных показателей средней задержки для конкретной сети, процента отброшенных ячеек и количества успешно переданных пакетов путем варьирования значения порога заполненности буферов коммутаторов.

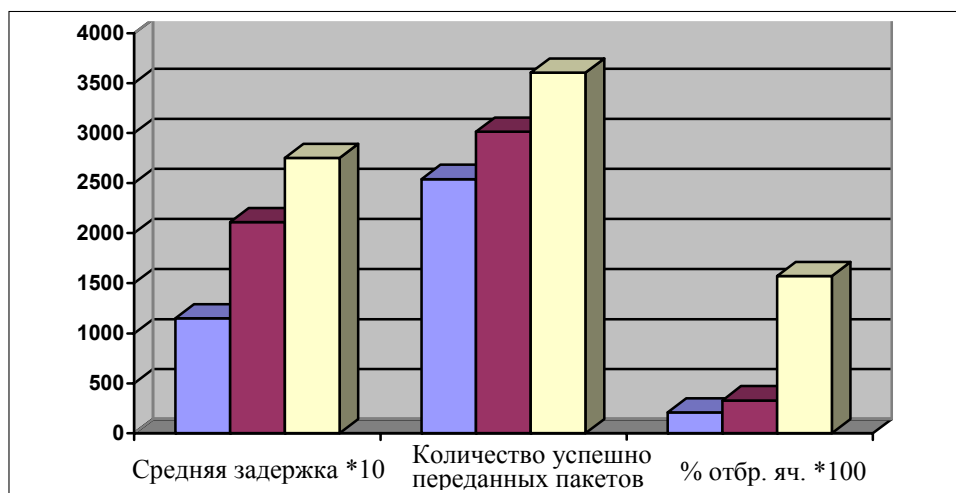


Рис. 9. Сводная диаграмма значений ключевых показателей при варьировании порога заполнения буфера: ■ — порог заполнения буфера 50 %; ■ — 80 %; □ — 90 %

Необходимо заметить, что использование этого метода подразумевает наличие у коммутаторов возможности реализации метода МНК для прогнозирования значений заполненности буферов, а также дополнительной памяти для хранения статистических данных, что намного усложняет организацию коммутатора и повышает его стоимость.

## ЛИТЕРАТУРА

1. Максимей И.В. Имитационное моделирование на ЭВМ. — М: Радио и связь, 1988. — 232 с.
2. Шеннон Р. Имитационное моделирование систем — искусство и наука. — М: Мир, 1978. — 418 с.
3. Кульгин М. Технологии корпоративных сетей: Энциклопедия. — СПб.: Питер, 1999. — 704 с.
4. Назаров А.Н., Симонов А.Н. АТМ: технология высокоскоростных сетей. — М.: Экотрендз, 1997. — 232 с.
5. Кучерявый А., Моисеев С., Пяттаев В. Технология АТМ на российских сетях связи. — М.: Радио и связь, 2002. — 240 с.
6. Олифер В.Г., Олифер Н.А. Компьютерные сети. Принципы, технологии, протоколы. — СПб.: Питер, 2001. — 672 с.
7. Скляр Б. Цифровая связь. Теоретические основы и практическое применение. Изд. 2. — СПб.: Вильямс, 2003. — 410 с.
8. Алленов О.М. Защита от перегрузок в сетях АТМ // Сети и системы связи. — 1998. — № 5. — С. 36–48.
9. Charney A., Clark D., Jain R. Congestion Control with Explicit Rate Indication // Proc. IEEE International Conference on Communications (ICC'95), June 1995. — P. 46–54.
10. Зайченко О.Ю., Зайченко Ю.П., Зайчикова О.В. Аналіз методів керування трафіком в мережах АТМ // Наук. вісті НТУУ «КПІ». — 2002. — № 3. — С. 12–17.
11. Зайченко О.Ю., Зайченко Ю.П., Круглов Д.Ю. Імітаційне моделювання корпоративної мережі АТМ із різними методами управління трафіками. — Наук. вісті НТУУ «КПІ». — 2003. — № 2. — С. 56–63.

Поступила 05.11.2004