

# Динамическая стратегия распределения буферной памяти АТМ коммутатора

Я.М. Агаларов, И.А. Соколов

**Аннотация.** Рассмотрен алгоритм динамического распределения буферной памяти АТМ коммутатора между соединениями различных служб АТМ сети. Алгоритм использует динамическую стратегию отбрасывания ячеек, учитывающую текущие длины очередей к выходным портам для каждого соединения и веса ячеек. Приведены результаты исследования эффективности данного алгоритма и результаты сравнительного анализа различных алгоритмов распределения буферной памяти.

## Введение

Одна из наиболее применяемых технологий в области связи в настоящее время является технология высокоскоростной передачи данных с использованием непрерывно следующих друг за другом ячеек фиксированной длины, называемая АТМ (Asynchronous Transfer Mode - асинхронный режим передачи). Технология АТМ является наиболее перспективным решением задачи переноса разнородной информации в широкополосных цифровых сетях с интеграцией служб.

Технология АТМ поддерживает различные уровни сервиса и качества обслуживания устанавливаемых соединений. Форум АТМ определил пять категорий сервиса [1], а именно, передачу:

- с постоянной скоростью (служба CBR);
- в реальном времени с переменной скоростью (служба rtVBR);
- с переменной скоростью не в реальном времени (служба nrtVBR);
- с незаданной заранее скоростью (служба UBR);
- с доступной скоростью (служба ABR).

Эти категории сервиса включают в себя следующие параметры качества сервиса (QoS):

- коэффициент потерь ячеек (Cell loss ratio), определяющий, какой процент высоко-

приоритетных ячеек может быть потерян за время передачи;

- задержка передачи ячейки (Cell transfer delay), определяющий количество времени (или среднее количество времени), требуемое для доставки ячейки адресату;

- изменение задержки передачи ячейки (Cell delay variation - CDV) задающее допустимые изменения в распределении группы ячеек между конечными станциями.

Обобщенно службы АТМ можно разделить на два основных класса: службы с гарантией (CBR, nrtVBR и rtVBR) и службы, доставляющие трафик с «максимальным усилием» (ABR и UBR). Для служб первого класса требуется необходимое количество сетевых ресурсов и свободной буферной памяти в коммутаторах, при этом они не будут «просить лишнего». Службы второго класса, поддерживающие доставку с «максимальным усилием» (best effort), получают ресурсы по остаточному принципу, однако, они жестко конкурируют между собой за общий пул буферного пространства и свободную часть полосы пропускания. Для того чтобы поддерживать требуемый уровень качества обслуживания для каждого виртуального соединения, коммутатор должен эффективно управлять своим буфером и другими ресурсами.

Механизмы управления буферами коммутатора являются одними из основных механизмов управления трафиком. Они не стандартизованы и представляют собой интеллектуальную собственность конкретной фирмы-производителя. Вместе с другими механизмами управления трафиком они и служат тем ключевым элементом, который отличает коммутаторы АТМ разных производителей друг от друга.

Производители оборудования применяют несколько методов разделения буферов, различаются и размеры буферов. Наиболее распространенные методы разделения буферов описаны в [2, 3]. Большинство коммутаторов АТМ реализует статический алгоритм отбрасывания ячеек, в основе которого лежат фиксированные граничные условия. Суть метода состоит в том, что в зависимости от размера всего буфера жестко определяется максимальное число ячеек одного соединения, которое может находиться в буфере. По его достижении ячейки этого соединения начинают отбрасываться. Однако такой алгоритм не способен обеспечить равномерное распределение ресурсов между соединениями, а буферное пространство используется неэффективно. Ячейки отбрасываются коммутатором без учета сервиса потока, к которому они принадлежат, и занятости буфера.

Может использоваться и другой алгоритм сброса ячеек - адаптивный. В коммутаторах от Digital, FORE, Hitachi, Olicom и Xylan на каждом из модулей имеются пулы буферов, которые по мере необходимости динамически распределяются по портам. При использовании адаптивного метода решение о сбросе ячеек учитывает текущие условия и основывается на рассмотрении двух факторов: количестве ячеек, находящихся в буфере для каждого соединения, и количестве ячеек, отведенном под соответствующую службу.

В некоторых коммутаторах, например, изделия Cisco, используется и динамическое, и фиксированное распределение буферов одновременно. Благодаря динамическим буферам достигается большая гибкость в предоставлении буферной емкости наиболее перегруженным портам. Любую дополнительную возможность управления буферами, обеспечиваемую

коммутатором, можно рассматривать как его дополнительное достоинство [4]. Отметим, что все используемые в настоящее время алгоритмы являются эвристическими, имеют свои преимущества и недостатки и задача оптимизации распределения буфера АТМ коммутатора остается открытой.

В работе рассматривается алгоритм динамического распределения оставшейся памяти АТМ коммутатора между соединениями различных служб. В данном алгоритме задаются эффективные правила выборочного отбрасывания ячеек, учитывающие текущие длины очередей к выходным портам для каждого соединения. Приведены результаты исследования предлагаемого алгоритма и результаты сравнительного анализа различных алгоритмов распределения буферной памяти.

## 1. Постановка задачи

Математическая модель узла представляется в виде системы массового обслуживания (СМО) с  $M < \infty$  различных потоков заявок (пакетов, кадров), каждый из которых обслуживается только на одном из  $M < \infty$  многоканальных приборов (линий связи), и общим числом мест (ячеек) хранения  $N$ ,  $0 \leq N < \infty$ .

Будем считать, что потоки заявок и соответствующие приборы пронумерованы числами  $1, \dots, M$ . Пусть выполняются следующие предположения:

1. Потоки заявок являются независимыми в совокупности пуассоновскими потоками с интенсивностями  $0 < \lambda_i < \infty$ ,  $i = \overline{1, M}$ . Для обслуживания  $i$ -заявки (заявки потока с номером  $i$ ) требуется одновременно одно место хранения и один канал прибора типа  $i$ ,  $i = \overline{1, M}$ .

2. Если в момент поступления  $i$ -заявки в  $i$ -м приборе есть свободный канал, то она сразу поступает на обслуживание. Если в момент поступления  $i$ -заявки в  $i$ -м приборе нет свободного канала, то поступившая заявка допускается в систему или отвергается (не допускается) согласно стратегии распределения мест хранения СМО, зависящей только от текущего состояния системы (числа заявок каждого типа в системе).

3. Принятые в СМО  $i$ -заявки на  $i$ -м приборе обслуживаются в порядке поступления.

4. Время занятия прибора  $i$ -заявкой (время выполнения  $i$ -заявки) – экспоненциально распределенная случайная величина с параметром  $0 < \mu_i < \infty$ , независимая от других случайных событий в узле,  $i = \overline{(1, M)}$ .

5. Выполненная заявка освобождает занятый канал и покидает систему навсегда.

6. Заявка  $i$ -потока имеет вес (стоимость), равный  $d_i > 0$ ,  $i = \overline{(1, M)}$ .

Введем обозначения:

$k_i$  - число  $i$ -заявок в системе в некоторый момент времени;

$\bar{k} = (k_1, \dots, k_M)$  – вектор-столбец, обозначающий состояние системы;

$c_i$  - число каналов в приборе  $i$ -го типа;

$W = \left\{ \bar{k} : \sum_{i=1}^M k_i \leq N \right\}$  – пространство состояний системы (множество всех возможных состояний системы).

Пусть  $\bar{r}(\bar{k}) = (r_1(\bar{k}), \dots, r_M(\bar{k}))$ ,  $\bar{k} \in N$ , где каждая компонента  $r_i(\bar{k})$  может принимать одно из трех значений 0, 1 или 2, причем, если  $r_i(\bar{k}) = 2$  для некоторого  $i \in \overline{(1, M)}$ , то  $r_j(\bar{k}) \neq 2$  для всех  $j = \overline{(1, M)}$ ,  $j \neq i$ . Политику распределения мест хранения определим с помощью целочисленной функции  $\bar{r}(\bar{k})$  следующим образом:

1)  $i$ -заявка принимается в систему, если в момент поступления система находится в состоянии  $\bar{k} \in W$  и  $r_i(\bar{k}) = 0$ ,

2)  $i$ -заявка не принимается в систему, если  $r_i(\bar{k}) \geq 1$ ,

3) поступившая  $i$ -заявка принимается в систему, а ранее принятая  $j$ -заявка выбрасывается из системы, если  $r_i(\bar{k}) = 0, r_j(\bar{k}) = 2$ ,  $i, j \in \overline{(1, M)}, i \neq j$ . Набор  $\bar{r} = \{\bar{r}(\bar{k}), \bar{k} \in W\}$  будем называть стратегией распределения буферной памяти. В дальнейшем мы рассматриваем только стационарные стратегии (политика во времени не меняется) и под словом стратегия будем понимать стационарную стратегию [5,6].

Пусть  $R$ - множество всех возможных стратегий вида  $\bar{r}$ .

Определим некоторые известные схемы распределения буферной памяти с помощью вектора-функции  $\bar{r}(k)$  следующим образом:

- для схемы полного разделения памяти (СР)( $\bar{r}^{cp}$ ):

$$r_i^{cp}(\bar{k}) = \begin{cases} 0, & \text{если } k_i < m_i, \\ 1, & \text{если } k_i \geq m_i, \end{cases}$$

$i = 1, \dots, n, \sum_{i=1}^n m_i = N, m_i$  - число мест хранения, выделенных для  $i$ -го потока;

- для полнодоступной схемы (СS)( $\bar{r}^{cs}$ ):

$$r_i^{cs}(\bar{k}) = \begin{cases} 0, & \text{если } \sum_{i=1}^M k_i < N, \\ 1, & \text{если } \sum_{i=1}^M k_i = N, \end{cases}$$

$i = 1, \dots, n$ ;

- для полнодоступной схемы с индивидуальными потолками (SMQ)( $\bar{r}^{smq}$ ):

$$r_i^{smq}(\bar{k}) = \begin{cases} 0, & \text{если } k_i < m_i, \\ 1, & \text{если } k_i \geq m_i, \end{cases}$$

где  $i = 1, \dots, n, \sum_{i=1}^M m_i \geq N$ ;

- для неполнодоступной схемы с индивидуальными потолками SMQMA ( $\bar{r}^{smqma}$ ):

$$r_i^{smqma}(\bar{k}) = \begin{cases} 0, & \text{если } k_i < m_i + a_i, \sum_{i=1}^M \chi(k_i > a_i) < m, \\ 1, & \text{если } k_i \geq m_i + a_i \text{ или } \sum_{i=1}^M \chi(k_i > a_i) \geq m, \end{cases}$$

где  $\chi(A) = \begin{cases} 1, & \text{если выполняется условие } A, \\ 0, & \text{если не выполняется условие } A, \end{cases}$

$a_i \geq 0, m_i \geq 0, i = 1, \dots, n, \sum_{i=1}^M m_i = m, \sum_{i=1}^M a_i + m = N$ .

Схема распределения буферной памяти, которая допускает выбивание из буферной памяти ранее принятой заявки вновь поступившей (динамическая стратегия распределения

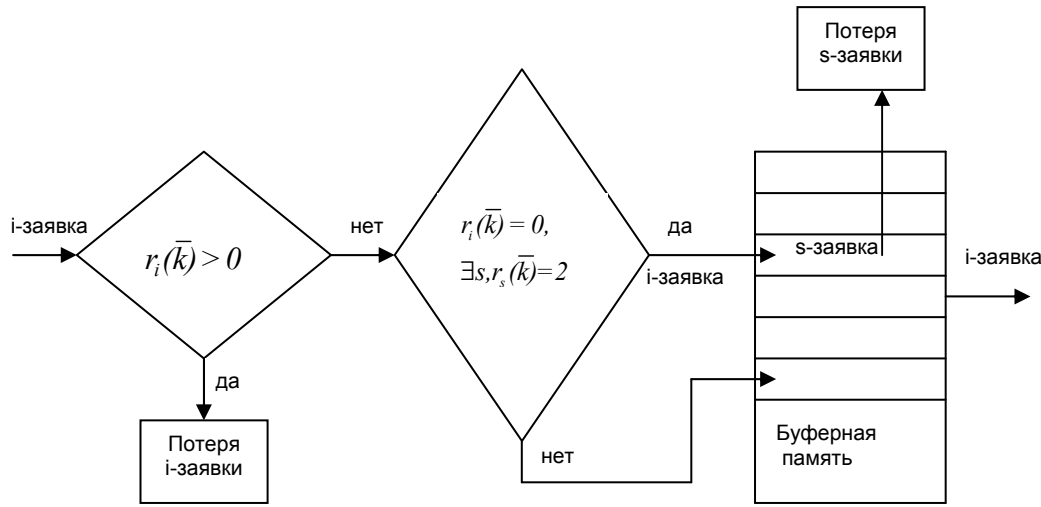


Рис. 1. Схема предоставления буферной памяти  $i$ -заявке ( $i=1, \dots, M$ ) при динамической стратегии

буферной памяти), с помощью функции  $(\bar{r}^{DS})$  определяется следующим образом:

$$r_i^{DS}(\bar{k}) = \begin{cases} 0, & \text{если } \bar{k} \in W_i^0, \\ 1, & \text{если } \bar{k} \in W_i^1, \\ 2, & \text{если } \bar{k} \in W_i^2, \end{cases}$$

где множества  $W_i^0, W_i^1, W_i^2$  такие, что для каждого  $i \in \overline{(1, M)}$   $W_i^0 \cup W_i^1 \cup W_i^2 = W$ ,  $W_i^s \cap W_i^j = \emptyset$  при  $s, j = 0, \dots, 2, s \neq j$  и, если  $W_i^2 \neq \emptyset$  для некоторого  $i=i_0$ , то для всех  $\bar{k} \in W_{i_0}^2$  верно  $k_{i_0} > 0$  и  $r_i(\bar{k}) < 2$  для всех  $i \neq i_0$ . Указанную схему распределения буферной памяти можно представить в виде Рис.1.

Обозначим через  $g^{\bar{r}}$  среднее суммарной стоимости заявок, теряемых в системе в единицу времени. Ставится задача: найти стратегию  $\bar{r}^* \in R$  такую, что

$$\min_{\bar{r} \in R} g^{\bar{r}} = g^{\bar{r}^*}. \quad (1)$$

## 2. Решение задачи

Для решения задачи воспользуемся итерационным алгоритмом динамического программирования, называемым итерационным алгоритмом (итерационной процедурой) Ховарда [5,6].

Отметим, что для каждой фиксированной стратегии  $\bar{r} \in R$  процесс перехода рассматриваемой СМО из одного состояния  $\bar{k}$  в другое

описывается марковским процессом (следует из введенных в постановке задачи предположений и типа рассматриваемых стратегий  $\bar{r} \in R$ ) [7]. В данном марковском процессе переход системы из одного состояния в другое может произойти только в двух случаях: при поступлении заявки или при окончании обслуживания заявки.

Обозначим:

$A^{\bar{r}} = (a^{\bar{r}}(\bar{k}, \bar{j}))_{(\bar{k}, \bar{j}) \in W}$  - матрица интенсивностей перехода рассматриваемого марковского процесса;

$$\mu_i(k_i) = \begin{cases} k_i \mu_i, & \text{если } k_i \leq c_i, \\ c_i \mu_i, & \text{если } k_i \geq c_i, \end{cases} \quad \text{интенсивность}$$

обслуженного потока  $i$ -заявок в состоянии  $\bar{k}$ ,  $i = 1, \dots, M$ ;

$1_i$  - вектор-столбец, у которого  $i$ -я компонента равна 1, а остальные равны 0,  $i = 1, \dots, M$ .

Интенсивность перехода из состояния  $\bar{k}$  в  $\bar{j}$ , вызванная поступлением или окончанием обслуживания  $i$ -заявки, равна

$$a_i^{\bar{r}}(\bar{k}, \bar{j}) = \begin{cases} \lambda_i, \bar{j} = \bar{k} + 1_i, r_i(\bar{k}) = 0, r_s(\bar{k}) \neq 2, s = 1, \dots, M, \\ \lambda_i, \bar{j} = \bar{k} + 1_i - 1_s, r_i(\bar{k}) = 0, r_s(\bar{k}) = 2, s \in \overline{(1, M)}, \\ \mu_i(k_i), k_i \geq 1, \bar{j} = \bar{k} - 1_i, \bar{k} \in W, \\ 0, \text{ в остальных случаях,} \end{cases} \quad (2)$$

при  $\bar{k} \neq \bar{j}$ ,  $\bar{k}, \bar{j} \in W$ . Тогда по определению матрицы интенсивностей перехода

$$a^{\bar{r}}(\bar{k}, \bar{j}) = \sum_{i=1}^M a_i^{\bar{r}}(\bar{k}, \bar{j}) \text{ при } \bar{k} \neq \bar{j}$$

$$\text{и } a^{\bar{r}}(\bar{k}, \bar{k}) = - \sum_{\bar{j} \in W, \bar{j} \neq \bar{k}} a^{\bar{r}}(\bar{k}, \bar{j}), \bar{k}, \bar{j} \in W.$$

Отметим, что рассматриваемый марковский процесс всегда имеет стационарные вероятности состояний (это следует из введенных предположений и типа рассматриваемой СМО – неприводимая марковская цепь и конечное число состояний) [7]. Как видим, для решения задачи можно пользоваться итерационным алгоритмом Ховарда. Определим начальную стратегию  $\bar{r}^h$ , положив

$$W_i^0 = \{\bar{k} : k_i < m_i\}, W_i^1 = \{\bar{k} : k_i \geq m_i\}, W_i^2 = \emptyset,$$

$$r_i^h(\bar{k}) = \begin{cases} 0 & \text{при } \bar{k} \in W_i^0, \\ 1 & \text{при } \bar{k} \in W_i^1, \end{cases}$$

$$m_i \geq 0, i = 1, \dots, M, \sum_{i=1}^M m_i = N.$$

Согласно начальной стратегии  $i$ -заявка в систему допускается, если число  $i$ -заявок в системе меньше  $m_i$ , и отвергается в противном случае, причем  $\sum_{i=1}^M m_i = N$ , т.е.  $\bar{r}^h$  является стратегией типа СР. Из (2) следует, что при  $\bar{r} = \bar{r}^h$   $a_i^{\bar{r}}(\bar{k}, \bar{j})$  имеют вид:

$$a_i^{\bar{r}}(\bar{k}, \bar{j}) = \begin{cases} \lambda_i, & \text{если } \bar{j} = \bar{k} + 1, k_i < m_i, \\ \mu_i(k_i), & \text{если } k_i \geq 1, \bar{j} = \bar{k} - 1, \\ 0, & \text{в остальных случаях,} \end{cases} \quad (3)$$

$\bar{k} \neq \bar{j}$ ,  $\bar{k}, \bar{j} \in W$ ,  $i = 1, \dots, M$ .

Обозначим:  $q_{\bar{k}}^{\bar{r}}$  - норму стоимостных потерь в состоянии  $\bar{k}$  (стоимость теряемых в единицу времени заявок, когда система находится в состоянии  $\bar{k}$ ) при стратегии  $\bar{r}$ ,  $\pi_{m_i}(\rho_i)$  - стационарная вероятность того, что в системе  $M/M/c_i/m_i$  заявками заняты  $m_i$  мест хранения, т.е. вероятность потери заявки. Ниже для краткости вместо слов «стоимостные потери» будем писать «потери». Заметим, что при  $\bar{r}^h$  каждые  $i$ -й прибор и  $m_i$  мест хранения образуют СМО типа  $M/M/c_i/m_i$  и верна формула [8]:

$$\pi_{m_i}(\rho_i) = C_{oi} \rho_i^{m_i} / c_i! (\rho_i / c_i)^{m_i - c_i},$$

$$\text{где } \rho_i = \frac{\lambda_i}{\mu_i}, C_{oi} = \left( \sum_{s=0}^{m_i} p_s \right)^{-1},$$

$$p_s = \begin{cases} \rho_i^s / s!, & s \leq c_i \\ \frac{\rho_i^{c_i}}{c_i!} \left( \frac{\rho_i}{c_i} \right)^{s - c_i}, & c_i \leq s \leq m_i. \end{cases}$$

Следовательно, имеем

$$q_{\bar{k}}^{\bar{r}^h} = \sum_{i: k_i \geq m_i} d_i \lambda_i, g^{\bar{r}^h} = \sum_{i=1}^M d_i \lambda_i \pi_{m_i}^{\bar{r}^h}(\rho_i), \bar{k} \in W. \quad (4)$$

Для произвольной стратегии  $\bar{r} \in R$

$$q_{\bar{k}}^{\bar{r}} = \sum_{i: r_i(\bar{k}) \geq 1} d_i \lambda_i + \sum_{s=1}^M d_s \chi(r_s(\bar{k}) = 2) \sum_{i: r_i(\bar{k}) = 0} \lambda_i, \bar{k} \in W.$$

Согласно итерационной процедуре Ховарда для улучшения стратегии достаточно найти решение  $\bar{r}(\bar{k})$  такое, чтобы хотя бы в одном состоянии  $\bar{k}$  выполнялось условие

$$q_{\bar{k}}^{\bar{r}} + \sum_{\bar{j} \in W} a^{\bar{r}}(\bar{k}, \bar{j}) V_{\bar{j}} < g^{\bar{r}}, \quad (5)$$

где  $g^{\bar{r}}$  и  $V_{\bar{j}}$ ,  $\bar{j} \in W$ , являются решением системы уравнений

$$g^{\bar{r}} = q_{\bar{k}}^{\bar{r}} + \sum_{\bar{j} \in W} a^{\bar{r}}(\bar{k}, \bar{j}) V_{\bar{j}} \quad (6)$$

при  $\bar{V}_0 = 0$ ,  $\bar{0} = (0, \dots, 0)$  – вектор-столбец, соответствующий нулевому состоянию системы.

Кроме того, если  $\bar{r}'$  - улучшенная стратегия, то

$$g^{\bar{r}'} - g^{\bar{r}} = \sum_{\bar{k} \in W} \pi_{\bar{k}}^{\bar{r}'} \gamma_{\bar{k}}, \quad (7)$$

где  $\gamma_{\bar{k}} = q_{\bar{k}}^{\bar{r}'} + \sum_{\bar{j} \in W} a^{\bar{r}'}(\bar{k}, \bar{j}) V_{\bar{j}} - q_{\bar{k}}^{\bar{r}} - \sum_{\bar{j} \in W} a^{\bar{r}}(\bar{k}, \bar{j}) V_{\bar{j}}$ .

Найдем улучшенную стратегию. Для краткости иногда вместо обозначения  $\pi_{m_i}(\rho_i)$  будем писать  $\pi_i$ . Для начальной стратегии  $\bar{r}^h$  из (6), подставив (3) и (4), получим

$$\sum_{i=1}^M d_i \lambda_i \pi_i = \sum_{i: k_i \geq m_i} d_i \lambda_i + \sum_{i: k_i < m_i} \lambda_i V_{\bar{k}+1} + \sum_{i: k_i \geq 1} \mu_i k_i V_{\bar{k}-1} - \left( \sum_{i: k_i < m_i} \lambda_i + \sum_{i: k_i \geq 1} \mu_i k_i \right) V_{\bar{k}}, \bar{k} \in W \quad (8)$$

Система уравнений (8) имеет решение вида:

$$V_{\bar{k}} = \sum_{i=1}^M v_{k_i}, v_{k_i} = v_{k_{i-1}} + u_{k_i}, v_{k_i} = 0$$

при  $k_i = 0, i = 1, \dots, M, \bar{k} \in W$ ,

$$u_{k_i} = \begin{cases} \frac{k_i - 1}{\rho_i} u_{k_{i-1}} + d_i \pi_i & \text{при } k_i = 1, \dots, c_i, \\ \frac{c_i}{\rho_i} u_{k_{i-1}} + d_i \pi_i & \text{при } k_i = c_i, \dots, m_i, \\ d_i (1 - \pi_i) c_i \mu_i / \lambda_i & \text{при } m_i \leq k_i \leq N, \\ u_{k_i} = 0 & \text{при } k_i = 0, \end{cases} \quad (9)$$

$i = 1, \dots, M$ .

Из (9) следует справедливость равенств:

$$u_{k_i} = \begin{cases} \frac{d_i \pi_{m_i}(\rho_i)}{\pi_{k_{i-1}}(\rho_i)} & \text{при } k_i = 1, \dots, m_i, \\ d_i [1 - \pi_{m_i}(\rho_i)] c_i \mu_i / \lambda_i & \text{при } k_i = m_i, \dots, N, \end{cases} \quad (10)$$

$i = 1, \dots, M$ .

Рассмотрим левую часть неравенства (5) для некоторого фиксированного состояния  $\bar{k} \in W$ . Проведя эквивалентные преобразования левой части неравенства (5) и решая затем задачу минимизации найденного выражения, получим, что левая часть (5) для любого состояния  $\bar{k}$  достигает минимум при стратегии  $\bar{r}$  такой, что

$$r_i(\bar{k}) = 0, i = 1, \dots, M \text{ если } \sum_{i=1}^M k_i < N \text{ и}$$

$$r_i(\bar{k}) = \begin{cases} 0, & \text{если } u_{k_{i+1}} - d_i < u_{k_s} - d_s, \\ 1, & \text{если } u_{k_{i+1}} - d_i \geq u_{k_s} - d_s, \\ 2, & \text{если } i = s, \end{cases} \quad (11)$$

$$i = 1, \dots, M, \text{ если } \sum_{i=1}^M k_i = N,$$

где  $u_{k_s} - d_s = \max_{i \in (1, M)} \{u_{k_i} - d_i\}$ ,  $u_{k_i}$  вычисляются для  $i = 1, \dots, M$  по формуле (10).

Таким образом, мы можем утверждать, что полученная стратегия  $\bar{r}$  типа DIN из (11) гарантированно улучшает начальную стратегию  $\bar{r}^H$  типа CP, но не гарантирует решение задачи (1).

### 3. Примеры

Ниже приведены результаты вычислительных экспериментов, проведенных с использованием имитационной модели узла коммутации. Во всех рассмотренных ниже примерах стратегия CP - оптимальная на своем классе, стратегия SMQ выбрана по правилу квадратного корня [2], которое дает удовлетворительное приближение оптимальной стратегии для полнодоступной схемы с одинаковыми потолками.

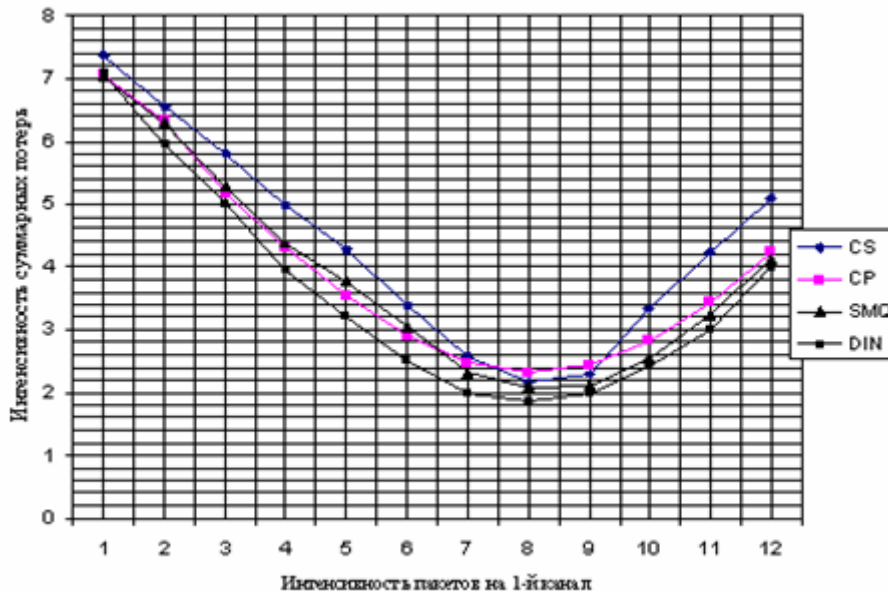


Рис.2. Зависимость интенсивности потерь пакетов от типа стратегии и интенсивности поступающей нагрузки ( $M=3, \lambda_1 + \lambda_2 + \lambda_3 = 18, \lambda_i = 1, \dots, 12, \mu_1 = 8, \mu_2 = 6, \mu_3 = 6, c_i = 1, d_i = 1, i = 1, 2, 3$ )

На Рис.2 изображены графики зависимостей суммарной интенсивности потерь пакетов от типа стратегии и интенсивности пакетов, поступающих на линии для случая сильно-загруженного узла и трех исходящих линий ( $M=3$ ), емкости накопителя  $N=20$ , суммарной поступающей нагрузки  $\lambda_1+\lambda_2+\lambda_3=18$ ,  $\lambda_i=1, \dots, 12$ ,  $\mu_1=8$ ,  $\mu_2=6$ ,  $\mu_3=6$ ,  $c_i=1$ ,  $d_i=1$ ,  $i=1,2,3$ . В Табл. 1 приведен табличный вид данной зависимости: в 1-й строке даны значения переменной  $\lambda_i$ , в строках 2-5 даны значения функции (1-й столбец – тип стратегии).

В Табл.2 и на Рис.3 показаны результаты, аналогичные первому случаю, для средне-загруженного узла.

Были проведены расчеты по сравнительному анализу стратегий полного разделения памяти и динамического распределения памяти по формуле  $\Delta g = g^{\bar{r}} - g^{\bar{r}''}$  (7). Табл. 3 содержит статистические оценки значений абсолютных и от-

носительных (в процентах) приращений интенсивности потерь для узла с параметрами  $M=3$ ,  $\lambda_1+\lambda_2+\lambda_3=18$ ,  $\lambda_i=1, \dots, 12$ ,  $\mu_1=8$ ,  $\mu_2=6$ ,  $\mu_3=6$ ,  $c_i=1$ ,  $d_i=1$ ,  $i=1,2,3$ .

В заключение отметим следующие выводы, вытекающие из полученных выше результатов:

- классы стратегий типа CS, CP, SMQ, SMQMA являются подклассами класса динамических стратегий DIN;

- предложенная в работе динамическая стратегия, улучшающая оптимальную стратегию полного разделения памяти, может снизить суммарные потери пакетов в узле до 30%;

- динамическая стратегия эффективнее, чем стратегии CS, CP, SMQ, по всему диапазону изменений нагрузки на линии связи, причем, наибольший эффект достигается при более равномерном распределении нагрузки на линии связи.

Табл. 1. Зависимость интенсивности потерь пакетов от типа стратегии и интенсивности поступающей нагрузки

	1	2	3	4	5	6	7	8	9	10	11	12
CS	7,36	6,54	5,82	5	4,29	3,4	2,57	2,16	2,3	3,33	4,23	5,09
CP	7,05	6,33	5,17	4,3	3,54	2,9	2,48	2,32	2,45	2,84	3,45	4,22
SMQ	7,04	6,3	5,27	4,37	3,77	3,05	2,33	2,1	2,12	2,55	3,23	4,14
DIN	7,09	5,96	5,02	3,95	3,22	2,52	1,98	1,86	2	2,43	3	4,01

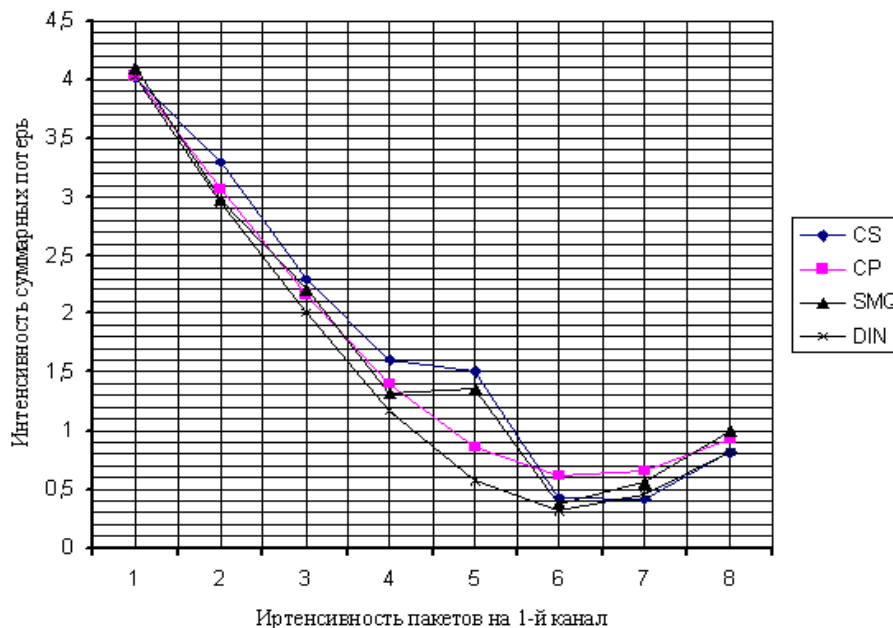


Рис.3. Зависимость интенсивности потерь пакетов от типа стратегии и интенсивности поступающей нагрузки ( $M=2$ ,  $\lambda_1+\lambda_2=9$ ,  $\lambda_i=1, \dots, 8$ ,  $\mu_1=8$ ,  $\mu_2=4$ ,  $c_i=1$ ,  $d_i=1$ ,  $i=1,2$ )

Табл.2. Зависимость интенсивности потерь пакетов от типа стратегии и интенсивности поступающей нагрузки

	1	2	3	4	5	6	7	8
CS	4,01	3,3	2,3	1,6	1,51	0,43	0,42	0,81
CP	4,02	3,06	2,16	1,4	0,86	0,61	0,66	0,93
SMQ	4,1	2,97	2,2	1,32	1,36	0,37	0,56	1,01
DIN	4,03	2,94	2	1,17	0,58	0,32	0,46	0,81

Табл. 3. Значения приращения  $\Delta g$ 

$\lambda_1$	1	2	3	4	5	6	7	8	9	10	11	12
$-\Delta g$	0,28	0,17	0,41	0,75	0,95	0,8	0,47	0,3	0,35	0,61	0,82	0,87
$-\Delta g / g^{*n}$ (%)	0,4	2,6	8	18,2	26,7	27,7	18,8	13,2	14,2	21,2	23,6	20,2

## Литература

1. ATM Forum. Traffic Management Specification. Draft Version 4.1, BTD-TM-02.02, December 1998.
2. Irland M. Buffer management in a packet switch //IEEE Trans. Commun. - 1978. - 26, №3. -328-337.
3. Башарин Г.П. Лекции по математической теории телеграфика. – М.: РУДН, 2004. -189с.
4. Кевин Толли, Эндрю Хэкер. Коммутаторы АТМ: испытания продолжаются / Сети. - 1998. - № 3, с.. 36-43.
5. Ховард Р. Динамическое программирование и Марковские процессы. -М.: Советское радио, 1964. 158с.
6. Майн Х., Осаки С. Марковские процессы принятия решений. -М.: Наука, 1977. 176с.
7. Карлин С. Основы теории случайных процессов. -М.: Мир, 1971. 536с.
8. Бочаров П.П., Печинкин А.В. Теория массового обслуживания. М.: РУДН, 1995. – 529с.

**Соколов Игорь Анатольевич.** Директор Института проблем информатики РАН. Окончил Московский государственный университет им. М.В. Ломоносова в 1976 году, доктор технических наук, профессор, академик РАН, автор более 90 публикаций. Область научных интересов – информационные технологии и автоматизация.

**Агаларов Я.М.** Кандидат технических наук. Институт проблем информатики РАН.