

## Численный анализ усовершенствованных алгоритмов ограничения доступа в системе $GI/G/1/N^1$

М.Г. Коновалов, Р.В. Разумчик

Федеральный исследовательский центр “Информатика и управление” Российской академии наук,  
Москва, Россия, Российский университет дружбы народов, Москва, Россия

Поступила в редколлегию 13.10.2017

**Аннотация**—Статья посвящена описанию и экспериментальному анализу новых стратегий активного управления очередью. Конструкция стратегий основана на развитии идеи функции сброса, которая используется в классической серии алгоритмов ограничения доступа RED. Используются три модификации функции сброса: с рандомизацией параметров, с отказом от структурирования и с учетом времени выполнения заданий. В качестве модели объекта управления выбрана система массового обслуживания  $G/G/1/N$ , для которой приведены результаты численных экспериментов. Во всех экспериментах эффективность стратегий (по различным критериям) возрастала по мере внедрения перечисленных усовершенствований. Преимущество новых алгоритмов особенно проявилось в случаях большой нагрузки.

**КЛЮЧЕВЫЕ СЛОВА:** активное управление очередью; алгоритмы ограничения доступа; функция сброса.

### 1. ВВЕДЕНИЕ

В этой статье речь пойдет об одном аспекте проблемы активного управления очередями (далее — AQM, Active Queue Management) в сетевых узлах — повышении эффективности функционирования алгоритмов подобных RED (Random Early Detection). Судя по публикациям в научной печати, в этой области уже накоплен большой объем научных знаний (включая разнообразные модели AQM-механизмов) и некоторый опыт их применения на практике (см., например, [1–4]). Однако несмотря на то, что проблема уже давно изучается, универсального алгоритма, который с важнейших (для оператора сети связи) точек зрения успешно справлялся бы с ней, до сих пор не построен. Об этом свидетельствует последняя рекомендация RFC 7567 (2015 г.) комитета IETF (см. [5]). Помимо освещения текущего состояния решения проблемы разработки AQM-механизмов, в [5] также приводится перечень актуальных направлений дальнейших исследований (см. [5, раздел 4]). Одно из них — это разработка новых, более эффективных по сравнению с известными, правил активного управления очередями. Именно к этой области относится исследование, некоторые результаты которого представлены в следующих разделах.

В качестве простой модели объекта управления выбрана система с одним входящим потоком, одной очередью ограниченной емкости и одним сервером, обслуживающим заявки по одной, без прерывания и в порядке поступления. Для такой системы, которая может быть моделью некоторых узлов связи (см., например, [3]), известно множество AQM-механизмов, среди которых самые известные представители — это алгоритмы типа RED: RED, GRED, REM, NI и др. Для алгоритмов такого типа известно несколько аналитических результатов

<sup>1</sup> Работа выполнена при поддержке Российского фонда фундаментальных исследований (грант 15-07-03406).

(см., например, [1, 3, 7–11] и ссылки в них), полученных для таких вероятностно-временных характеристик, как моменты числа последовательных потерь, моменты числа заявок, время ожидания которых превосходит наперед заданное значение. Получение аналитических результатов существенно осложняется при усложнении модели, в частности, при отказа от экспоненциальности распределений. При сравнении AQM-алгоритмов аналитические приемы и вовсе оказываются бессильными и приходится обращаться к статистическому моделированию.

Взяв классический алгоритм RED за эталон (его описание будет приведено в следующем разделе), сформулируем следующий вопрос: есть ли в классе RED-подобных механизмов механизм лучше RED с точки зрения вероятности потерь, средней длины очереди и ее дисперсии? Оказывается, что ответ положителен уже при простейшем обобщении, когда параметры исходного алгоритма RED рассматриваются как случайные величины. Дополнительный эффект можно получить, если отказаться от структурированности функции сброса, а также если учитывать при сбросе значение оставшегося времени обслуживания. Все эти выводы, сделанные по результатам численных экспериментов, составляют основное содержание статьи.

## 2. ПОСТАНОВКА ЗАДАЧИ

Рассматривается обслуживающий прибор (сервер), на который поступает рекуррентный входной поток заявок на выполнение заданий. Независимые промежутки времени между поступлениями заявок характеризуются функцией распределения  $F$ . Выполнение заданий происходит на единственном месте обслуживания, причем время выполнения каждого задания не зависит от остальных величин и имеет распределение  $G$ . В системе имеется также бункер из  $N$  мест ожидания, в котором может образовываться очередь, обслуживаемая по принципу FIFO. Всего в системе может находиться, самое большее,  $N + 1$  задание. Выполнение заданий, присутствующих в системе  $\mu = \mu(F, G, N)$  (в обозначениях теории массового обслуживания — это система  $G/G/1/N$ ), осуществляется «без перерыва». Это означает, во-первых, что выполнение задания, раз начавшись, не прерывается. И, во-вторых, если в момент окончания обслуживания какого-либо задания очередь не пуста, то в этот же момент начинается выполнение следующего задания. В момент поступления заявки принимается решение о том, будет ли задание принято на обслуживание или же заявка будет «сброшена» (потеряна). Если все места ожидания заняты, то заявка сбрасывается автоматически. В ином случае решение зависит от принятого в системе алгоритма обработки поступающих заявок. Этот алгоритм, или стратегия управления очередью (размером очереди) является предметом обсуждения.

Система функционирует в непрерывном времени,  $t \geq 0$ . Пусть  $0 \leq \tau_1 < \tau_2 < \dots$  — моменты поступления заявок в систему, которая изначально была пустая, и пусть  $y_n$  — индикатор события {задание, поступившее в момент  $\tau_n$ , принято на обслуживание}. Стратегией управления очередью в системе  $\mu$  назовем последовательность случайных величин  $y = \{y_n\}$ . Задание такой последовательности определяет вероятность  $P$  в пространстве траекторий процесса, в частности, последовательность условных вероятностей  $\sigma = \{\sigma_n\}$ , где  $\sigma_n = \mathbf{P}(y_n = 1 | \cdot)$  — вероятность «сброса» заявки в зависимости от предыстории. Задание последовательности  $\sigma$  эквивалентно заданию последовательности  $y$  в том смысле, что каждая из них порождает один и тот же процесс. Однако в конструктивном отношении гораздо удобнее иметь дело с последовательностью  $\sigma$ . Поэтому, говоря о стратегии, будем иметь в виду именно последовательность  $\sigma$ . В частности, будем писать  $\mathbf{P} = \mathbf{P}_{\mu, \sigma}$ , отражая зависимость вероятности от параметров системы и от выбранной стратегии. На выбор компонент  $\sigma_n$  — правил управления — накладывается естественное ограничение, что они должны зависеть только от наблюдаемой предыстории до момента  $\tau_t$ , в остальном этот выбор может быть произвольным. В свою очередь, ограничений на наблюдаемость не предполагается.

Для того чтобы иметь возможность обсуждать предпочтения среди различных стратегий, необходимо выбрать критерии оценки их качества. В данном случае рассматриваются, главным образом, два показателя, часто используемые в литературе по дисциплинам обслуживания очереди — это вероятность потерь и средняя длина очереди. Предельная вероятность потерь определяется в общем случае как

$$q = q(\mu, \sigma) = \lim_{n \rightarrow \infty} \mathbf{P}_{\mu, \sigma}(y_n = 1),$$

а предельная средняя длина очереди определяется как

$$l = l(\mu, \sigma) = \lim_{n \rightarrow \infty} \mathbf{M}_{\mu, \sigma}(\nu(t) + 1),$$

где  $\nu(t)$  — число заданий в бункере в момент  $t$ ,  $\mathbf{M}_{\mu, \sigma}$  — знак математического ожидания. Поскольку все рассматриваемые далее стратегии стационарны в том смысле, что начиная с некоторого  $n < \infty$  все правила  $\sigma_n$  одинаковы, то пределы во всех рассматриваемых далее случаях существует. В качестве дополнительного показателя, важного в приложениях будем использовать также предельное среднеквадратичное отклонение длины очереди, которое определяется аналогичным образом. По смыслу проблемы управления размером очереди ясно, что все три критерия требуют минимизации. При этом, по крайней мере, два из них — вероятность отказа и средняя длина очереди — являются конкурирующими, поскольку уменьшение  $q$  влечет увеличение  $l$  и наоборот. В математической теории многокритериальность разрешается путем постановки специальных оптимизационных задач. Наиболее распространенными являются условная оптимизация, а также безусловная оптимизация единственной целевой функции (свертки), которая конструируется из имеющихся критериев. В то же время на практике часто действуют по-другому: вначале создают алгоритмы, а затем описывают их характеристики по всему спектру критериев, предоставляя пользователю самостоятельно сделать выбор. В следующих разделах статьи будет использован своего рода смешанный подход. Стратегии будут находиться в результате решения нижеперечисленных оптимизационных задач, составленных на базе двух критериев, вероятности отказа и средней длины очереди, а среднеквадратичное отклонение длины очереди будет выступать как дополнительная характеристика. Оптимизационные задачи формулируются следующим образом.

Пусть  $\mu$  — сервер с очередью, а  $\Sigma$  — некоторое заданное подмножество допустимых стратегий.

*Задача оптимизации 1* (минимум потерь при ограничении на среднюю очередь). Пусть  $L > 0$  — заданное число. Найти стратегию  $\sigma^{(1)}$  такую, что

$$q(\mu, \sigma^{(1)}) = \min_{\sigma \in \Sigma} q(\mu, \sigma) \quad \text{и} \quad l(\mu, \sigma^{(1)}) \leq L.$$

*Задача оптимизации 2* (минимальная очередь при ограничении на вероятность потерь). Пусть  $Q > 0$  — заданное число. Найти стратегию  $\sigma^{(2)}$  такую, что

$$L(\mu, \sigma^{(2)}) = \min_{\sigma \in \Sigma} L(\mu, \sigma) \quad \text{и} \quad q(\mu, \sigma^{(2)}) \leq Q.$$

*Задача оптимизации 3* (минимум свертки двух критериев). Пусть целевая функция задана в виде  $w(\mu, \sigma) = q(\mu, \sigma) + Cl(\mu, \sigma)$ , где  $C > 0$  — фиксированное число. Найти стратегию  $\sigma^{(3)}$  такую, что  $w(\mu, \sigma^{(3)}) = \min_{\sigma \in \Sigma} w(\mu, \sigma)$ .

### 3. АЛГОРИТМЫ ОГРАНИЧЕНИЯ ДОСТУПА

Самое простое, что можно сделать в плане регулирования очереди — это не вмешиваться в процесс накопления заданий. В этом случае бункер автоматически заполняется по мере

поступления заданий, пока есть свободные места, а потери имеют место только в моменты переполнения. Будем говорить о таком поведении, используя термин «пассивная стратегия» и обозначение  $\sigma_*$ .

Широкий класс простых алгоритмов управления размером очереди описывается с помощью так называемой функции сброса (dropping function), которая применительно к рассматриваемой модели представляет собой вектор  $p = (p_1, \dots, p_N)$ , где  $p_i$  — вероятность «сброса» заявки, если в момент ее поступления в системе находится  $i$  заданий. Отметим, что количество заданий в системе в момент  $t$  равняется  $\nu(t) + 1$ , если  $\nu(t) > 0$ , но оно может равняться как 0, так и 1, если  $\nu(t) = 0$ . Стратегию, порождаемую вектором  $p$ , обозначим через  $\sigma_{DF} = \sigma_{DF}(p)$ , а множество всех таких стратегий обозначим через  $\Sigma_{DF}$ .

Как отмечалось во введении, существует много модификаций для функции сброса и большинство из них строятся следующим образом. На отрезке  $0 \leq x \leq N$  задается некоторая аппроксимирующая «структурная функция»  $\tilde{p}(x)$ ,  $0 < \tilde{p}(x) \leq 1$ . Эта функция чаще всего выбирается кусочно-линейной, но существуют и нелинейные варианты. Затем подбираются параметры, определяющие конкретный вид функции  $\tilde{p}(x)$ . Компоненты вектора  $p$  определяются как  $p_i = \tilde{p}(i)$ . Примером может служить однопараметрическая структурная функция ступенчатого вида,

$$\tilde{p}(x) = \begin{cases} 0, & \text{если } 0 \leq x < T, \\ 1, & \text{если } T \leq x \leq N, \end{cases}$$

которая задает алгоритм порогового типа (с порогом  $T$ ) для сброса заявок. В качестве еще одного примера рассмотрим структурную функцию  $\tilde{p}(x)$ , которая порождает алгоритм RED — один из наиболее известных в рассматриваемой области:

$$\tilde{p}_{RED}(x) = \begin{cases} 0, & \text{если } 0 \leq x < a, \\ \frac{b(x-a)}{N-a}, & \text{если } a \leq x \leq N, \end{cases}$$

где  $0 < a < N$ ,  $0 \leq b \leq 1$ . Согласно стратегии  $\sigma_{RED} = \sigma_{RED}(a, b)$ , определенной с помощью этой структурной функции, допуск в систему новой заявки организован следующим образом. Если имеющееся количество заявок в системе меньше чем  $a$ , то новая заявка принимается. Если количество заявок в системе,  $i$ , находится в диапазоне от  $a$  до  $N$ , то заявка принимается с вероятностью  $1 - \tilde{p}_{RED}(i)$ . Если в системе заняты все места для ожидания, то заявка теряется. Вероятность отказа изменяется, таким образом, кусочно-линейно в диапазоне от 0 до  $b$  в зависимости от загруженности системы. Множество всех двухпараметрических стратегий  $\sigma_{RED} = \sigma_{RED}(a, b)$  обозначим через  $\Sigma_{RED}$ . Аналогичным образом устроены упомянутые во введении алгоритмы GRED, REM, NI. Наряду с алгоритмами  $\sigma_{RED}(a, b)$  рассмотрим некоторое их обобщение, которое заключается в том, что параметры  $a$  и  $b$  структурной функции  $\tilde{p}_{RED}(x)$  являются случайными величинами с распределениями соответственно  $q_a$  и  $q_b$  ( $q_a$  — распределение на интервале  $0 < a < N$ , а  $q_b$  — распределение на отрезке  $0 \leq b \leq 1$ ). Управление осуществляется таким образом, что каждый раз, когда с помощью функции  $\tilde{p}_{RED}(x)$  определяется вероятность отказа, в качестве значений параметров  $a$  и  $b$  выбираются реализации соответствующих случайных величин. Такие стратегии будем обозначать через  $\bar{\sigma}_{RED} = \bar{\sigma}_{RED}(q)$ ,  $q = (q_a, q_b)$ , а всю их совокупность — через  $\bar{\Sigma}_{RED}$ .

Ясно, что семейство алгоритмов RED «беднее», чем совокупность рандомизированных алгоритмов RED, а та, в свою очередь, является строгим подмножеством всех стратегий, построенных на основе функции сброса, то есть  $\Sigma_{RED} \subset \bar{\Sigma}_{RED} \subset \Sigma_{DF}$ . Возникает вопрос, можно ли улучшить показатели эффективности, выйдя за рамки множества  $\Sigma_{RED}$ . В следующем разделе на ряде примеров дан положительный ответ на этот вопрос. Для этой цели с помощью имитационной модели сравниваются показатели эффективности для стратегий, являющихся

приближенными решениями оптимизационных задач 1–3 с конкретными объектами  $\mu$  и различными множествами допустимых стратегий.

Аналогичная схема применена также для выяснения вопроса о том, можно ли повысить эффективность, расширив само множество  $\Sigma_{DF}$ . Рассматриваются «обобщенные функции сброса» вида

$$P = \{p(\tau), \tau > 0\},$$

где  $p(\tau) = (p_1(\tau), \dots, p_N(\tau))$ , а  $p_i(\tau)$  — вероятность сброса заявки при условии, что в момент ее поступления в системе было  $i$  заданий, причем выполняемое задание будет занимать место обслуживания еще время  $\tau$ . Таким образом, предлагается считать вероятность отказа зависящей не только от загруженности системы, но и от оставшегося времени до выполнения задания, находящегося «в работе». Стратегии такого вида обозначаются как  $\sigma_{DF} = \sigma_{DF}(P)$ , а совокупность всех таких стратегий обозначается символом  $\Sigma_{GDF}$ .

Следует сделать замечание относительно решения сформулированных в п. 2 оптимизационных задач. В аналитическом виде получить ответ на них в настоящее время не представляется возможным. Поэтому применяются опять-таки имитационная модель и алгоритмы адаптивной градиентной оптимизации, общую теорию которых можно найти в [12]. Детальное изложение этой части исследования не входит в план статьи, поэтому приближенные решения оптимизационных задач используются как заданные.

#### 4. ЭКСПЕРИМЕНТАЛЬНОЕ СРАВНЕНИЕ АЛГОРИТМОВ ОГРАНИЧЕНИЯ ДОСТУПА

В этом разделе содержится описание результатов трех серий экспериментов с имитационной моделью системы обслуживания  $\mu = \mu(F, G, N)$ , Размер бункера  $N = 9$ . Варьируются типы распределений  $F$  (распределение времени между поступлениями заявок) и  $G$  (распределение времени выполнения заданий), а также параметры этих распределений. Цель экспериментов — сравнить показатели различных типов стратегий, определенных в разделе 2, где они обозначены как  $\sigma^*$ ,  $\sigma_{RED}$ ,  $\bar{\sigma}_{RED}$ ,  $\sigma_{DF}$  и  $\sigma_{GDF}$ .

Пассивная стратегия  $\sigma^*$  означает, что никакого сброса заявок, кроме как вызванного переполнением бункера не происходит.

Параметры стратегии  $\sigma_{RED}$  взяты из работы [6]. Их значения одинаковы во всех экспериментах и равны  $a = 3,00008$ ,  $b = 0,71501$ . Параметры, очевидно, были подобраны специально для рассматриваемой системы, поэтому данная стратегия может служить определенным эталоном для сравнения с другими алгоритмами.

Параметры остальных трех стратегий назначаются в каждом эксперименте индивидуально, путем приближенного решения оптимизационных задач 1–3. В качестве множества допустимых стратегий в зависимости от типа стратегии выбирались определенные подмножества  $\sigma \subset \bar{\sigma}_{RED}$ ,  $\sigma \subset \sigma_{DF}$  или  $\sigma \subset \sigma_{GDF}$ . В первых двух примерах приводятся значения параметров приближенно оптимальных стратегий.

*Пример 1.* Распределение  $F$  является экспоненциальным с параметром  $\lambda$ . Распределение  $G$  — вырожденное, с атомом 1.

Сравним новые стратегии,  $\bar{\sigma}_{RED}$ ,  $\sigma_{DF}$  и  $\bar{\sigma}_{GDF}$  со стратегией  $\sigma_{RED}$  в шести экспериментах, в которых интенсивность входного потока  $\lambda$  принимает соответственно значения 0,5; 0,75; 1; 1,5; 2; 3. Значения параметров стратегий для различных  $\lambda$  приведены в таблицах 1–3. В этих и последующих таблицах пустые клетки означают, что данные значения несущественны (либо нули, либо соответствуют недостижимым при данной стратегии значениям) В таблице 1 указаны значения параметров стратегии  $\bar{\sigma}_{RED}$ , которые по определению представляют собой распределения  $q_a$  и  $q_b$  на интервале  $(0, N)$  и на отрезке  $[0, 1]$  соответственно. В данном случае эти распределения дискретны и в таблице 1 каждый вариант описан строкой вероятностей.

**Таблица 1.** Параметры стратегии  $\bar{\sigma}_{RED}$  в примере 1

значения $a$ и $b$	$q_a$									$q_b$			
	0	1	2	3	4	5	6	7	8	0	1/3	2/3	1
$\lambda = 0,5$	0,07	0,06	0,08	0,12	0,13	0,13	0,14	0,12	0,15	0,51	0,25	0,14	0,10
$\lambda = 0,75$			0,02	0,94	0,04					0,2			0,8
$\lambda = 1$				0,65	0,35							0,05	0,95
$\lambda = 1,5$				0,32	0,60	0,06		0,02					1
$\lambda = 2$	0,40	0,33	0,21	0,05									1
$\lambda = 3$	0,55	0,30	0,09	0,04	0,01	0,01							1

В таблице 2 содержатся параметры стратегии  $\sigma_{DF}$ . Эта стратегия определяется вектором  $p$  вероятностей отказа в зависимости от количества заданий в системе.

**Таблица 2.** Вероятности сброса для стратегии  $\sigma_{DF}$  в примере 1

длина очереди	2	3	4	5	6	7	8	9
$\lambda = 0,5$			0,03	0,35	0,48			
$\lambda = 0,75$			0,06	0,30	0,49	0,53		
$\lambda = 1$			0,02	0,07	0,91	0,85	0,65	0,54
$\lambda = 1,5$	0,03				0,53	1		
$\lambda = 2$	0,16	0,03	1					
$\lambda = 3$	0,48	0,45	0,37	0,08	0,02	1		

В таблице 3 приведены значения параметров стратегии  $\sigma_{GDF}$ . В общем случае стратегия задается набором вероятностей  $P$  сброса заявки в зависимости от размера очереди и от времени  $\tau$  до окончания выполнения задания, которое находится на месте обслуживания. В данном случае вероятность отказа определяется размером очереди и событием  $\tau \leq 0,5$ . Это означает, что вероятность отказа зависит от количества заявок в системе и от выполнения или невыполнения условия  $\tau \leq 0,5$ .

**Таблица 3.** Вероятности сброса для стратегии  $\sigma_{GDF}$  в примере 1

длина очереди	$\tau \leq 0,5$									$\tau > 0,5$								
	2	3	4	5	6	7	8	9	2	3	4	5	6	7	8	9		
$\lambda = 0,5$			0,01	0,27	0,47	1					0,09	0,41	1					
$\lambda = 0,75$			0,03	0,42	0,77	1					0,02	0,60	1					
$\lambda = 0,1$				0,03	0,98	1					0,01	0,09	0,98	1				
$\lambda = 1,5$					1									1				
$\lambda = 2$	0,16	0,04	0,28	0,65	0,96	0,86			0,13	0,04	0,65	0,81	0,92	1				
$\lambda = 3$	0,51	0,52	0,54	0,58	0,64	0,72	0,77	1	0,51	0,53	0,55	0,61	0,68	0,75	0,82	1		

Результаты имитации работы системы с описанными стратегиями сведены в таблицу 4.

На рисунке 1 на основе данных из таблицы 4 наглядно изображена относительная эффективность изучаемых стратегий. За единицу измерения приняты значения показателей для стратегии  $\sigma_{RED}$ . Высота столбиков означает отношение вероятности потерь (черные столбцы) и средней длины очереди (серые столбцы) для стратегий  $\bar{\sigma}_{RED}$ ,  $\sigma_{DF}$  и  $\sigma_{GDF}$  к аналогичным показателям стратегии  $\sigma_{RED}$ . Цифры 1, 2 и 3 относятся соответственно к стратегиям  $\bar{\sigma}_{RED}$ ,  $\sigma_{DF}$  и  $\sigma_{GDF}$ .

На верхних трех рисунках столбики, относящиеся к показателю средняя длина очереди имеют высоту, приблизительно равную 1. Это соответствует тому, что *оптимизационная задача 1* решалась при ограничении на среднюю длину очереди, равному значению этого показателя для стратегии  $\sigma_{RED}$ . (Некоторые отклонения связаны с погрешностью решения оптимизационной задачи.) В то же время отношение показателей вероятность потерь заметно меньше единицы для всех стратегий  $\bar{\sigma}_{RED}$ ,  $\sigma_{DF}$  и  $\sigma_{GDF}$ . Аналогичная картина имеет место для

Таблица 4. Показатели эффективности при различных стратегиях в примере 1

стратегия	$\sigma_{RED}$	$\bar{\sigma}_{RED}$	$\sigma_{DF}$	$\sigma_{GDF}$	$\sigma_{RED}$	$\bar{\sigma}_{RED}$	$\sigma_{DF}$	$\sigma_{GDF}$
				$\lambda = 0,5$				
процент потерь, %	0,199	0,169	0,147	0,139	9,128	9,052	8,786	8,742
средняя длина очереди	0,741	0,745	0,743	0,743	3,000	3,003	3,003	3,006
				$\lambda = 0,75$				
процент потерь, %	2,002	1,963	1,906	1,886	50,004	50,036	50,031	50,083
средняя длина очереди	1,575	1,581	1,580	1,576	7,146	5,551	4,354	4,144
				$\lambda = 1,5$				
процент потерь, %	33,516	33,535	33,490	33,515	66,667	66,672	66,670	66,674
средняя длина очереди	5,751	5,552	5,315	5,246	8,390	6,940	6,632	6,015

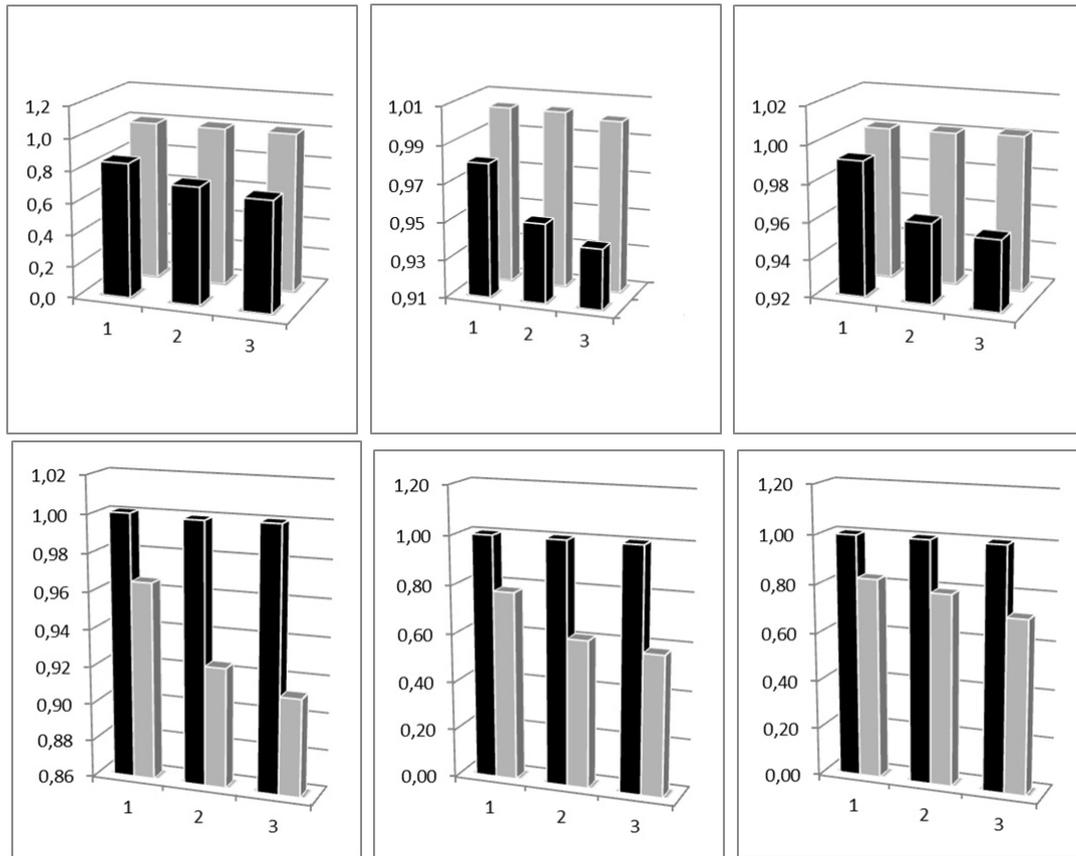


Рис 1. Сравнение показателей новых стратегий со стратегией  $\sigma_{RED}$  (1 — стратегия  $\bar{\sigma}_{RED}$ , 2 — стратегия  $\sigma_{DF}$ , 3 — стратегия  $\sigma_{GDF}$ ; черный цвет — вероятность потерь, серый цвет — средняя длина очереди)

нижних трех рисунков. В этих примерах стратегии определялись в результате решения оптимизационной задачи 2. Ограничения на вероятность потерь полагались равными значению этого показателя для стратегии  $\sigma_{RED}$ . Поэтому все четыре стратегии дают примерно равные вероятности потерь, но при этом три новые стратегии обеспечивают лучший результат по показателю средней длины очереди. Во всех примерах эффективность четырех рассмотренных стратегий ранжируется по схеме  $\sigma_{RED} \rightarrow \bar{\sigma}_{RED} \rightarrow \sigma_{DF} \rightarrow \sigma_{GDF}$  по основным показателям. Размер выигрыша варьируется в достаточно большом диапазоне и особенно заметен (до 40%) при высокой нагрузке.

Пример 2. В качестве распределения  $F$  взято распределение Парето, причем параметры подобраны таким образом, что среднее значение всегда равно 3 (система перегружена), а дис-

персия меняется в шести экспериментах в диапазоне примерно от 0,04 до 5,5. Параметры распределения  $F$  представлены в таблице 5. Время выполнения заданий фиксировано и равно 1.

**Таблица 5.** Параметры распределения  $F$  входного потока в примере 2

№ эксперимента	1	2	3	4	5	6
параметры распределения	$3; \frac{2}{9}$	2, 5; 0, 2	$2, 25; \frac{5}{27}$	$2, 1; \frac{11}{63}$	$2, 05; \frac{7}{41}$	$2, 01; \frac{101}{603}$
значения дисперсии	0,0370	0,0889	0,1975	0,5291	1,0840	5,5279

Параметры новых алгоритмов в каждом эксперименте получались из решения *оптимизационной задачи 1*. При этом ограничение на среднюю длину очереди выбиралось равным значению этого показателя для стратегии  $\sigma_{RED}$ . Это обеспечивало в каждом эксперименте примерное равенство средней длины очереди для всех алгоритмов. Значения параметров новых стратегий приведены в таблицах 6 и 7.

**Таблица 6.** Параметры стратегий  $\bar{\sigma}_{RED}$  и  $\sigma_{DF}$  в примере 2

№ эксперимента	стратегия $\bar{\sigma}_{RED}$					стратегия $\sigma_{DF}$						
	$q_a$				$q_b$	длина очереди						
	0	1	2	3	1	2	3	4	5	6	7	
1	0,83	0,16	0,01		1	0,51	0,51	0,40	0,77	0,91	0,85	
2	0,54	0,35	0,09	0,02	1	0,44	0,39	0,27	0,01	0,89	1	
3	0,54	0,38	0,08		1	0,44	0,41	0,34	0,07	0,37	1	
4	0,63	0,32	0,05		1	0,34	0,25	0,13	0	0,65	1	
5		1			1	0,39	0,33	0,24	0,02	0,32	1	
6	0,76	0,24			1	0,38	0,32	0,23	0,02	0,28	1	

**Таблица 7.** Вероятности сброса для стратегии  $\sigma_{GDF}$  в примере 2

длина очереди	$\tau \leq 0,5$								$\tau > 0,5$							
	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	
1	0,20	0,76	0,71	0	0,77	0,33	1		0,58	0,89	0,49	0,05	1	0	1	
2	0,54	0,20	0,03	0,62	0,04	0	1		0,57	0,10	0,41	0	0	0,56	1	
3	0,10	0,90	0	0,29	0,11	0	0,72	1	0,78	0,24	0	0	0	0,02	1	
4	0,42	0,70	0,67	0,46	0,26	0	1		0,10	0,90	0,67	0,06	0	0,50	1	
5	0,90	0,51	0,50	0,49	0,13	0,33	1		0,33	0,28	0,41	0,58	0,39	0,07	1	
6	0,45	0,18	0,01	0	0,01	0	0,87	1	0,18	0,10	0,42	0,17	0	1		

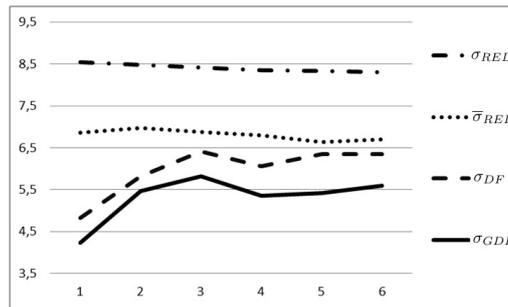
Результаты экспериментов в примере 2 собраны в таблице 9, в которой помимо вероятности потерь и средней длины очереди указаны значения среднеквадратичного отклонения (с.к.о.) длины очереди.

Наглядная иллюстрация результатов изображена на рисунках 2–5. На рисунке 2 показаны абсолютные значения вероятности потерь при четырех различных стратегиях. Поскольку алгоритм  $\sigma_{RED}$  сильно уступает остальным, то на следующих двух рисунках сравниваются между собой только новые стратегии. На рисунке 3 показаны отношения вероятностей потерь для стратегий  $\sigma_{DF}$  и  $\sigma_{GDF}$  по отношению к стратегии  $\bar{\sigma}_{RED}$ . На рисунке 4 показан относительный выигрыш, который доставляют более сложные стратегии. На рисунке 5 показаны абсолютные значения среднеквадратичного отклонения длины очереди.

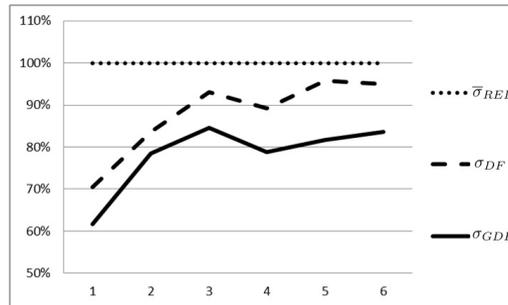
Результаты данной серии экспериментов показывают, что эффективность алгоритмов увеличивается по той же схеме, что и в примере 1:  $\sigma_{RED} \rightarrow \bar{\sigma}_{RED} \rightarrow \sigma_{DF} \rightarrow \sigma_{GDF}$ . Причем это касается как вероятности потерь, так и среднеквадратичного отклонения длины очереди (при

**Таблица 8.** Показатели эффективности при различных стратегиях в примере 2

стратегия	$\sigma_{RED}$	$\bar{\sigma}_{GDF}$	$\sigma_{DF}$	$\sigma_{GDF}$	$\sigma_{RED}$	$\bar{\sigma}_{GDF}$	$\sigma_{DF}$	$\sigma_{GDF}$	
№ эксперимента	1				2				
процент потерь, %	66,67	66,68	66,70	66,708	66,719	66,738	66,762	66,776	
длина очереди	среднее	8,545	6,852	4,833	4,228	8,481	6,967	5,820	5,465
	с.к.о.	1,001	1,179	0,941	0,802	1,137	1,268	0,759	0,723
№ эксперимента	3				4				
процент потерь, %	66,800	66,843	66,850	66,873	66,901	66,972	67,000	67,055	
длина очереди	среднее	8,415	6,879	6,585	5,878	8,352	6,791	6,059	5,349
	с.к.о.	1,287	1,389	0,910	0,856	1,427	1,504	1,074	1,000
№ эксперимента	5				6				
процент потерь, %	66,948	67,047	67,043	67,082	66,998	67,081	67,100	67,160	
длина очереди	среднее	8,326	6,630	6,350	5,791	8,301	6,695	6,353	5,600
	с.к.о.	1,1484	1,555	1,124	1,020	1,538	1,587	1,166	1,043



**Рис 2.** Вероятность потерь для различных стратегий при приблизительно одинаковой средней длине очереди в примере 2



**Рис 3.** Вероятность потерь для стратегий  $\sigma_{DF}$  и  $\sigma_{GDF}$  по отношению к стратегии  $\bar{\sigma}_{RED}$  в примере 2

одинаковой средней длине очереди). Выигрыш в вероятности потерь, который получается при новых стратегиях существенно больше при маленьких значениях дисперсии входного потока.

*Пример 3.* В качестве распределения  $F$  взято распределение Парето. Для шести экспериментов параметры выбраны таким образом, что среднее время между поступлениями заявок составляет 1, 25; 1, (1); 1; 2/3; 0, 5; 1/3 (таблица 9). Распределение  $G$  — равномерное на отрезке  $[0, 5; 1, 5]$ .

**Таблица 9.** Параметры распределения  $F$  входного потока в примере 3

№ эксперимента	1	2	3	4	5	6
параметры распределения	2,6007811; 0,7693752	2,4948471; 0,6657487	2,4142136; 0,5857864	2,2018504; 0,3638910	2,1180340; 0,2639320	2,0540926; 0,1710557
интенсивность входного потока	0,8	0,9	1	1,5	2	3

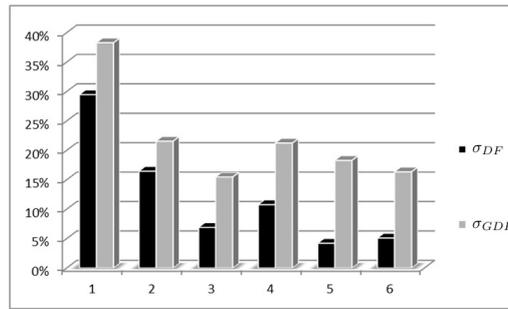


Рис 4. Относительное уменьшение вероятности потерь для стратегий  $\sigma_{DF}$  и  $\sigma_{GDF}$  по отношению к стратегии  $\bar{\sigma}_{RED}$  в примере 2

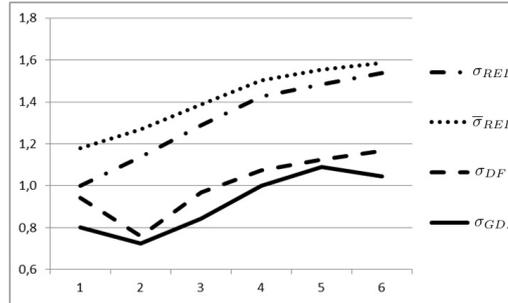


Рис 5. Среднеквадратичное отклонение длины очереди для различных стратегий в примере 2

К четырем испытываемым алгоритмам, задействованным в предыдущих примерах, добавлена пассивная стратегия  $\sigma_*$ . Она служит для сравнения эффективности, определяемой с помощью целевой функции

$$w = q + 0,1l,$$

где  $q$  — вероятность отказа,  $l$  — средняя длина очереди. Соответственно, параметры новых алгоритмов определялись из приближенного решения оптимизационной задачи 3. Значения целевой функции содержатся в таблице 10.

Таблица 10. Значения целевой функции при различных стратегиях в примере 3

стратегия	$\sigma_*$	$\sigma_{RED}$	$\bar{\sigma}_{RED}$	$\sigma_{DF}$	$\sigma_{GDF}$	$\sigma_*$	$\sigma_{RED}$	$\bar{\sigma}_{RED}$	$\sigma_{DF}$	$\sigma_{GDF}$		
№ эксперимента		1						2				
целевая функция	0,116	0,116	0,116	0,116	0,116	0,278	0,211	0,201	0,191	0,187		
№ эксперимента		3						4				
целевая функция	0,603	0,364	0,321	0,272	0,269	1,233	0,934	0,727	0,601	0,581		
№ эксперимента		5						6				
целевая функция	1,430	1,222	1,021	0,769	0,733	1,619	1,501	1,334	0,937	0,914		

Наглядное отображение результатов представлено на рисунках 6–8. Значения целевой функции представлены на рисунке 6 (абсолютные значения) и на рисунке 7 (относительные, по отношению к стратегии  $\sigma_*$ ). На рисунке 8 изображен относительный выигрыш в целевой функции для стратегий  $\sigma_{DF}$  и  $\sigma_{GDF}$  о отношению к стратегии  $\bar{\sigma}_{RED}$ .

Упорядоченность стратегий по критерию выбранной целевой функции совпадает с предыдущими примерами — новые алгоритмы однозначно имеют преимущество. Для всех стратегий выигрыш в целевой функции (по сравнению с пассивной стратегией) близок к нулю при малой нагрузке. При большой, больше 1, нагрузке стратегии  $\sigma_{DF}$  и  $\sigma_{GDF}$  доставляют выигрыш

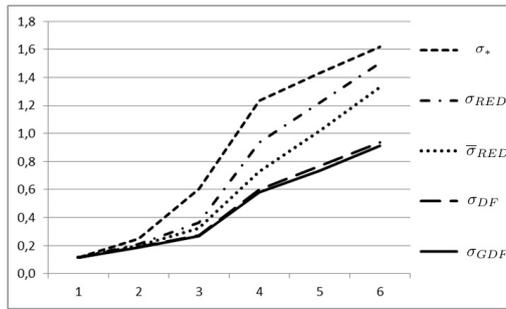


Рис 6. Значения целевой функции в примере 3

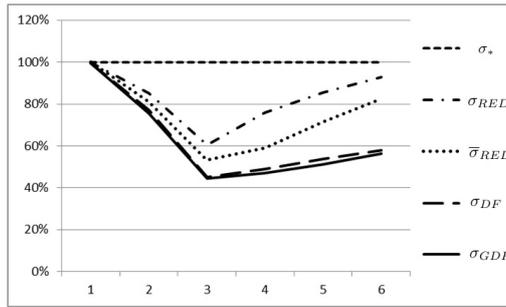


Рис 7. Относительные значения целевой функции по отношению к стратегии в примере 3

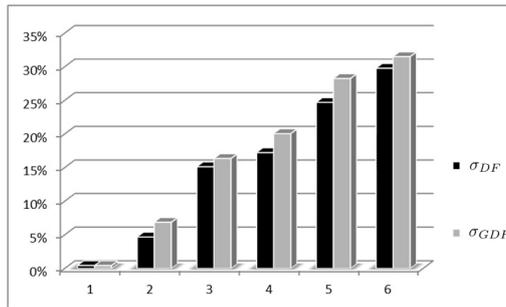


Рис 8. Относительный выигрыш в целевой функции для стратегий  $\sigma_{DF}$  и  $\sigma_{GDF}$  по отношению к стратегии  $\bar{\sigma}_{RED}$  в примере 3

примерно в 40–50% по сравнению со стратегией  $\sigma_*$ . Относительный выигрыш для стратегий  $\sigma_{DF}$  и  $\sigma_{GDF}$  по сравнению со стратегией  $\bar{\sigma}_{RED}$  монотонно увеличивается с ростом нагрузки и достигает примерно 30%. Преимущество стратегии  $\sigma_{GDF}$  по сравнению со стратегией  $\sigma_{DF}$  не так заметно, как в предыдущих примерах, хотя имеет систематический характер.

### 5. ЗАКЛЮЧЕНИЕ

Подводя итог описанию новых алгоритмов управления размером очереди и их экспериментальному анализу, можно сделать следующие выводы.

Эффективность алгоритмов, основанных на использовании функции сброса заданной структуры, может быть повышена за счет расширения множества допустимых стратегий. Простейший способ такого расширения заключается в рандомизации параметров алгоритмов. Этот прием продемонстрирован на примере подмножества алгоритмов ограничения доступа, известного под названием RED. Кусочно-линейная структура таких алгоритмов изначально парамет-

ризована двумя параметрами. Рандомизация этих параметров с последующей оптимизацией по отношению к трем различным целям оказалась эффективной. Показатели эффективности рандомизированного варианта алгоритма RED выше, чем у алгоритма с фиксированными детерминированными значениями параметров. Еще одно предложенное усовершенствование состоит в отказе от предваренной структуры функции сброса и поиска напрямую вероятностей сброса в зависимости от текущей длины очереди. Во всех исследованных примерах новая стратегия показала существенное преимущество по показателям качества по сравнению с алгоритмами RED.

Дальнейшее расширение множества допустимых стратегий заключается в учете дополнительной информации, основанной на наблюдении за временем выполнения заданий. Такой подход является существенным изменением по сравнению алгоритмами, основанными на функции сброса. Вероятность сброса выбирается зависящей не только от длины очереди, но и от остаточного времени выполнения задания на месте обслуживания. Во всех примерах показатели такой «сложной» стратегии выше, чем у всех остальных. Эффективность стратегий сравнивалась в численных экспериментах по трем критериям: по вероятности отказа (при примерно одинаковой средней длине очереди), по средней длине очереди (при примерно одинаковой вероятности отказа) и по значениям целевой функции — линейной свертке обоих показателей. Во всех без исключения экспериментах эффективность стратегий возрастала по схеме «алгоритм RED ( $\sigma_{RED}$ ) → рандомизированный алгоритм RED ( $\bar{\sigma}_{GDF}$ ) → алгоритм с функцией сброса общего вида ( $\sigma_{DF}$ ) → алгоритм, учитывающий время выполнения заданий ( $\sigma_{GDF}$ )». Особенно заметно проявляется преимущество новых алгоритмов для систем с большой входной нагрузкой. Так, стратегии  $\sigma_{DF}$  и  $\sigma_{GDF}$  обнаруживают такое же преимущество над алгоритмом  $\sigma_{RED}$ , как алгоритм  $\sigma_{RED}$  по отношению к неуправляемой системе. Указанная сравнительная эффективность стратегий сохраняется и по отношению к критерию среднеквадратичного отклонения длины очереди. Подчеркнем, что перечисленные сравнительные результаты получены, в том числе, для неэкспоненциальных распределений.

Следует отметить, что новые алгоритмы требуют для реализации несколько большего объема памяти, чем алгоритмы типа RED, поскольку параметризуются большим количеством параметров. Например, в рассмотренной модификации алгоритма RED необходимо помнить значения двух вещественных параметров, в то время как в наиболее сложной из новых стратегий таких значений (в худшем случае) —  $2N$ , где  $N$  — размер бункера. Однако для современных технических возможностей такое различие вряд ли может стать принципиальным в случае возможного внедрения. Вычислительная сложность новых стратегий примерно совпадает с традиционными алгоритмами. Так, для одношаговой реализации алгоритма RED, то есть для определения, надо ли отклонять новую заявку, необходимо выполнить арифметические операции по заданной формуле. В новых алгоритмах аналогичная процедура требует реализации случайной величины с заданным дискретным распределением.

Рассмотренные в этой работе алгоритмы использовались, как статические, с заданными заранее значениями параметров. Это касается, в частности, алгоритма RED, параметры которого взяты из работы [6]. Оптимизационные задачи нахождения параметров для стратегий ограничения доступа математически сложны и пока не поддаются решению, тем более, с произвольными вероятностными характеристиками входного потока и времени обслуживания. Техника, использованная (без обсуждения) в настоящей работе для определения параметров новых алгоритмов, потенциально пригодна также для оптимизации параметров классических алгоритмов. Более того, она же может служить основой для перехода от стратегий со статическими параметрами к алгоритмам, в которых настройка параметров осуществляется непосредственно в процессе функционирования системы. Последняя тема является перспективной для дальнейших исследований.

## СПИСОК ЛИТЕРАТУРЫ

1. Chydzinski A., Mrozowski P. "Queues with dropping functions and general arrival processes" // *PLoS ONE*, 2016, vol. 11, no. 3, paper e0150702.
2. Adams R. "Active queue management: A survey". *IEEE Communications Surveys Tutorials*, 2013, vol. 15, no. 3, pp. 1425–1476.
3. Bonald T., May M., Bolot J. C. "Analytic evaluation of red performance". *Proceedings IEEE INFOCOM 2000 Conference on Computer Communications*, 2000, vol. 3, pp. 1415–1424.
4. Zhernovyi Yu., Kopytko B., Zhernovyi K. "On characteristics of the  $M^{\theta}/G/1/m$  and  $M^{\theta}/G/1$  queues with queue-size based packet dropping". *Journal of Applied Mathematics and Computational Mechanics*, 2014, vol. 13, no. 4, pp. 163–175.
5. Baker F., Fairhurst G. "IETF recommendations regarding active queue management", 2014, <https://tools.ietf.org/html/draft-ietf-aqm-recommendation-07>.
6. Chydzinski A., Chrost L. "Analysis of AQM queues with queue size based packet dropping" // *Int. J. Appl. Math. Comput. Sci.*, 2011, vol. 21, no. 3, pp. 567–577.
7. Tikhonenko O., Kempa W. "Queue-Size Distribution in  $M/G/1$ -type system with bounded Capacity and packet Dropping". *Dudin A., Klimenok V., Tsarenkov G., Dudin S. (eds) Modern Probabilistic Methods for Analysis of Telecommunication Networks. Communications in Computer and Information Science*, 2013, vol 356. Springer, Berlin, Heidelberg, pp. 177–186.
8. Tikhonenko O., Kempa W. "Performance evaluation of an  $M/G/n$ -type queue with bounded capacity and packet dropping". *International Journal of Applied Mathematics and Computer Science*, 2016, vol. 26, no. 4, pp. 841–854.
9. Hao W., Wei Y. "An extended  $GI^X/M/1/N$  queueing model for evaluating the performance of AQM algorithms with aggregate traffic". *X. Lu and W. Zhao (Eds.), Networking and Mobile Computing, Lecture Notes in Computer Science*, 2005, vol. 3619, Springer-Verlag, Berlin, Heidelberg, pp. 395–404.
10. Kempa W. "A direct approach to transient queue-size distribution in a finite-buffer queue with AQM". *Applications of Mathematics & Information Sciences*, 2013, vol. 7, no. 1, pp. 909–915.
11. Tikhonenko O., Kempa W. "Queueing system with processor sharing and limited memory under control of the AQM mechanism". *Automation and Remote Control*, 2015, vol. 76, no. 10, pp. 1784–1796.
12. Коновалов М. Г. Методы адаптивной обработки информации и их приложения. М.: ИПИ РАН, 2007. 212 с.

## Numerical analysis of improved queue length control algorithms for $GI/G/1/N$ systems

Konovall M.G., Razumchik R.V.

The paper is devoted to the description and numerical analysis of new active queue management mechanisms. In the core of the mechanisms lies the idea of the dropping function, which is the bases of the classic AQM mechanisms like RED. Three modifications of the dropping function are proposed: randomization of its control parameters, alteration of the function structure and addition of new arguments (remaining service time). It is assumed that the system may be modelled by a  $G/G/1/N$  queue, and the inter-arrival time distribution may be heavy-tailed. All results of our numerical experiments, including those presented in the paper, show that the efficiency of the mechanisms is increased with the introduction of each new modification. The gain is especially pronounced in case of high system load.

**KEYWORDS:** active queue management; access restriction algorithms; dropping function.