

МАТЕМАТИЧЕСКИЕ МОДЕЛИ АЛГОРИТМОВ РЕГУЛИРОВАНИЯ И ФОРМИРОВАНИЯ ТРАФИКА

Институт компьютерных технологий Национального авиационного университета

Рассмотрена задача анализа регуляторов трафика вычислительных сетей. Разработаны математические модели регуляторов типа «дырявое ведро» и «маркерное ведро». Выведены выражения для вероятностных характеристик качества формирования трафика, используя которые, можно получать количественные оценки потребных объемов буферной памяти и скорости обработки входных потоков данных.

Введение. Эффективность использования сети в значительной степени определяется качеством управления в условиях перегрузки. Пока сеть загружена незначительно, число принимаемых и обрабатываемых пакетов равно числу пришедших. Однако, когда в сеть поступает слишком большой объем данных, может возникнуть перегрузка, и рабочие характеристики ухудшаются. При чрезмерных нагрузках пропускная способность канала или сети может стать нулевой [1]. Такая ситуация приводит к коллапсу сети.

Отчасти это может быть связано с недостатком памяти для входных буферов, но даже если маршрутизатор имеет бесконечную память, эффект перегрузки может оказаться еще более тяжелым. Это связано со временем ожидания обработки. Если оно превышает длительность таймаута, появляются повторно передаваемые пакеты, что приводит к снижению полезной пропускной способности сети. Причиной перегрузки может быть медленный процессор или «узкое горло» — низкая пропускная способность отдельного участка сети. Простое повышение быстродействия процессора или интерфейса не всегда решает проблему — узкое место, как правило, переносится в другой фрагмент сети.

Перегрузка порождает лавинные процессы: переполнение буфера приводит к потере пакетов, которые придется передавать повторно или даже несколько раз. Процессор передающей стороны получает дополнительную паразитную загрузку. Все это свидетельствует о том, что кон-

троль перегрузки является крайне важным процессом. Следует различать контроль потока и контроль перегрузки. Под контролем потока подразумевается балансировка потока отправителя и возможности приема и обработки получателем. При этом виде контроля предполагается наличие обратной связи между получателем и отправителем. В процессе участвуют, как правило, только два партнера. Перегрузка — более общее явление, относящееся к сети в целом или к ее фрагменту.

Постановка задачи. Одним из распространенных методов борьбы с перегрузками является управление с обратной связью. Механизм управления с обратной связью может улучшить производительность сети, сокращая потери пакетов, и предотвратить распространение перегрузки. В принципе можно послать уведомление о перегрузке отправителю, однако при этом перегруженный участок сети нагружается еще больше. Поэтому задача управления решается на транспортном уровне средствами протокола TCP [2]. При выявлении перегрузки скорость передачи понижается путем уменьшения размера скользящего окна.

По существу, имеет место управление с запаздывающей обратной связью. При неправильном учете характеристик запаздывания система может потерять устойчивость и перейти в незагухающий колебательный режим, либо корректировка интенсивности потока будет осуществляться слишком поздно. Это приводит к ухудшению производительности сети,

особенно для приложений реального времени. Компенсация задержки обратной связи может выполняться методами предсказания, например, с использованием модели авторегрессии и скользящего среднего (АРСС) или путем усреднения параметров окна. Второй вариант проще, но, естественно, обеспечивает значительно более низкое качество сервиса.

Положительного результата также можно достичь путем вариации значений таймаутов, изменения политики повторной передачи пакетов. В некоторых случаях позитивный результат может быть получен изменением схемы буферизации.

Управление с обратной связью широко используется в архитектуре интегрированных служб (*Integrated Service Architecture – ISA*) для поддержки служб с различными уровнями качества сервиса (*Quality of Service – QoS*) в Интернет и в частных объединенных сетях.

Для систем без обратной связи решение проблемы выравнивания скорости передачи данных может быть решено с помощью алгоритмов «дырявого ведра» и «маркерного ведра» [3].

Алгоритм «дырявого ведра», представляет собой механизм регулирования трафика, когда часть потока пакетов, превышающая пропускную способность сети, просто отбрасывается или помечается. Более совершенным алгоритм, в котором реализован механизм формирования (сглаживания всплесков) трафика, является алгоритм «маркерного ведра». Алгоритмы «дырявого ведра» и «маркерного ведра» являются основным средством обеспечения *QoS* в технологии дифференцированных служб (*Differentiated Services – DiffServ* или *DS*) – поддержки различных уровней *QoS* для разных потоков трафика и разных сетевых служб, различающихся по производительности.

Алгоритмы «дырявого ведра» и «маркерного ведра» используются для управления трафиком в разных сетях, в том числе *Frame Relay*, *ATM* и *ISDN*. Для этих алгоритмов есть, по определению, теоретическая взаимосвязь между сред-

ним размером пакета B , средней скоростью передачи R и временем передачи t :

$$R = B/t. \quad (1)$$

Поэтому для маркерного или дырявого ведра можно конкретизировать среднюю скорость передачи данных. Например, при скорости передачи 1,2 Мбит/с и размере пакета 1500 байт пакет маркеров составляет 12 000 бит, а период следования маркеров – 10 мс. В соответствии со спецификацией скорость согласованного трафика, в наихудшем случае, составляет 100 пакетов в секунду по 1500 байт каждая и со средней скоростью передачи, не превышающей 1,2 Мбит/с.

Однако модель регулятора (формирователя) трафика, описываемая выражением (1), является слишком упрощенной. Используя такую модель, невозможно добиться приемлемой эффективности управления и контроля перегрузок сети, особенно для трафика, характеризующегося разнородностью, самоподобием и существенной нестационарностью.

Целью настоящей работы является разработка математических моделей регуляторов трафика. На основе разработанных моделей можно оптимизировать алгоритмы регулирования, формирования и сглаживания трафика реального времени в гетерогенных сетях с большим разбросом скоростей передачи.

Теоретическое обоснование и анализ результатов. 1. *Дырявое ведро.* Алгоритм дырявого ведра, прежде всего, используется для формирования трафика, уходящего от интерфейса в сеть. В этой модели предусмотрена обработка с очередями и расписаниями. Трафик теоретически отбывает от интерфейса с постоянной битовой скоростью, но, по существу, многобитовые пакеты отправляются со скоростью передачи, приближающейся к теоретическому пределу, измеренному за более длинный интервал. Например, можно послать один 1500-байтовый пакет каждые 10 мс или один 500-байтовый пакет каждые 3,3 мс. Также можно сформировать дырявые ведра с разными скоростями передачи, в которых

исходящий трафик имеет переменную скорость передачи в зависимости от активности или простоя на предыдущих этапах.

Приложения в общем случае стремятся найти маршруты с наибольшей достижимой скоростью передачи. Теоретически, 10-мегабитный поток, сформированный алгоритмически, и поток, который вынужденно идет со скоростью 10 Мбит/с, потому что его узким местом является 10-мегабитный *Ethernet*, формально неразличимы. В зависимости от конфигурации, приближение к теоретическому выравниванию может изменяться в соответствии с изменениями максимальной длины единицы передачи (*MTU*) от одного маркерного интервала к другому. При использовании общего ресурса передачи один трафик может также конкурировать с другим трафиком.

2. *Маркерное Ведро*. Маркерное ведро измеряет частоту вхождения трафика от другого устройства. Этот трафик может быть первоначально сформирован с использованием формирователя типа дырявое ведро или его эквивалента. Маркерное ведро определяет, соответствует ли трафик спецификации. Обычно используются маркерные ведра с несколькими скоростями передачи (маркерные ведра с пиковой скоростью, средней скоростью и др.), как, например, описанные в [3]. В данном случае абсолютное сглаживание не ожидается, но существует соответствие одной или более указанным скоростям.

Другими словами, параметры потока данных $\{R, B\}$ согласуются с характеристиками простого маркерного ведра, если система получает на любом интервале t , как минимум, количество данных, не превышающее $(R * t) + B$.

Для случая многоскоростного маркерного ведра поток данных является согласованным, если для каждой из скоростей поток согласовывается с профилем маркерного ведра, подходящим для трафика соответствующего класса. Например, входящий трафик, который прибывает предварительно классифицированным,

по одной из «дополнительных» скоростей, только сравнивается с соответствующим «дополнительным» профилем маркерного ведра.

Существуют следующие разновидности алгоритма маркерного ведра:

– *TBF (Token Bucket Filter)* – без разделения. Полоса пропускания всего проходящего трафика ограничивается до заданного значения.

– *DRR (Deficit Round Robin)* – с одноуровневым разделением. Загрузка канала равномерно распределяется между всем классами.

– *HTB (Hierarchical Token Bucket или интеллектуальный формирователь)* – алгоритм с иерархическим разделением. Загрузка канала динамически распределяется по иерархии классов. Позволяет задать гарантированную и максимальную скорости передачи данных для каждого класса.

3. *Особенности функционирования алгоритмов*. Из-за того, что данные протокола Интернет упакованы в пакеты переменной длины, возникает некоторая неопределенность в достижении согласования при использовании любого измерителя потока. Измеритель пытается определить соответствие профилю трафика, который теоретически строится для пакетов фиксированной длины.

При использовании в качестве формирователя механизма дырявого ведра имеют место неожиданные эффекты. Дырявое ведро выпускает пакет только, когда все его биты были бы заняты: оно не занимает места для будущих данных. Если синхронизатор является достаточно скоростным, это не существенно. Но если синхронизатор относительно медленен (а для сетевых узлов необходима синхронизация по миллисекундам или по нескольким миллисекундам), может иметь место джиттер (дрожание) формируемого потока.

Возникает дилемма конструирования измерителя маркерного ведра. Когда число маркеров с полосой пропускания b , оставшихся в маркерном ведре, меньше,

чем размер обрабатываемого пакета L , может выполняться одно из трех действий.

03.1. Весь размер пакета может быть вычтен из размера ведра с отрицательным результатом. При добавлении новых маркеров в ведро, к величине b должен быть добавлен новый объем маркеров B , вместо того, чтобы просто «заполнить» ведро. Эта опция потенциально размещает на данном интервале объем данных больше чем требуется, и, соответственно, меньше в следующий. Благодаря такому подходу выдерживается среднее количество пакетов, принятых за цикл обработки маркерного ведра, равное объему пачки маркеров. Трафик принимается, если хотя бы один бит в пакете был принят, и, если это необходимо, дополняется до одного объема MTU с использованием одного или более последующих интервалов. Такая реализация измерения маркерного ведра предполагает «свободное» соответствие маркерному ведру.

13.2. При альтернативном подходе пакет может быть отвергнут, и количество маркеров в ведре остается неизменным (можно сделать попытку принять пакет с другим пороговым уровнем в другом ведре), помня, что при последующем добавлении новых маркеров должен быть добавлен новый маркерный объем B к величине b вместо того, чтобы просто «заполнить» ведро. Это потенциально допускает размещение в этот интервал маркерного ведра пачки данных размером менее, чем минимально возможный и, соответственно, больше в следующий. Подобно первой опции, это позволяет выдерживать среднее число пакетов, принятых на интервале обработки маркерного ведра, равным размеру пачки маркеров. При данном подходе трафик принимается, только если каждый бит в пакете был принят, и занимает вплоть до одного MTU объема от одного или более предыдущих интервалов, если это необходимо. При такой реализации измерителя предполагается «строгое» (или как можно более строгое) соответствие маркерному ведру.

Эта опция согласована с [4,5] и часто используется в реализациях *ATM* и *Frame Relay*.

3.3. Переменная TB может быть установлена к нулю, чтобы учесть первую часть пакета, а остаток размера пакета может находиться вне следующего окрашенного ведра. Это, конечно, имеет другой дефект: такой же пакет не может иметь как согласованных, так и не соответствующих требованиям компонентов в архитектуре *Diffserv* и поэтому действительно не подходит. В дальнейшем эта опция не рассматривается.

К сожалению, точное соответствие спецификации пачки маркеров с пакетами случайного размера не может быть достигнуто. Поэтому маркерные ведра в случае пакетов переменной длины всегда имеют некоторые отличия от теоретической оценки. Это также наблюдается в спецификации категории обслуживания с гарантированной частотой кадров (*GFR*) *ATM* и *Frame Relay*.

4. Математическая модель маркерного ведра со строгим соответствием. Схема алгоритма маркерного ведра изображена на рис. 1. Суть алгоритма заключается в следующем. Максимальная скорость отправки потока пакетов из управляющего узла зависит от скорости прибытия в него разрешений-маркеров: один маркер на передачу N единиц данных. Очередной пакет может быть отправлен только при получении числа маркеров, достаточного для передачи данных, объем которых больше или равен размеру пакета. Если пакет поступит в управляющее устройство, не располагающее необходимым количеством разрешений, он будет отброшен, как и маркер, поступивший в переполненный буфер-формирователь.

На рис. 2 графически изображен процесс формирования трафика с постоянной (разрешенной) скоростью и сглаживания всплесков трафика по алгоритму маркерного ведра. На входе формирователя (см. рис.1) имеется буфер с конечным объемом памяти.



Рис. 1. Схема алгоритма маркерного ведра

Поступающий со случайной скоростью трафик постепенно заполняет буфер, занимая и область выше разрешенной скорости. Генератор разрешений выдает маркеры с постоянной скоростью, и, соответственно, пакеты данных проходят через устройство управления также с постоянной скоростью. Если всплеск трафика заканчивается, прежде чем буфер заполнится до

предела, пакеты выстраиваются в последовательность с постоянным периодом следования. В случае переполнения буфера происходит сброс вновь поступающих пакетов, и часть пакетов теряется. При более совершенном механизме регулирования трафика пакеты могут помечаться и возвращаться на узел-отправитель.

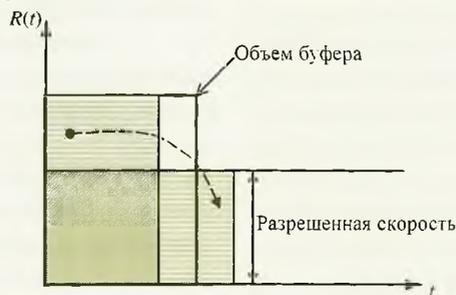


Рис. 2. Схема работы алгоритма маркерного ведра

Процесс формирования трафика, изображенный на рис. 2, может иметь место в идеальных условиях:

- все входящие пакеты умещаются в буфере данных;
- после постановки в очередь все пакеты проходят формирователь;
- скорость выходного трафика постоянна, а период следования пакетов больше, чем максимальная длительность пакета во входном трафике;
- ни один пакет не отбрасывается.

Реально входной трафик представляет собой поток пакетов со случайными параметрами: периодом, интенсивностью и длительностью каждого пакета. Чтобы

получить некоторые усредненные оценки качества регулирования трафика, необходимо задать статистические характеристики потока. Представим входной поток в виде последовательности пакетов переменной длительности (см. рис.3).



Рис. 3. Входной поток

Мгновенная интенсивность потока [2] – случайная величина с бесконечным множеством состояний v_i . Обозначим ее $\lambda_{\text{мгп}}$. Процесс смены состояний через малые промежутки времени τ_k , на которых мгновенная интенсивность считается по-

стоянної, можно розглядати як Марковський процес [6]. Для вичислення сумарних ймовірностей стану зададим наступні можливі стани системи. Вважається, що період слідування маркерів більше максимально можливої тривалості пакета, яка задається адміністратором даного фрагмента мережі. Розглядається інтенсивність трафіку λ , усереднена на певному інтервалі спостереження.

1. Інтенсивність трафіку λ_0 настільки мала, що буфер даних завжди опустошений, а буфер маркерів завжди заповнений. Очередной пакет даних, поступаючий на вхід формувача, негайно обслуговується (передається на вихід). Середня частота вихідного потоку даних дорівнює середній частоті входного потоку. Стан s_0 .

2. Інтенсивність трафіку λ_1 мала, але буфер даних частково заповнений і ніколи не переповнюється. Виникають випадкові коливання обсягу заповненої частини буфера даних. Буфер маркерів завжди заповнений, але його переповнення не має місця. Пакети даних, поступаючі в моменти всплеску інтенсивності трафіку, очікують своєї черги на обробку в формувачі, але ні один пакет не втрачається. Частота вихідного потоку даних дорівнює частоті слідування маркерів. Стан s_1 .

3. Інтенсивність трафіку λ_2 настільки велика, що буфер даних завжди заповнений, а буфер маркерів завжди опустошений.

Пакети даних, поступаючі в момент повного заповнення буфера, або просто відкидаються і втрачаються, або маркуються і відправляються по іншому маршруту. Частота вихідного потоку даних дорівнює частоті слідування маркерів. Стан s_2 .

Представимо миттєву інтенсивність входного трафіку як елементарну частинку, а процес переходу між станами в друге – як процес випадкового блукання цієї частинки. Переміщення частинки може відбуватися зі стану s_0 до стану s_1 і навпаки, а також з s_1 до стану s_2 і навпаки. Очевидно, стан s_1 є певним ідеальним станом, ймовірність якого $p_1 = P(s = s_1)$ дорівнює нулю. Тому розглядаємо процес, строго кажучи, не як Марковський. [7] Для формального приведення процесу до Марковського введемо деякі додаткові стани $s_{1+\epsilon}$ і $s_{1-\epsilon}$. Ймовірності $p_{1+\epsilon} = P(s = s_{1+\epsilon})$ і $p_{1-\epsilon} = P(s = s_{1-\epsilon})$ цих станів (так звані фіктивні ймовірності) є величинами другого порядку малості порівняно з ймовірностями $p_0 = P(s = s_0)$ і $p_2 = P(s = s_2)$ станів s_0 і s_2 відповідно. Тут $s_1 \pm \epsilon$ – мала околиця точки s_1 . Тоді процес можна розглядати як блукання частинки між двома пружними жорсткими екранами S_0 і S_1 (рис.4).

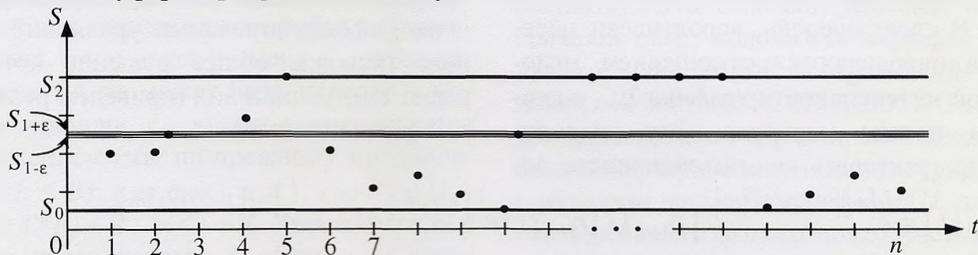


Рис. 4. Процес переходу між станами

Наявність екранів означає наступне:

– якщо частинка потрапляє в точку S_0 , то в наступний момент часу частинка с

ймовірністю r потрапить в точку $S_0 + \epsilon$ або з ймовірністю $1-r$ залишиться в точці S_0 ;

- если частица попадает в точку S_2 , то в следующий момент времени частица с вероятностью q попадет в точку $S_2 - \epsilon$.
- либо с вероятностью $1 - q$ останется в точке S_2 .

Учитывая, что вероятность состояния s_1 равна нулю, рассмотрим возмож-

ные состояния $s_0, s_{1-\epsilon}, s_{1+\epsilon}, s_2$. На рис. 5 изображен размеченный граф состояний системы. Против каждой стрелки представлена соответствующая вероятность перехода.

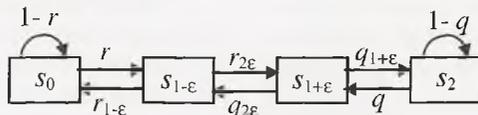


Рис. 5. Граф состояний системы

Изменения во времени соответствующих вероятностей состояния $p_0, p_{1-\epsilon}, p_{1+\epsilon}, p_2$ удовлетворяют уравнениям Кол-

могорова [6,7]. Для рассматриваемого случая система уравнений имеет следующий вид:

$$\left. \begin{aligned} \frac{dp_0(t)}{dt} &= -rp_0(t) + (1-r)p_0(t) + r_{1-\epsilon}p_{1-\epsilon}(t), \\ \frac{dp_{1-\epsilon}(t)}{dt} &= -r_{1-\epsilon}p_{1-\epsilon}(t) - r_{2\epsilon}p_{1-\epsilon}(t) + rp_0(t) + q_{2\epsilon}p_{1+\epsilon}(t), \\ \frac{dp_{1+\epsilon}(t)}{dt} &= -q_{2\epsilon}p_{1+\epsilon}(t) - qp_{1+\epsilon}(t) + r_{2\epsilon}p_{1-\epsilon}(t) + q_{1+\epsilon}p_2(t), \\ \frac{dp_2(t)}{dt} &= -qp_2(t) + (1-q)p_2(t) + q_{1+\epsilon}p_{1+\epsilon}(t). \end{aligned} \right\} \quad (2)$$

Интегрируя эту систему уравнений при заданных начальных условиях, получаем вероятности состояний как функции времени. Начальные условия выбираются в зависимости от того, каково состояние системы в момент начала отсчета ($t = 0$). В частности, для примера, которому соответствует график на рис. 4, начальные условия таковы: при $t = 0$ $p_{1-\epsilon} = 1, p_0 = p_{1+\epsilon} = p_2 = 0$.

В свою очередь, вероятности переходов определяются отношением мгновенной интенсивности трафика $\lambda_{\text{мгн}}$ к скорости потока маркеров. Эту скорость можно трактовать как интенсивность по-

тока обслуживания пакетов μ . Для случая неизменной на интервале наблюдения интенсивности трафика λ стационарное решение системы уравнений (2) тривиально. Оно представляет собой экспоненциальную функцию, параметры которой полностью определяются соотношением λ/μ и начальным состоянием системы.

В случае переменной интенсивности трафика получаем систему параметрических дифференциальных уравнений первого порядка, общее решение которой имеет следующий вид (приведем решение для вероятности p_0):

$$p_0(t) = e^{\{-\int [1-2r(t)]dt\}} \int p_{1-\epsilon}(t) r_{1-\epsilon}(t) e^{\int r(t)dt} dt + C e^{\{-\int [1-2r(t)]dt\}}, \quad (3)$$

где C – постоянная, определяемая из начальных условий.

Таким образом, записав систему уравнений Колмогорова типа (2) и отыскав ее совместное решение для вероятностей состояний в виде системы уравнений типа (3), можно получить количест-

венные оценки качества регулирования трафика в зависимости от интенсивности входного потока и скорости обработки в устройстве управления.

Строгое соответствие поведения маркерного ведра, определенное в [5,8-9], не обязательно для согласования с каки-

ми-нибудь текущими стандартами или рекомендациями. Здесь представлено математическое определение функционирования маркерного ведра под управлением по двум параметрам. Этот алгоритм согласован с рекомендациями международных организаций по стандартизации и может также использоваться для определения профиля формирования.

Пусть маркерное ведро имеет размер B и скорость накопления R . В момент времени $b(t)$ маркер мгновенно занимает ведро. Предположим, что $b(0) = B$. В течение произвольного времени до момента прибытия пакета $b(t)$ не будет изменяться, так как ведро уже заполнено маркерами.

Предположим, что пакет размером L байт прибывает в момент t' . Степень занятости ведра – это все еще B . Затем, пока $L \leq B$, измеритель оценивает пакет, после чего $b(t)$ изменяется: $b(t') = B - L$.

Предположим, проходит интервал времени $\delta t = t - t'$ перед тем, как прибывает следующий пакет, размера $L' \leq B$. Непосредственно перед этим, к моменту $t - \epsilon$, ведро накопило $\delta t * R$ маркеров на интервале, вплоть до максимального количества маркеров B таким образом, что:

$$0b(t - \epsilon) = \min\{B, b(t') + \delta t * R\}$$

1) Для строгого маркерного ведра приведем тест соответствия:

$$2) \text{if } (b(t - \epsilon) - L' \geq 0) \{$$

$$3) \text{ * the packet conforms * } b(t) = b(t - \epsilon) - L'; \}$$

$$4) \text{else \{ /* the packet does not conforms */$$

$$5) b(t) = b(t - \epsilon); \}$$

Эта функция может также использоваться для определения профиля формирования. Если пакет размера L прибывает в момент t , он может быть передан в момент времени t_e , которое определяется выражением (мы по-прежнему предполагаем $L \leq B$): $t_e = \max\{t, t''\}$, где $t'' = (L - b(t') + t' * R) / R$; $b(t'') = L$ – время, когда в ведре накапливается L пакетов, и когда размер пакета согласован с мерой маркерного ведра. $t_e = t''$, только если $t > t''$.

Выводы. По результатам анализа можно сделать следующие выводы:

01. Оперативный измеритель маркерного ведра целесообразно использовать для трафика, который сформирован «дырявым ведром» или передается последовательно. Однако трафик в Интернете редко формируется таким способом: *TCP* не применяет никакого формирования трафика, скорее, он зависит от долговременного поведения АСК-синхронизации, чтобы аппроксимировать определенную скорость передачи, и явно возникает всплески трафика в течение медленного старта, повторной передачи, и быстрого обновления. Для реализации видео-через-IP, таких как [9], может быть создан доступный формирователь дырявого ведра, но часто этого не делают, а просто ставят в очередь выходные данные из кодера для передачи через подходящий интерфейс. В результате, в каждом из этих случаев, измеритель маркерного ведра может кратковременно (на одном маркерном интервале) отвергнуть трафик, который был бы принят, если бы было более долговременное представление и который требуется принять для того, чтобы приложения работали должным образом. Чтобы достичь этого, маркерный интервал B/R должен в принципе приблизиться или даже превысить время двойного оборота сигнала (*RTT*) сеанса и размер пачки B и, таким образом, приспособиться к наибольшему размеру пачки, которую может послать источник.

2. Поведение свободного маркерного ведра отлично от описания маркерного ведра для *ATM* и для *Frame Relay*. Свободное маркерное ведро не принимает пакетов, пока подсчет маркеров дает негативный результат. Это означает, что как только большой пакет «одолежит» маркеры у будущего, даже малый постулирующий пакет (напр., 40-байтовый *TCP ACK/SYN*) не будет принят. Таким образом, если такое свободное маркерное ведро сконфигурировано с размером пачки, близким к *MTU*, для пакетов меньшего размера может иметь место некоторая дискриминация. При использовании большего размера пачки удастся избежать этой проблемы.

3. Смысл предыдущего пункта – то, что заполненное маркерное ведро иногда не принимает больших пакетов, когда свободное ведро могло бы это сделать. В

этом случае, если такое свободное маркерное ведро сконфигурировано с размером пачки, близким к *MTU*, может иметь место некоторая дискриминация для больших пакетов. Опять-таки при использовании большего размера пачки удается избежать этой проблемы.

4. В глобальных приложениях *MTU* часто больше, чем размер пачки, предложенный канальным уровнем сети поставщиком услуг. Если это так, возможно, что строгая мера маркерного ведра привела бы к тому, что трафик никогда не соответствовал бы указанному профилю: этого можно избежать, не разрешая использовать такую спецификацию. Со свободным маркерным ведром данная ситуация не может возникать, так как минимальный размер пачки, который может быть сконфигурирован, составляет по определению 1 бит, вследствие чего возможность свободного маркерного ведра ограничивается наличием размера пачки, большей, чем один *MTU*.

5. Как строгие, так и свободные спецификации маркерного ведра, как определено в [4,5], подлежат постоянному опустошению. Они аккумулируют объем пачки в течение какого-то времени, вплоть до максимального размера пачки. Если максимальный размер пачки точно соответствует размеру посылаемых пакетов, только в этом случае возможна «строгая» реализация маркерного ведра. Если принят один пакет, глубина маркера становится нулевой и начинается новое накопление. Если следующий пакет получен в любое время раньше, чем поздний маркерный интервал, он не будет принят. Если следующий пакет прибывает в точное время, он будет принят, и глубина маркера опять установится равной нулю. Если он прибывает позже, однако, накопление маркеров остановится, потому что они перекрываются максимальным размером пачки: в течение интервала между заполнением ведра и фактическим прибытием пакета, никакие новые маркеры не добавляются. В результате, джиттер пакетов, который накапливается из-за множе-

ственных прыжков в сети, приводит к снижению эффективности алгоритма и сокращению полезной фактической скорости передачи. Поэтому обычно имеет смысл установить максимальный размер маркерного ведра, несколько больший, чем *MTU*, для того, чтобы скомпенсировать джиттер и обеспечить на практике полезную скорость передачи в линии больше, чем желаемая теоретическая норма.

6. Для получения количественных оценок размера маркерного ведра, скорости обработки с точностью, приемлемой для работы формирователя в условиях нестационарности и выбросов интенсивности трафика, разработан метод решения на основе теории Марковских процессов и уравнений Колмогорова.

Список литературы

1. *Виноградов Н.А., Дровозов В.И., Лесная Н.Н., Зембицкая А.С.* Анализ нагрузки на сети передачи данных в системах критичного применения // Зв'язок, № 1 (61), 2006. - С. 9 – 12.
2. *Олифер В.Г., Олифер Н.А.* Компьютерные сети. Принципы, технологии, протоколы: Учебник для вузов. 3-е изд. – СПб.: Питер, 2006. – 958 с.
3. *Столлингс В.* Современные компьютерные сети. 2-е издание. – СПб.: Питер, 2003. – 783с.
4. *Y. Bernet, S. Blake, D. Grossman, A. Smith.* An Informal Management Model for Diffserv Routers. – May 2002. – 56 p.
5. *Heinanen, J. and R. Guerin,* “A Single Rate Three Color Marker”, RFC 2697, September 1999.
6. *Тихонов В.И., Миронов М.А.* Марковские процессы. – М.: Сов. радио. – 1977. – 488 с.
7. *Вентцель Е. С.* Исследование операций. – М.: Сов. радио. – 1972. – 552 с.
8. *Heinanen, J. and R. Guerin,* “A Two Rate Three Color Marker”, RFC 2698, September 1999.
9. *McCanne, S. and Jacobson, V.,* “vic: A Flexible Framework for Packet Video”, ACM Multimedia '95, November 1995, San Francisco, CA. - p. 511-522.