

УДК 621.395.74

Дискретное имитационное моделирование алгоритма организации очереди в буфере маршрутизатора*

Е.А. БАРАБАНОВА¹, И.А. БЕРЕСНЕВ²

¹ 414056, РФ, г. Астрахань, ул. Татищева, 16, Астраханский государственный технический университет, кандидат технических наук, доцент. E-mail: elizavetaalex@yandex.ru

² 414056, РФ, г. Астрахань, ул. Татищева, 16, Астраханский государственный технический университет, аспирант. E-mail: gold1290@mail.ru

Неравномерный рост скоростей каналов передачи данных неизбежно приводит к возникновению «узких» мест в инфокоммуникационной сети и, соответственно, к возникновению перегрузок, особенно при подключении сетей доступа к транспортной сети. Традиционные протоколы управления очередями и предотвращения перегрузок не справляются с управлением трафиком высокой плотности и нелинейно изменяемой нагрузки, что приводит к возникновению перегрузок, вызывает уменьшение эффективной скорости передачи данных и ухудшает параметры качества, такие как процент потерянных пакетов, задержки и вариации задержек. Наиболее часто в сетях передачи данных используется пассивный механизм сброса излишка пакетов данных при переполнении очереди. При передаче TCP трафика этот механизм приводит к возникновению явления глобальной синхронизации. Для заблаговременного предотвращения переполнения очереди были разработаны механизмы раннего обнаружения перегрузок. В статье описываются принципы построения имитационной модели для алгоритмов активного управления очередью. Приводятся алгоритмы и листинги программ, выполняющие дискретно-событийное моделирование. Данные алгоритмы позволяют выполнять имитационное моделирование с помощью языков программирования высокого уровня. Повышение сложности современных телекоммуникационных технологий создает проблему в применении аналитических методов для оценки характеристик проектируемых систем и сетей и обуславливает широкое применение имитационного моделирования. В статье с целью предотвращения перегрузок в узлах сетей с коммутацией пакетов используются алгоритмы активного управления очередью. В соответствующей системе обслуживания, моделирующей работу узла сети, каждый поступающий пакет может быть отброшен с определенной вероятностью, даже если буфер еще полностью не заполнен. Вероятность отбрасывания зависит от длины очереди в момент поступления пакета. В статье приводятся принципы имитационного дискретно-событийного моделирования алгоритмов активного управления очередью. Описываются алгоритмы, позволяющие выполнять моделирование алгоритмов на языках высокого уровня.

Ключевые слова: дискретно-событийное моделирование, моделирование, управление переполнением, управление очередью, активное управление очередью, RED, AGRED, буфер, размер буфера, правила трафика, язык программирования, JAVA, память роутера, управление памятью, планировщик пакетов

DOI: 10.17212/1814-1196-2015-1-135-147

* Статья получена 31 июля 2014 г.

ВВЕДЕНИЕ

Явление переполнения буферных устройств коммутационного оборудования является фундаментальным вопросом в высокоскоростных сетях передачи данных. Переполнение происходит в буфере маршрутизатора, когда входящие пакеты превышают доступные сетевые ресурсы (как, например, буферное пространство). Переполнение приводит к увеличению потерь пакетов, задержки в очереди и, наконец, ухудшает пропускную способность, делает невозможным создание перспективной высокоскоростной оптической коммутации [1, 5, 8, 16].

Алгоритмы обработки очередей являются одним из главных механизмов обеспечения необходимого уровня качества обслуживания. Различные алгоритмы нацелены на решения проблем и по-разному воздействуют на качество обслуживания сетью пакетов разных типов.

Алгоритмы обслуживания очередей позволяют устанавливать различный уровень QoS пакетам разных классов. Часто используется несколько очередей, которые управляют пакетами своего строго определенного класса. Требуется, чтобы наиболее приоритетные пакеты обрабатывались с меньшей задержкой, но при этом не занимали полностью полосу пропускания, одновременно с этим необходимо передавать и трафик более низкого приоритета.

Различные алгоритмы активного управления очередью (AQM) решают проблему переполнения разными способами. Один из способов – применение алгоритма произвольного раннего обнаружения (RED – Random Early Detection), который является одним из самых известных активных алгоритмов управления очередями и был рекомендован Internet Engineering Task Force (IETF). Недостаток данного алгоритма заключается в том, что нет возможности учитывать параметры качества обслуживания (QoS).

Упрощенный алгоритм произвольного раннего обнаружения (GRED) был предложен для решения в первую очередь проблемы с алгоритмом RED. Алгоритм GRED оценивали с использованием той же имитационной модели, что и для RED. Несмотря на то что GRED предлагали для преодоления ограничений RED [2], коэффициент потери пакетов был по-прежнему высок в ситуации переполнения. Адаптивный упрощенный алгоритм раннего произвольного обнаружения (AGRED) был предложен, чтобы получить более качественный результат в потере пакетов. Моделирование AGRED было реализовано с использованием языка программирования JAVA [1–3, 7, 19].

Существует несколько типов симуляторов для имитации: NetCracker, OMNeT++, NS-2, NS-3, OPNET от OPNETTechnologies и другие, у каждого из которых есть недостатки, такие как крупные интегрированные компоненты, стоимость лицензии, сложность и время обучения людей для работы с симулятором, необходимость внесения множества изменений. Вследствие чего AGRED [3] моделировалась с использованием языка программирования Java, детальное описание этапа моделирования приведено не будет. (Следует отметить, что «Для устранения выявленных недостатков у уже имеющихся программных продуктов предназначенных для моделирования работы сетей, было принято решение выполнить имитационное моделирование алгоритма AGRED на языке программирования высокого уровня JAVA, исходный код полученного программного продукта не входит в данное исследование».)

Данное исследование нацелено на разработку дискретной имитационной модели, управляемой событиями для моделирования алгоритма активного управления буфером. Дискретное имитационное моделирование необходимо для определения оптимальных параметров алгоритмов управления очередью. Выполнение данных операций является критичным при оптимизации алгоритмов. Данное исследование позволяет изменять параметры генерирования пакетов, оптимизировать параметры алгоритмов, подбирать оптимальные значения размера буфера маршрутизатора. Программный продукт, разработанный по итогам данного исследования, позволяет выполнять данные операции «на лету», без перезапуска программы.

Таким образом, возникает задача разработать алгоритм и среду моделирования на языке программирования высокого уровня JAVA для выполнения имитационного моделирования алгоритма под названием «адаптивный произвольный алгоритм раннего обнаружения».

1. АЛГОРИТМ RANDOM EARLY DETECTION

До введения технологии RED существовало два способа борьбы с переполнением очереди: PPD (Past Packet Drop) и EPD (Early Packet Drop). В первом случае при переполнении пакеты отбрасывались с конца, а во втором случае – с начала очереди [9, 11, 13, 14].

Одним из возможных подходов при решении проблемы перегрузки является алгоритм RED, позволяющий маршрутизатору отбрасывать пакеты, даже когда в очереди еще имеется место. Главной целью алгоритма RED является исключение ситуации, когда несколько TCP-потоков перегружаются почти одновременно и затем синхронно начинают процедуру восстановления. RED пытается увеличить число коротких перегрузок и избежать долгих. Задача RED заключается в том, чтобы сообщить отправителю о возможности перегрузки, и отправитель должен адаптироваться к этой ситуации.

Основной принцип работы алгоритма заключается в том, что когда размер очереди превышает заданное пороговое значение, прибывающий пакет отбрасывается с определенной вероятностью, зависящий от уровня превышения установленного порога.

Чаще всего существуют два пороговых значения \min_{th} , \max_{th} . Когда размер очереди превышает первый порог, вероятность отбрасывания пакета \max_p линейно возрастает от 0 до 100 % у второго порога. С увеличением очереди до второго порога возрастает вероятность отбрасывания прибывшего пакета, так как он не умещается в очереди, при этом вероятность отклонения пакета большего размера выше, чем у пакета меньшего размера [4]:

$$\begin{aligned}
 d(k) &= 0 \text{ если } k < \min_{th}, \\
 d(k) &= 1 \text{ если } k > \max_{th}, \\
 d(k) &= \max_p \left(\frac{k - \min_{th}}{\max_{th} - \min_{th}} \right) \text{ иначе.}
 \end{aligned}
 \tag{1}$$

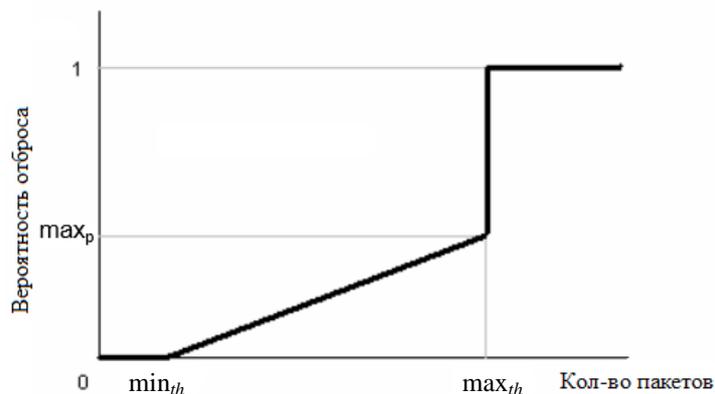


Рис. 1. Функция отброса пакета для RED

Примем следующие обозначения:

\max_{th} – верхний порог,

\min_{th} – нижний порог,

avg – средний размер очереди,

$Count$ – число пришедших пакетов со времени последнего сброса или со времени достижения нижней границы,

P_a – вероятность сброса пакета,

P_b – промежуточная вероятность,

\max_p – максимальная вероятность сброса пакета,

Q – текущая длина очереди,

Wq – весовой коэффициент для вычисления средней длины очереди,

$f()$ – некая линейная функция,

$time$ – текущее время,

q_time – время, когда последний раз очередь опустела.

Алгоритм обслуживания очереди RED тогда можно описать следующим образом:

```

Avg=0
Count=-1
Для каждого приходящего пакета {
  Count++
  Вычисление Avg
  If (minth<Avg<maxth) { Вычисление вероятности Pa}
  If (Avg>maxth) { Pa=1 }
  If (Avg<minth) { Pa=0; Count=-1}
  If (Rnd<Pa) то { сбросить пакет; Count=-1}
-}

```

Вычисление средней длины очереди происходит по алгоритму фильтра НЧ с экспоненциальным весом:

```

If (очередь не пуста) {
    avg=(1 - Wq) * avg + Wq * q
}
Else {
    m=f(time - q_time)
    avg=(1-Wq)m * avg
}

```

Алгоритм особенно эффективен при малых размерах очередей, так как он более устойчив к скачкообразному трафику [5].

2. АЛГОРИТМ AGRED

AGRED является одним из алгоритмов активного управления очередями и обнаружения начинающихся перегрузок в буфере маршрутизатора. Алгоритм AGRED состоит из двух отдельных алгоритмов, так же как алгоритм RED. Первый алгоритм используется для расчета средней длины очереди (AQL) и определяет степень сжатия, которая будет использована в очереди шлюза. Формула для расчета средней длины очереди:

$$aql = aql \cdot x(l - q_w) + q_w \cdot queueSize, \quad (2)$$

где q_w – вес очереди, $queueSize$ – мгновенный размер очереди.

Второй алгоритм применяется для вычисления вероятности маркировки пакета и определяет, как часто шлюз отмечает пакеты с учетом текущего уровня перегрузки. Цель для шлюза заключается в том, чтобы отметить пакеты через достаточно равномерные промежутки во избежание предвзятости, избежать глобальной синхронизации и помечать пакеты достаточно часто, чтобы контролировать среднюю длину очереди. Формула для расчета вероятности отклонения пакета [7]:

$$Dp = \frac{D_{init}}{t - c + D_{init}}, \quad (3)$$

где c – счетчик пакетов, пришедших в буфер маршрутизатора и не отброшенных с момента последнего отклоненного пакета, D_{init} рассчитывается из формулы (4):

$$D_{init} = D_{max} + (1 - D_{max}) \frac{aql - \max Threshold}{\max Threshold}. \quad (4)$$

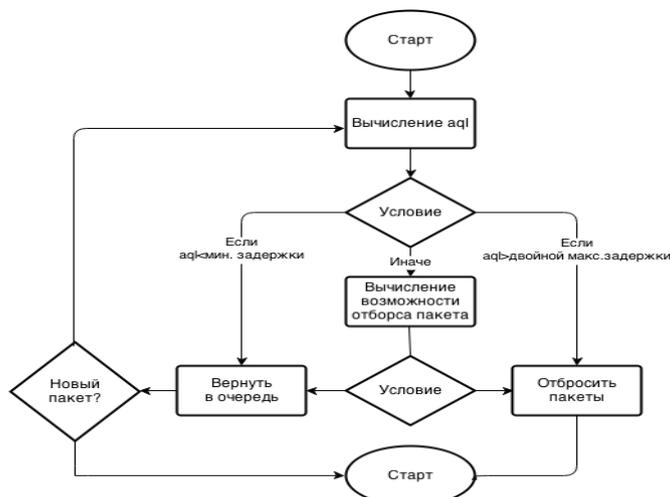


Рис. 2. Блок-схема алгоритма AGRED

На рис. 2 показана блок-схема, где AGRED определяет, будет пакет отклонен из очереди или добавлен в нее. Алгоритм вычисляет AQL для каждого приходящего пакета, а затем он сравнивает AQL с тремя порогами, представленными на рис. 3. Он продолжает добавлять пакеты в очередь при $AQL < \min Threshold$ (минимальный порог буфера) и отбрасывать все прибывающие пакеты, если $AQL > \min Threshold$. В противном случае AGRED необходимо вычислить вероятность отброса и решить, следует ли добавить или отклонять прибывающие пакеты, $\min Threshold < AQL < \max Threshold$ или если $\min Threshold < AQL < \max Threshold \cdot 2$.

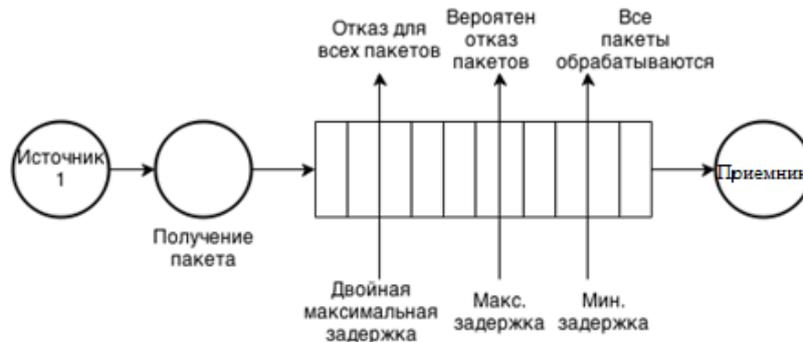


Рис. 3. Модель алгоритма AGRED

3. МОДЕЛИРОВАНИЕ АЛГОРИТМА AGRED

Модель алгоритма AGRED была реализована с использованием языка программирования общего назначения JAVA. Для генерирования пакетов применялся алгоритм геометрического распределения.

Последовательно проводится несколько независимых испытаний до появления некоторого события A , вероятность которого в каждом испытании

одна и та же и равна p . Тогда число X произведенных испытаний есть дискретная случайная величина с геометрическим распределением вероятности. Примером может служить стрельба по некоторой цели до первого попадания, причем вероятность попадания при каждом выстреле не зависит от результатов предыдущих выстрелов и сохраняет постоянное значение p . Число X произведенных выстрелов будет случайной величиной, возможные значения которой – натуральные числа [6, 20, 22]. Геометрический закон распределения задается формулой

$$P\{X = k\} = q^{k-1} p, \quad k = 1, 2, \dots, \quad \text{где } q = 1 - p. \quad (5)$$

Псевдокод геометрического распределения, используемый в данном моделировании, представлен на рис. 4.

```
public double geo(double mean) {
    Random r = new Random();
    double ran = r.nextDouble();
    return Math.ceil(Math.log(ran)/Math.log(1.0-(1/mean)));
}
```

Рис. 4. Псевдокод геометрического распределения

4. ХАРАКТЕРИСТИКИ ПРОИЗВОДИТЕЛЬНОСТИ

В компьютерных терминах показателями эффективности являются параметры, применяемые для оценки и анализа данного алгоритма, модели, программного обеспечения, такие как пропускная способность, задержка. В качестве показателя производительности в данном моделировании был выбран параметр потери пакетов PL. Потеря пакетов происходит, когда один или несколько пакетов не в состоянии достичь места назначения во время прохождения через компьютерные сети [4]. В данном исследовании величина потери пакетов определяется разностью количества полученных пакетов (nap) из числа отправленных пакетов (ndp):

$$Pl = nap - ndp. \quad (6)$$

В этом разделе описываются условия выполнения имитационного моделирования. Для проведения моделирования необходимо описать события, возникающие в системе, что также поможет в понимании статистического состава. Статистический состав системы – целостное представление системы, определяемое событиями. В данном исследовании статистическим составом является количество пакетов [8].

С точки зрения объектного подхода в имитационном моделировании сеть представляет собой совокупность сетевых объектов (Network Objects), каждый из которых способен определенным образом реагировать на некоторое множество событий (Events). Обо всех событиях сетевые объекты извещаются так называемым планировщиком (Scheduler), который содержит хронологическую таблицу событий. Каждое событие имеет обязательные атрибуты: время наступления и сетевой объект, к которому относится это событие. Планировщик по очереди (в хронологическом порядке) выбирает

из таблицы события и извещает соответствующие сетевые объекты о наступлении этих событий, сообщая также момент времени, в который произошло это событие. С точки зрения программиста, такое «извещение о событии» представляет собой просто вызов метода-обработчика для соответствующего объекта. Вся сопутствующая информация передается через параметры этого метода.

Помещать события в таблицу планировщика могут как сами сетевые объекты, так и пользователь при описании модели. В первом случае событие называется пакетом (так как по смыслу предметной области обмен пакетами – это и есть способ взаимодействия между сетевыми объектами), во втором – *at-событием*. Последние предназначены для управления ходом моделирования.

Таким образом, любое взаимодействие между сетевыми объектами осуществляется через посредство планировщика. Отправка пакета одним сетевым объектом другому есть не что иное, как постанова отправителем в таблицу события, адресованного получателю и помеченного тем моментом времени, когда пакет должен до этого получателя дойти (возможно, с учетом задержки при передаче). Получение пакета, соответственно, представляет собой извещение планировщиком получателя о событии и обработку этого события получателем.

При дискретном моделировании хронологическая последовательность событий представляет собой этапы работы системы. Событие представляет собой случай, который изменяет статистический состав системы, поскольку статистический состав определяется как количество пакетов, которое может увеличиваться или уменьшаться. Ниже представлен алгоритм *AGRED* демонстрации событий, которые влияют на количество пакетов. Описанные шаги алгоритма *AGRED* получены в виде событий.

1. Система запускается из состояния покоя. (Количество пакетов = 0, Размер буфера = 0, Статус приемника = Ожидание.)

2. Источник создает пакет.

3. Источник передает пакет.

1. Статус приемника меняется на «проверен».

4.1. Если приемник в состоянии ожидания, то пакет передается в него для обработки.

4.2. Если приемник занят/загружен, то направить пакет в буфер маршрутизатора, учитывая его доступность. (Вычисляется средняя длина очереди для передаваемого пакета (*AQL*).)

4.2.1. Если $AQL \geq$ двойному максимальному порогу то, пометить/отклонить пакет с параметром $Dp = 1$.

4.2.2. Проверить, если минимальный порог $\leq AQL <$ двойного максимального порога, то пометить/отклонять приходящие пакеты в зависимости от значения Dp на каждом из них.

4.2.3. Проверить, если $AQL <$ минимального порога (очередь свободна), поставить пакет в очередь.

5. Пакет, обработанный приемником, покидает систему.

5.1. Проверить размер буфера. Если размер буфера > 0 , взять пакет из буфера роутера и отправить его на обработку в приемник.

5.2. Если очередь пуста, установить состояние приемника в ожидание.

На шаге 2 источник генерирует пакеты: количество пакетов увеличивается и, таким образом, считается как событие (прибытие пакета), а на шаге 5 выход пакетов из системы рассматривается как событие за счет снижения количества пакетов (отправка пакетов).

Для генерирования пакетов используется геометрическое распределение, как и в AGRED. При моделировании геометрическое распределение представляет типы сгенерированных пакетов, которые существуют в реальности [9].

Планировщик является частью модели и компонентом, который выбирает событие/функцию на основе времени запуска. Другими словами, у событий нет предопределенного времени запуска. Единственная задача планировщика заключается в выборе события из списка событий. В этом проекте задача планировщика – это выбор между событием приема и отправки в короткий промежуток времени.

ЗАКЛЮЧЕНИЕ

В данной работе был разработан алгоритм для создания имитационной модели алгоритма AGRED на языке программирования JAVA. Данный симулятор использует геометрическое распределение для создания трафика. Разработанный алгоритм позволяет достаточно просто подобрать оптимальные значения ключевых параметров при дискретном имитационном моделировании, так как позволяет путем изменения параметров моделируемого алгоритма с течением времени подобрать оптимальные значения в пределах одного запуска. Данный алгоритм может быть проверен с помощью эксперимента, основанного на среднем времени приема пакетов.

Разработанный алгоритм также можно использовать для моделирования протоколов GRED, RED, а также алгоритмов, основанных на экспоненциальном распределении, но для этого потребуется немного изменить разработанный алгоритм.

СПИСОК ЛИТЕРАТУРЫ

1. *Барабанова Е.А., Мальцева Н.С., Барабанов И.О.* Алгоритм работы буферного устройства АТМ-коммутатора с параллельной настройкой // Вестник Астраханского государственного технического университета. Серия: Управление, вычислительная техника и информатика. – 2012. – № 1. – С. 97–103.
2. *Барабанова Е.А., Мальцева Н.С., Барабанов И.О.* Разработка универсального алгоритма для многокаскадных коммутаторов с параллельной настройкой // Вестник Астраханского государственного технического университета. Серия: Управление, вычислительная техника и информатика. – 2013. – № 1. – С. 118–125.
3. *Дмитриев В.Н., Тушинов А.С., Сергеева Е.В.* Имитационное моделирование системы мониторинга многозвенной сети передачи данных // Вестник Астраханского государственного технического университета. Серия: Управление, вычислительная техника и информатика. – 2013. – № 2. – С. 86–91.
4. *Квятковская И.Ю., Фам К.Х.* Система показателей оценки качества телекоммуникационных услуг и методы их оценки // Вестник Астраханского государственного технического университета. Серия: Управление, вычислительная техника и информатика. – 2013. – № 2. – С. 98–103.

5. Performance assessment of AGRED, RED and GRED congestion control algorithms / M. Baklizi, H. Abdel-jaber, S. Ramadass, N. Abdullah, M. Anbar // *Information Technology Journal*. – 2012. – Vol. 11, iss. 2. – P. 255–261. – doi: 10.3923/itj.2012.255.261.
6. Performance investigations of some active queue management techniques using simulation / H. Abdel-jaber, F. Thabtah, A.M. Daoud, J. Ababneh // *International Journal of New Computer Architectures and Their Applications (IJNCAA)*. – 2012. – Vol. 2, no. 1. – P. 286–301.
7. Floyd S. Recommendation on using the "gentle" variant of RED [Electronic resource]. March 2000. – URL: <http://www.aciri.org/floyd/red/gentle.html> (accessed: 06.03.2015).
8. Krifa A., Barakat C., Spyropoulos T. Optimal buffer management policies for delay tolerant networks // *Proceedings of 5th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, 16–20 June 2008, SECON'08. – San Francisco, California, 2008. – P. 260–268. – doi: 10.1109/SAHCN.2008.40.
9. Floyd S., Jacobson V. Random early detection gateways for congestion avoidance // *IEEE/ACM Transactions on Networking (TON)*. – 1993. – vol. 1, iss. 4. – P. 397–413. – doi: 10.1109/90.251892.
10. Барабанова Е.А., Мальцева Н.С., Барабанов И.О. Алгоритм параллельной обработки данных в оптических сетях // *Научный вестник НГТУ*. – 2014. – № 3 (56). – С. 88–95.
11. Optical packet buffers for backbone internet routers / N. Beheshti, E. Burmeister, Y. Ganjali, J. Bowers, D. Blumenthal, N. McKeown // *IEEE/ACM Transactions on Networking (TON)*. – 2010. – Vol. 18, iss. 5. – P. 1599–1609. – doi: 10.1109/TNET.2010.2048924.
12. Lin D., Hamdi M., Muppala J. Designing packet buffers using random round robin // *Proceedings of IEEE Global Telecommunications Conference (GLOBECOM 2010)*, 6–10 December 2010. – Miami, Florida, USA, 2010. – P. 1–5. – doi: 10.1109/GLOCOM.2010.5683309.
13. Kabra M., Saha S., Lin B. Fast buffer memory with deterministic packet departures // *Proceedings of 14th IEEE Symposium on High-Performance Interconnects (HOTI 2006)*, 23–25 August 2006. – Stanford, California, 2006. – P. 67–72. – doi: 10.1109/HOTI.2006.13.
14. Reddy T.B., Ahammed A., Banu R. Performance comparison of active queue management techniques // *International Journal of Computer Science and Network Security*. – 2009. – Vol. 9, N 2. – P. 405–408.
15. Spyropoulos T., Psounis K., Raghavendra C.S. Efficient routing in intermittently connected mobile networks: the multiple-copy case // *IEEE/ACM Transactions on Networking*. – 2008. – Vol. 16, iss. 1. – P. 63–76. – doi: 10.1109/TNET.2007.897962.
16. Preemptive virtual clock: a flexible, efficient, and cost-effective QoS scheme for networks-on-chip / B. Grot, S.W. Keckler, O. Mutlu // *Proceedings of 42nd Annual IEEE/ACM International Symposium on Microarchitecture MICRO-42*, 12–16 December 2009. – New York, USA, 2009. – P. 268–279.
17. Albert R., Barabasi A.-L. Statistical mechanics of complex networks // *Reviews of Modern Physics*. – 2002. – Vol. 74, iss. 1. – P. 47–97. – doi: 10.1103/RevModPhys.74.47.
18. Adaptive backpressure: efficient buffer management for on-chip networks / D.U. Becker, Nan Jiang, G. Michelogiannakis, W.J. Dally // *Proceedings of IEEE 30th International Conference on Computer Design (ICCD 2012)*, 30 September–3 October 2012. – Montreal, Quebec, Canada, 2012. – P. 419–426. – doi: 10.1109/ICCD.2012.6378673.
19. Fathima G., Wahidabanu R.S.D. A new queuing policy for delay tolerant networks // *International Journal of Computer Applications*. – 2010. – Vol. 1, N 20. – P. 56–60. – doi: 10.5120/422-622.
20. Routing for disruption tolerant networks: taxonomy and design / T. Spyropoulos, R.N. Bin Rais, T. Turlatti, K. Obraczka, A. Vasilakos // *Wireless Networks*. – 2010. – Vol. 16, iss. 8. – P. 2349–2370. – doi: 10.1007/s11276-010-0276-9.
21. Padmavathi G., Chitra K. Classification and performance of AQM-based schemes for congestion avoidance // *International Journal of Computer Science and Information Security*. – 2010. – Vol. 8, N 1. – P. 331–340.
22. Thiruchelvi G., Raja J. A survey on active queue management mechanisms // *International Journal of Computer Science and Network Security*. – 2008. – Vol. 8, N 12. – P. 130–145.

Барabanова Елизавета Александровна, кандидат технических наук, доцент кафедры «Связь» Астраханского государственного технического университета. Основное направление научных исследований – перспективные высокоскоростные коммутационные системы. Имеет более 30 публикаций, в том числе 2 монографии. E-mail: elizavetaaleb@yandex.ru

Береснев Илья Александрович, аспирант кафедры «Связь» Астраханского государственного технического университета. Основное направление научных исследований – перспективные направления высокоскоростной коммутации. Имеет более 10 публикаций. E-mail: gold1290@mail.ru

*Discrete simulation of the queuing algorithm in the router buffer**

E.A. BARABANOVA¹, I.A. BERESNEV²

¹Astrakhan State Technical University, 16 Tatichshev St. 414025, Astrakhan, Russian Federation, PhD (Eng.) associate professor. E-mail: ElizavetaAlexB@yandex.ru

²Astrakhan State Technical University, 16 Tatichshev St. 414025, Astrakhan, Russian Federation, postgraduate student. E-mail: gold1290@mail.ru, ili4.prizrak@gmail.com

Uneven growth rates of data transmission channels inevitably lead to “narrow” places in the communications network and thus to congestions, especially when connecting access level networks to the wide area network. Traditional protocols of queue management and congestion avoidance cannot cope with high density traffic management and nonlinearly changing load, which leads to congestions. This also causes a decrease in the effective rate of data transmission and degrades the quality parameters such as the percentage of lost packets, delays and jitters. A passive mechanism for resetting the excess data packets overflow queue is mostly used in data networks. When sending the TCP traffic, this mechanism gives rise to the phenomenon of global synchronization. New mechanisms have been developed for early prevention of queue overflows and early detection of overloads. This article describes the principles of building a simulation model for active queue management algorithms. Algorithms and program listings performing discrete-event simulation are proposed. These algorithms are used to perform simulation using high-level programming languages. Increasing complexity of modern telecommunication technologies creates a problem in the application of analytical techniques to evaluate the performance of designed systems and networks, which results in an extensive use of simulation. To prevent overloading in the nodes of packet-switched networks active queue management algorithms are used in the paper. In the relevant service system which simulates the work of a network node, each incoming packet can be dropped with a certain probability, even if the buffer is not fully filled yet. A probability to be dropped depends on the length of the queue at the time of receipt of the package. The paper presents the principles of discrete-event simulation algorithms of active queue management. The algorithms of simulation in high-level programming languages are described.

Keywords: discrete event simulation, modeling, overflow control, queue management, active queue management, RED, AGRED, buffer, buffer size, traffic policy, programming language, JAVA, router memory, memory policy, packet scheduling

DOI: 10.17212/1814-1196-2015-1-135-147

REFERENCES

1. Barabanova E.A., Mal'ceva N.S., Barabanov I.O. Algoritm raboty buferного ustrojstva ATM-kommutatora s parallel'noj nastrojkoj [Algorithm of buffer device working of ATM-switch with parallel adjustment]. *Vestnik Astrahanskogo gosudarstvennogo tehničeskogo universiteta. Seri-*

* Received 31 July 2014.

ja: Upravljenje, vychislitel'naja tehnika i informatika – Vestnik of Astrakhan state technical university. Series: Management, Computer science and Informatics, 2012, no. 1, pp. 97–103.

2. Barabanova E.A., Mal'ceva N.S., Barabanov I.O. Razrabotka universal'nogo algoritma dlja mnogokaskadnyh kommutatorov s parallel'noj nastrojkoj [Development of universal algorithm for multistage switches with parallel adjustment]. Vestnik Astrahanskogo gosudarstvennogo tehničeskogo universiteta. Serija: Upravljenje, vychislitel'naja tehnika i informatika – Vestnik of Astrakhan state technical university. Series: Management, Computer science and Informatics, 2013, no. 1, pp. 118–125.

3. Dmitriev V.N., Tushnov A.S., Sergeeva E.V. Imitacionnoe modelirovanie sistemy monitoringa mnogozvennoj seti peredachi dannyh [Simulation of monitoring multilink data transmission network]. Vestnik Astrahanskogo gosudarstvennogo tehničeskogo universiteta. Serija: Upravljenje, vychislitel'naja tehnika i informatika – Vestnik of Astrakhan state technical university. Series: Management, Computer science and Informatics, 2013, no. 2, pp. 86–91.

4. Kvjatkovskaja I.Ju., Fam K.H. Sistema pokazatelej ocenki kachestva telekommunikacionnyh uslug i metody ih ocenki [System of criteria of evaluation of quality of telecommunication services and method of their evaluation]. Vestnik Astrahanskogo gosudarstvennogo tehničeskogo universiteta. Serija: Upravljenje, vychislitel'naja tehnika i informatika – Vestnik of Astrakhan state technical university. Series: Management, Computer science and Informatics, 2013, no. 2, pp. 98–103.

5. Baklizi M., Abdel-jaber H., Ramadass S., Abdullah N., Anbar M. Performance assessment of AGRED, RED and GRED congestion control algorithms. *Information Technology Journal*, 2012, vol. 11, iss. 2, pp. 255–261. doi: 10.3923/itj.2012.255.261

6. Abdel-jaber H., Thabtah F., Daoud A.M., Ababneh J. Performance investigations of some active queue management techniques using simulation. *International Journal on New Computer Architectures and Their Applications (IJNCAA)*, 2012, vol. 2, no. 1, pp. 286–301.

7. Floyd S. Recommendation on using the “gentle” variant of RED. March 2000. Available at: <http://www.aciri.org/floyd/red/gentle.html> (accessed 06.03.2015)

8. Krifa A., Barakat C., Spyropoulos T. Optimal buffer management policies for delay tolerant networks. *Proceedings of 5th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks, SECON'08*, San Francisco, California, 16–20 June 2008, pp. 260–268. doi: 10.1109/SAHCN.2008.40

9. Floyd S., Jacobson V. Random early detection gateways for congestion avoidance. *IEEE/ACM Transactions on Networking (TON)*, 1993, vol. 1, iss. 4, pp. 397–413. doi: 10.1109/90.251892

10. Barabanova E.A., Mal'ceva N.S., Barabanov I.O. Algoritm parallel'noj obrabotki dannyh v opticheskix setjah [The algorithm of parallel data processing in optical networks]. *Nauchnyi vestnik Novosibirskogo gosudarstvennogo tehničeskogo universiteta – Science Bulletin of the Novosibirsk State Technical University*, 2014, no. 3 (56), pp. 88–95.

11. Beheshti N., Burmeister E., Ganjali Y., Bowers J., Blumenthal D., McKeown N. Optical packet buffers for backbone internet routers. *IEEE/ACM Transactions on Networking (TON)*, 2010, vol. 18, iss. 5, pp. 1599–1609. doi: 10.1109/TNET.2010.2048924

12. Lin D., Hamdi M., Muppala J. Designing packet buffers using random round robin. *Proceedings of IEEE Global Telecommunications Conference (GLOBECOM 2010)*, Miami, Florida, USA, 6–10 December 2010, pp. 1–5. doi: 10.1109/GLOCOM.2010.5683309

13. Kabra M., Saha S., Lin B. Fast buffer memory with deterministic packet departures. *Proceedings of 14th IEEE Symposium on High-Performance Interconnects (HOTI 2006)*, Stanford, California, 23–25 August 2006, pp. 67–72. doi: 10.1109/HOTI.2006.13

14. Reddy T.B., Ahammed A., Banu R. Performance comparison of active queue management techniques. *International Journal of Computer Science and Network Security*, 2009, vol. 9, no. 2, pp. 405–408.

15. Spyropoulos T., Psounis K., Raghavendra C.S. Efficient routing in intermittently connected mobile networks: the multiple-copy case. *IEEE/ACM Transactions on Networking*, 2008, vol. 16, iss. 1, pp. 63–76. doi: 10.1109/TNET.2007.897962

16. Grot B., Keckler S.W., Mutlu O. Preemptive virtual clock: a flexible, efficient, and cost-effective QOS scheme for networks-on-chip. *Proceedings of 42nd Annual IEEE/ACM International Symposium on Microarchitecture MICRO-42*, New York, USA, 12–16 December 2009, pp. 268–279.

17. Albert R., Barabasi A.-L. Statistical mechanics of complex networks. *Reviews of Modern Physics*, 2002, vol. 74, iss. 1, pp. 47–97. doi: 10.1103/RevModPhys.74.47
18. Becker D.U., Nan Jiang, Michelogiannakis G., Dally W.J. Adaptive backpressure: efficient buffer management for on-chip networks. *Proceedings of IEEE 30th International Conference on Computer Design (ICCD 2012)*, Montreal, Quebec, Canada, 30 September–3 October 2012, pp. 419–426. doi: 10.1109/ICCD.2012.6378673
19. Fathima G., Wahidabanu R.S.D. A new queuing policy for delay tolerant networks. *International Journal of Computer Applications*, 2010, vol. 1, no. 20, pp. 56–60. doi: 10.5120/422-622
20. Spyropoulos T., Bin Rais R.N., Turletti T., Obraczka K., Vasilakos A. Routing for disruption tolerant networks: taxonomy and design. *Wireless Networks*, 2010, vol. 16, iss. 8, pp. 2349–2370. – doi: 10.1007/s11276-010-0276-9
21. Padmavathi G., Chitra K. Classification and performance of AQM-based schemes for congestion avoidance. *International Journal of Computer Science and Information Security*, 2010, vol. 8, no. 1, pp. 331–340.
22. Thiruchelvi G., Raja J. A survey on active queue management mechanisms. *International Journal of Computer Science and Network Security*, 2008, vol. 8, no. 12, pp. 130–145.