

УДК 004.021, 519.2

## К вопросу о классификации алгоритмов RED

А. В. Королькова, Д. С. Кулябов, А. И. Черноиванов

*Кафедра систем телекоммуникаций  
Российский университет дружбы народов  
ул. Миклухо-Маклая, 6, Москва, Россия, 117198*

В статье предложена классификация алгоритмов на базе алгоритма RED (Random Early Detection). Показано, что практически все варианты модификаций RED ортогональны и их можно комбинировать.

**Ключевые слова:** алгоритм случайного раннего обнаружения (RED), взвешенный RED, расширенный RED (ERED), самонастраивающийся RED (ARED), улучшенный ARED (RARED), POWRED, гибкий RED (GRED), RED с двойным наклоном (DSRED), RED с зависимостью от состояния (SDRED), устойчивый RED (SRED), RED с профильными и непрофильными пакетами (RIO), SF-RED.

### 1. Введение

Алгоритм Random Early Detection (RED) [1] лежит в основе ряда механизмов предотвращения и контроля перегрузок в очередях маршрутизаторов. Его основное предназначение заключается в сглаживании временных всплесков трафика и предупреждении длительной перегрузки сети посредством уведомления источников трафика о необходимости снижения интенсивности передачи информации.

Существует множество модификаций RED, в той или иной степени улучшающих характеристики алгоритма. В работе [2] была предложена их предварительная классификация и было показано, что фактически все варианты модификаций RED ортогональны и их можно комбинировать для достижения определённых оптимальных характеристик. В данной работе более подробно описывается принцип предлагаемой классификации и объясняется отнесение той или иной модификации алгоритма RED к определённому классу.

### 2. Алгоритм RED (Random Early Detection)

Принцип функционирования и сам алгоритм RED подробно рассмотрены в работе [1]. Здесь приведём лишь формулы для расчёта вероятности сброса и экспоненциально взвешенной скользящей средней длины очереди  $\hat{q}$ .

Использование  $\hat{q}$  вызвано необходимостью сглаживания (low-pass filter) выбросов мгновенного значения длины очереди  $q$ . Для вычисления  $\hat{q}$  используется формула экспоненциально взвешенного скользящего среднего (Exponentially Weighted Moving-Average, EWMA):

$$\hat{q}_{k+1} = (1 - w)\hat{q}_k + wq_k, \quad (1)$$

где  $w$ ,  $0 < w < 1$  — весовой коэффициент, определяемый пользователем. В работе [3] предложена следующая формула для определения  $w$ :

$$w = 1 - e^{-1/C},$$

где  $C$  — пропускная способность канала (пакетов в секунду).

Вероятность  $p_b$  маркировки на отбрасывание пакетов представляет собой функцию, линейно зависящую от  $\hat{q}$ , минимального  $q_{\min}$  и максимального  $q_{\max}$

пороговых значений и параметра  $p_{\max}$ , определяющего часть отбрасываемых пакетов при достижении средним размером очереди значения  $q_{\max}$  и вычисляется следующим образом:

$$p_b = \begin{cases} 0, & 0 < \hat{q} \leq q_{\min}, \\ \frac{\hat{q} - q_{\min}}{q_{\max} - q_{\min}} p_{\max}, & q_{\min} < \hat{q} \leq q_{\max}, \\ 1, & \hat{q} > q_{\max}. \end{cases}$$

Окончательная же вероятность  $p_a$  маркировки на отбрасывание пакета может иметь геометрическое распределение с параметром  $p_b$ :

$$p_a = (1 - p_b)^{n-1} p_b$$

или равномерное распределение:

$$p_a = \begin{cases} \frac{p_b}{1 - (n-1)p_b} \prod_{i=0}^{n-2} \left(1 - \frac{p_b}{1 - ip_b}\right) = p_b, & 1 \leq n \leq \frac{1}{p_b}, \\ 0, & n > \frac{1}{p_b}, \end{cases}$$

Для второго случая вероятность быть отмеченным на сброс для каждого входящего пакета равна  $p_b / (1 - \text{count} \cdot p_b)$ , где  $\text{count}$  — количество неотмеченных пакетов, прибывших после последнего отмеченного.

### 3. Модель RED

В некотором устройстве функционирование модуля, реализующего алгоритм RED, можно представить следующим образом (рис. 1). Пакет при поступлении в систему попадает в модуль сброса. Решение о сбросе пакета принимается на основе значения вероятности  $p$ , получаемого от управляющего модуля. Вероятность  $p$  сброса пакетов зависит от экспоненциально взвешенного скользящего среднего размера длины очереди  $\hat{q}$ , также вычисляемого управляющим модулем, основываясь на текущем значении длины очереди  $q$ .

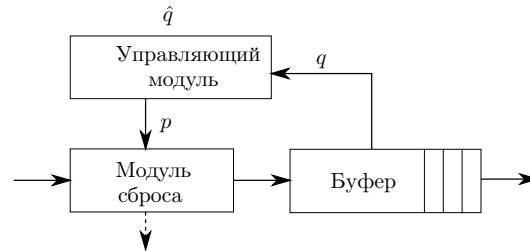


Рис. 1. Модуль RED

Под сбросом понимается как собственно отбрасывание пакета (с вероятностью  $p_d$ ), так и маркировка пакета (с вероятностью  $p_m$ ) для последующей отсылки сообщения ECN (Explicit Congestion Notification) [4]. Таким образом, вероятность сброса представима в виде  $p = p_d + p_m$ .

Поведение модуля RED можно описать двумя основными функциями: функцией экспоненциально взвешенной скользящей средней длины очереди  $\hat{q} = Q(p)$  и функцией сброса  $p = P(\hat{q})$ , задающими систему:

$$\begin{cases} \hat{q} = Q(p), \\ p = P(\hat{q}). \end{cases} \quad (2)$$

Решение  $(\hat{q}_s, p_s)$  системы (2) задаёт равновесное состояние этой системы (рис. 2), т.е. в установившемся состоянии экспоненциально взвешенная скользящая средняя длина очереди будет  $\hat{q}_s$ , а вероятность сброса —  $p_s$ <sup>1</sup>.

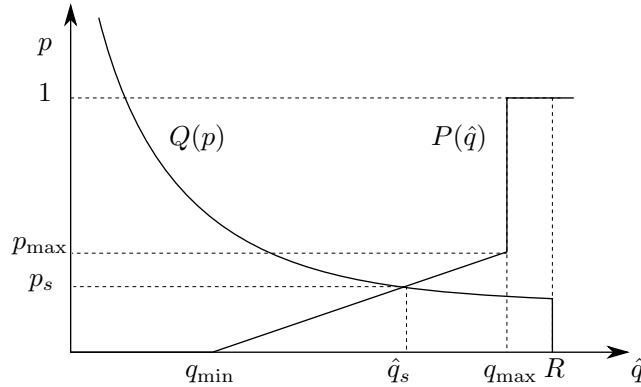


Рис. 2. Точка равновесия для системы (2)

Параметры  $q_{\min}$  и  $q_{\max}$  задают рабочую область системы (2). Алгоритм эффективно действует только в её границах. При этом выход за верхнюю границу менее желателен, чем за нижнюю.

Функция сброса  $p = P(\hat{q})$  должна удовлетворять следующим условиям:

- в районе нижней границы рабочего диапазона её наклон (первая производная) должен быть как можно меньше (для повышения эффективности использования полосы пропускания);
- в районе верхней границы рабочего диапазона её наклон должен быть максимальным (для предотвращения выхода из рабочего диапазона).

Таким образом желательно, чтобы функция сброса была нелинейной. Но основным требованием при разработке алгоритма RED [1, 5] стало обеспечение малых накладных расходов при обработке трафика,<sup>2</sup> что проявилось в простоте всех вычислений в этом алгоритме. Он построен так, чтобы минимизировать сложные расчёты (умножение, вычисления с плавающей точкой). Поэтому при модификации базового алгоритма вместо нелинейной функции сброса можно использовать её аппроксимацию кусочно-линейной функцией.

Функция экспоненциально взвешенной скользящей средней длины очереди  $\hat{q} = Q(p)$  зависит от многих параметров (характеристик потока, дисциплины обслуживания и т.д.) и требует построения математической модели для своего описания. Можно модифицировать формулу (1), считая, что вся зависимость от внешних параметров учитывается мгновенной длиной очереди  $q$ . При этом  $\hat{q}$  должна строиться так, чтобы противодействовать выходу системы из рабочей области.

## 4. Построение классификации

При построении классификации используется фасетный метод, подразумевающий параллельное разделение множества объектов на независимые классификационные группы. Классы обозначаются прописными латинскими буквами, подклассы — в скобках после символа класса.

Будем классифицировать системы RED по следующим характеристикам:

- количество модулей RED — класс M:

<sup>1</sup>Система будет осциллировать около устойчивого состояния.

<sup>2</sup>Именно малые накладные расходы, а соответственно и малые задержки, присущие технологии RED, позволяют рекомендовать её для применения в мультисервисных сетях.

- алгоритмы с единым для всех модулей RED буфером — подкласс CS (Complete Sharing);
- алгоритмы со своим (виртуальным) буфером у каждого модуля — подкласс CP (Complete Partitioning);
- вид функции очереди  $Q$  — класс Q;
- вид функции сброса  $P$  — класс P.
  - нелинейная в рабочей области функция — подкласс NL (Non-Linear);
  - нелинейная с кусочно-линейной аппроксимацией в рабочей области функция — подкласс NLA (Non-Linear with Approximation);
  - изменённая вне рабочей области функция, возвращающая систему в устойчивое состояние — подкласс RS (Return to Steady);
  - с дополнительной аддитивной или мультипликативной вероятностью — подклассы PA (Additive Probability) и PM (Multiplicative Probability) соответственно.

Предлагаемую структуру классификации можно пояснить следующим образом.

Модуль RED может обрабатывать лишь один поток трафика. Разновидности с несколькими блоками RED применимы в случае, когда имеется несколько потоков, которые должны быть обработаны с разными параметрами<sup>3</sup>. При этом параметры RED (пороговые значения и максимальная вероятность сброса) могут быть едиными для всех типов трафика или индивидуальными для каждого потока. Как уже было сказано, поведение модуля RED описывается функциями  $P$  и  $Q$ .

Изменения экспоненциально взвешенной скользящей средней длины очереди (1) направлены на то, чтобы увеличить крутизну функции в области значений  $\hat{q}$ , меньших нижнего порога, и уменьшить крутизну в областях значений, больших верхнего порога. Это обусловлено тем, что при выходе за пределы рабочей области система должна скорее вернуться в рабочий диапазон.

Наибольший интерес представляют алгоритмы, изменяющие вид функции сброса. Предположение о предпочтительности нелинейного вида функции сброса уже обсуждалось выше. Также было сказано о необходимости уменьшения вычислительной сложности алгоритма, что можно достигнуть аппроксимацией нелинейной функции сброса кусочно-линейной функцией.

Изменение вида функции сброса за пределами рабочей области служит, как и в случае с функцией очереди, возвращению системы в устойчивое состояние.

При необходимости задать дополнительную зависимость функции сброса от других параметров (например, явно учитывать количество входящих потоков) можно задать дополнительную вероятность, которая может быть аддитивной:  $p = P_{\text{RED}}(\hat{q}) + P_{\text{add}}$ , или мультипликативной:  $p = P_{\text{RED}}(\hat{q}) \times P_{\text{add}}$ .

Для иллюстрации построенной классификации приведём описание наиболее характерных алгоритмов для каждого класса RED.

## 5. Обзор алгоритмов на базе RED

### 5.1. Алгоритм Enhanced Random Early Detection

В алгоритме Enhanced Random Early Detection (ERED), предложенном в работе [6], изменениям подвергается функция экспоненциально взвешенной скользящей средней длины очереди:

$$\hat{q}_{k+1} = \begin{cases} \frac{1-w}{\alpha} \hat{q}_k + wq_k, & q < q_{\min}, \quad \alpha > 1, \\ (1-w) \hat{q}_k + wq_k, & q_{\min} \leq q < q_{\max}, \\ (1-w) \hat{q}_k + \frac{wq_k}{\beta}, & q \geq q_{\max}, \quad \beta > 1, \end{cases} \quad (3)$$

<sup>3</sup>Обычно такая ситуация возникает в системах DiffServ.

где  $\alpha$  и  $\beta$  — некоторые коэффициенты, позволяющие управлять задержкой реагирования  $\hat{q}$  на изменение текущей длины очереди  $q$ , что уменьшает амплитуду и частоту колебаний экспоненциально взвешенной скользящей средней длины очереди.

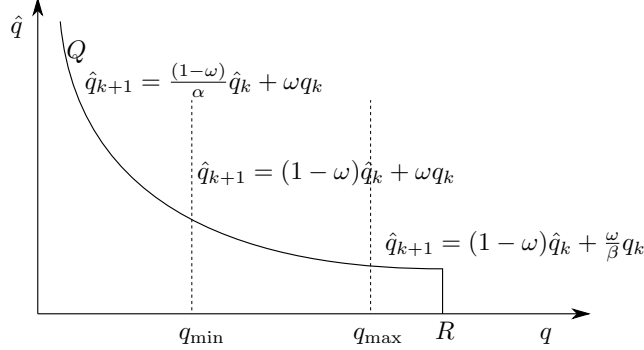


Рис. 3. Функция экспоненциально взвешенной скользящей средней длины очереди в алгоритме ERED

В этом алгоритме поведение функции очереди модифицируется таким образом, чтобы при выходе за пределы рабочей области значение  $\hat{q}$  как можно скорее возвратилось обратно. Отнесём его к классу Q.

## 5.2. Адаптивные алгоритмы: Adaptive RED, Refined ARED, POWARED

Подбор оптимальных параметров RED представляет определённую сложность. Возможный вариант — использовать адаптивные механизмы, позволяющие динамически изменять параметры в зависимости от характера поведения трафика.

В алгоритме Adaptive RED (ARED), разработанном Фенгом [7, 8], функция сброса модифицируется посредством изменения по принципу MIMD<sup>4</sup> (Multiplicative Increase Multiplicative Decrease) параметра  $p_{\max}$  таким образом, чтобы удерживать экспоненциально взвешенную скользящую среднюю длину очереди  $\hat{q}$  между заданными минимальным  $q_{\min}$  и максимальным  $q_{\max}$  порогами (т.е.  $p_{\max} = p_{\max}(\hat{q})$ ).

Данный алгоритм был усовершенствован Флойд [9] и теперь под алгоритмом ARED подразумевают именно эту модификацию. В этой модификации используется принцип AIMD<sup>5</sup> (Additive Increase Multiplicative Decrease).

Алгоритм ARED функционирует следующим образом. Для каждого интервала **interval** (параметр) в секундах, если  $\hat{q}$  больше целевой (желаемой)  $\hat{q}_t$  и  $p_{\max} \leq 0,5$ , то  $p_{\max}$  увеличивается на некоторую величину  $\alpha$ ; в противном случае, если  $\hat{q}$  меньше целевой  $\hat{q}_t$  и  $p_{\max} \geq 0,01$ , то  $p_{\max}$  уменьшается в  $\beta$  раз (выбор значений параметров объяснён в [9]):

$$p_{\max} = \begin{cases} p_{\max} + \alpha, & \hat{q} > \hat{q}_t, \quad p_{\max} \leq 0,5, \\ p_{\max} \cdot \beta, & \hat{q} < \hat{q}_t, \quad p_{\max} \geq 0,01, \end{cases} \quad (4)$$

где

$$\alpha = \min\left(0,01; \frac{p_{\max}}{4}\right), \quad \beta = 0,9,$$

$$q_{\min} + 0,4(q_{\max} - q_{\min}) < \hat{q}_t < q_{\min} + 0,6(q_{\max} - q_{\min}).$$

<sup>4</sup> Данный принцип заключается в том, что увеличение и уменьшение некоторой величины производится путём умножения на некоторые параметры.

<sup>5</sup> Принцип AIMD заключается в том, что увеличение некоторой величины производится путём сложения с некоторым параметром, а уменьшение — путём умножения на параметр.

Таким образом, посредством изменения  $p_{\max}$  строится нелинейная функция сброса. Алгоритм относится к классу P(NL).

Алгоритм Refined ARED (RARED) [10] является модификацией ARED и предлагает более активно изменять вероятность сброса  $p_{\max}$ , чтобы иметь возможность быстрой адаптации к изменяющейся экспоненциально взвешенной скользящей средней длине очереди  $\hat{q}$ :

$$p_{\max} = \begin{cases} p_{\max} + \alpha, & \hat{q} > \hat{q}_t, \quad p_{\max} \leq 0,5, \\ \beta p_{\max}, & \hat{q} \leq \hat{q}_t, \quad p_{\max} \geq 0,01, \end{cases}$$

где

$$q_{\min} + 0,48(q_{\max} - q_{\min}) < \hat{q}_t < q_{\min} + 0,52(q_{\max} - q_{\min}),$$

$$\alpha = \left(0,25 \frac{\hat{q} - \hat{q}_t}{\hat{q}_t}\right) p_{\max}, \quad \beta = 1 - \left(0,17 \frac{\hat{q}_t - \hat{q}}{\hat{q}_t - q_{\min}}\right).$$

Основной особенностью данного алгоритма является подстраивание коэффициентов  $\alpha$  и  $\beta$  в зависимости от  $\hat{q}$  для того, чтобы быстрее возвращать систему в стационарный режим.

Алгоритм POWARED [11] также является модификацией алгоритма ARED. Его суть заключается в том, что интервал  $Q = [q_{\min}; q_{\max}]$  ( $\hat{q} \in Q$ ) разбивается на 3 части:  $Q_1 = [q_{\min}; \hat{q}_t]$ ,  $Q_2 = (\hat{q}_t; q_{\max}]$  и  $Q_3 = \hat{q}_t$ , а  $p_{\max}$  вычисляется по формуле:

$$p_{\max} = \begin{cases} p_{\max} - \delta_1, & \hat{q} \in Q_1, \\ p_{\max} + \delta_2, & \hat{q} \in Q_2, \\ p_{\max}, & \hat{q} \in Q_3, \end{cases}$$

где

$$\delta_1 = \left| \left( \frac{\hat{q} - \hat{q}_t}{\beta \hat{q}_t} \right)^K \right|, \quad \delta_2 = \left| \left( \frac{\hat{q}_t - \hat{q}}{\beta(R - \hat{q}_t)} \right)^K \right|, \quad \hat{q}_t = \frac{1}{2}(q_{\max} + q_{\min}),$$

$K$  ( $K > 1$ ,  $K = 2, 3, \dots$ ) — множитель,  $\beta$  — параметр сжатия. Легко видеть, что здесь используется принцип AIAD<sup>6</sup> (Additive Increase Additive Decrease).

Данный алгоритм является улучшением алгоритма ARED (в основном за счёт введения дополнительных параметров). Отнесём его к классу P(NL).

**Пример 1.** Пусть заданы следующие параметры: размер буфера  $R = 100$ ,  $q_{\min} = 20$ ,  $q_{\max} = 80$ ,  $p_{\max} = 0,01$ ,  $\hat{q}_t = \frac{1}{2}(q_{\max} + q_{\min})$ . Для ARED  $\beta = 0,9$ , для POWARED рассматриваются 2 случая:  $\beta = 2$  и  $\beta = 5$ . Остальные параметры вычисляются согласно формулам соответствующих алгоритмов.

На рис. 4–7 показано функционирование рассматриваемых алгоритмов в стационарном режиме.

Из рис. 4–5 видно, что в начале работы, когда очередь пуста, система стремится к стационарной точке  $\hat{q}_t$ . Затем при увеличении  $\hat{q}$  вероятность сброса пакетов повышается и при  $\hat{q} = \hat{q}_{\max}$  достигает своего максимального для данного состояния значения. Затем начинается вторая часть цикла, когда  $\hat{q} \rightarrow \hat{q}_{\min}$ . Значение  $p_{\max}$  на данном этапе уже отлично от первоначального. Поскольку соблюдается условие  $\hat{q} > \hat{q}_t$ ,  $p_{\max} \leq 0,5$  алгоритма (как ARED, так и RARED), то значение  $p_{\max}$  увеличивается на коэффициент  $\alpha$ . Далее, по мере того, как  $\hat{q} \rightarrow \hat{q}_t$ , вероятность сброса увеличивается. Этот процесс будет происходить до тех пор, пока система не достигнет своей стационарной точки  $\hat{q} = \hat{q}_t$ . После неё начнётся нелинейное уменьшение вероятности сброса пакетов, которое достигнет своего минимума при  $\hat{q} = \hat{q}_{\min}$ , и на этом первый цикл заканчивается.

Второй цикл снова начинается из точки  $\hat{q} = \hat{q}_{\min}$ , однако  $p_{\max}$  уже отличается от своего начального значения  $p_{\max} = 0,01$ . Из графиков видно, что система, стремясь к стационарной точке  $\hat{q}_t$ , снова понижает вероятность сброса пакетов

<sup>6</sup>Принцип AIAD заключается в том, что увеличение / уменьшение некоторой величины производится путём сложения / вычитания с некоторым параметром.

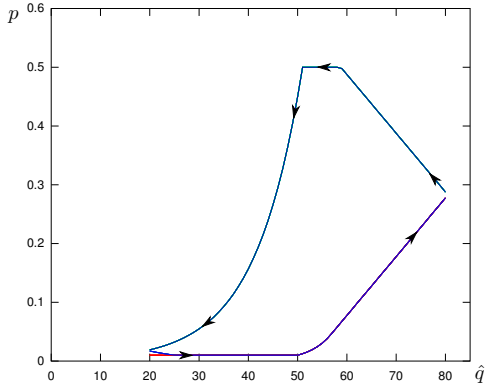


Рис. 4. Функционирование ARED в стационарном режиме

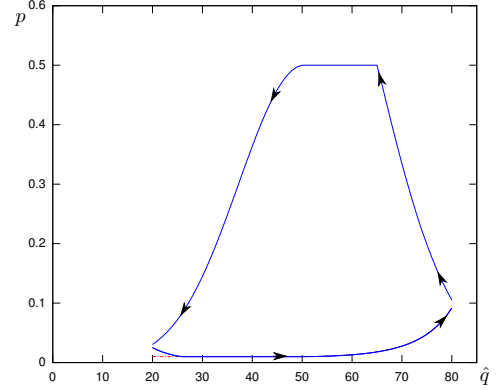
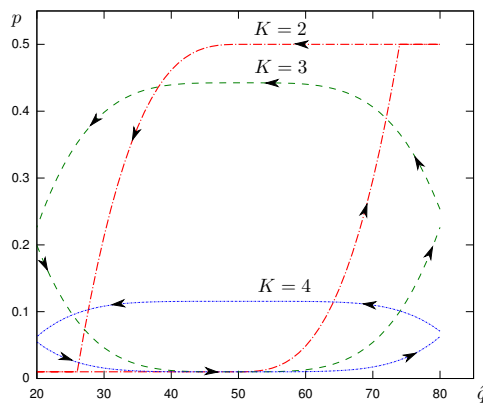
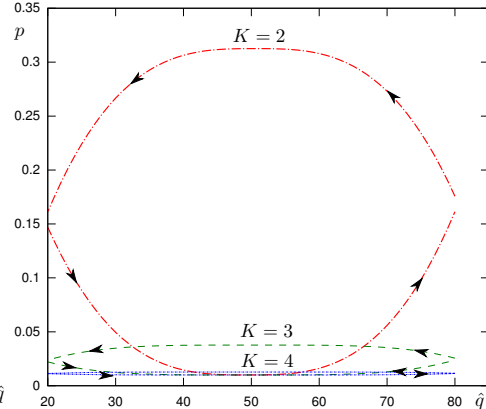


Рис. 5. Функционирование RARED в стационарном режиме

и, пройдя её, начинает повышать  $p_{\max}$  и, соответственно, вероятность сброса пакетов, которая достигнет своего максимума для данного состояния в точке  $\hat{q}_{\max}$ . Затем система пойдет по второй части цикла, повышая вероятность сброса на пути к стационарной точке, после которой начнется её нелинейное понижение. Таким образом закончится второй цикл, а система войдёт в своё стационарное состояние, в рамках которого она и будет продолжать функционировать.

Алгоритм POWARED более агрессивно реагирует на изменение  $\hat{q}$ , чем ARED (рис. 6, 7). Особенно хорошо это видно в случае  $K = 2$ .

Рис. 6. Функционирование POWARED в стационарном режиме,  $K = 2, 3, 4, \beta = 2$ Рис. 7. Функционирование POWARED в стационарном режиме,  $K = 2, 3, 4, \beta = 5$ 

В то же время при  $K = 4$  вероятность сброса меняется очень медленно, что может снизить эффективность алгоритма. Значение  $K = 3$  представляется наиболее оптимальным по эффективности — вероятность сброса меняется, но не резко.

Параметр  $\beta$  представляет собой коэффициент сжатия — чем он больше, тем меньше разброс значений вероятности сброса.

Также видно, что после прохождения стационарной точки  $\hat{q} = \hat{q}_t$  вероятность сброса пакетов остается на своём прежнем уровне довольно долгое время. Исходя из этого получается, что система старается все время держать очередь рядом с желаемым значением  $\hat{q}_t$ .

### 5.3. DSRED

В алгоритме Double Slope Random Early Detection (DSRED) [12] вводится дополнительное пороговое значение  $q_{\text{mid}}$  между минимальным  $q_{\text{min}}$  и максимальным  $q_{\text{max}}$  порогами, причём  $q_{\text{mid}} = 0,5(q_{\text{max}} - q_{\text{min}})$ . Функция сброса описывается двумя линейными сегментами с углами наклона  $\alpha$  и  $\beta$  соответственно, регулируемые задаваемым селектором режимов  $\gamma$  (см. рис. 8).

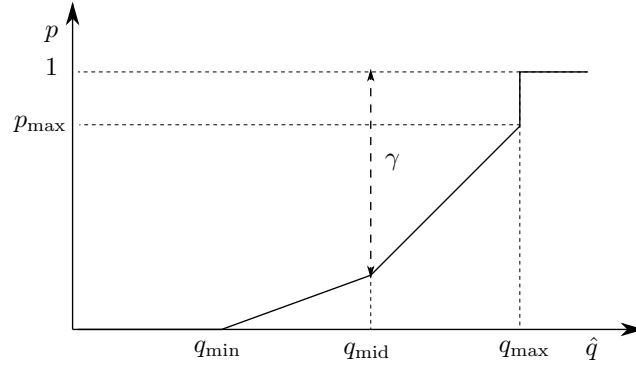


Рис. 8. Вид функции сброса в алгоритме DSRED

Функция вероятности сброса пакетов в алгоритме DSRED имеет вид:

$$p(\hat{q}) = \begin{cases} 0, & \hat{q} < q_{\text{min}}, \\ \alpha(\hat{q} - q_{\text{min}}), & q_{\text{min}} \leq \hat{q} < q_{\text{mid}}, \\ 1 - \gamma + \beta(\hat{q} - q_{\text{mid}}), & q_{\text{mid}} \leq \hat{q} < q_{\text{max}}, \\ 1, & \hat{q} \geq q_{\text{max}}, \end{cases} \quad (5)$$

где

$$\alpha = \frac{2(1 - \gamma)}{q_{\text{max}} - q_{\text{min}}}, \quad \beta = \frac{2\gamma}{q_{\text{max}} - q_{\text{min}}},$$

а  $\hat{q}$  вычисляется так же, как и в RED по формуле (1).

Таким образом, идеей данного алгоритма является задание нелинейной функции сброса, аппроксимированной ломаной линией с двумя звеньями. Алгоритм отнесём к классу P(NLA).

### 5.4. GRED

Функция вероятности сброса пакетов в алгоритме Gentle RED (GRED) [13] имеет вид:

$$p(\hat{q}) = \begin{cases} 0, & \hat{q} < q_{\text{min}}, \\ \frac{\hat{q} - q_{\text{min}}}{q_{\text{max}} - q_{\text{min}}} p_{\text{max}}, & q_{\text{min}} \leq \hat{q} < q_{\text{max}}, \\ \frac{\hat{q} - q_{\text{max}}}{q_{\text{max}}} (1 - p_{\text{max}}) + p_{\text{max}}, & q_{\text{max}} \leq \hat{q} < 2q_{\text{max}}, \\ 1, & \hat{q} \geq 2q_{\text{max}}, \end{cases} \quad (6)$$

При формальной схожести поведения функции сброса алгоритмов DSRED и GRED между ними существует принципиальное различие. В DSRED функция имеет вид ломаной линии в пределах рабочего диапазона, фактически аппроксимируя нелинейную функцию. В GRED в рабочем диапазоне находится стандартная функция сброса RED. Дополнительный же сегмент служит для обеспечения более плавного поведения функции сброса за пределами рабочего диапазона. Отнесём этот алгоритм к классу P(RS).



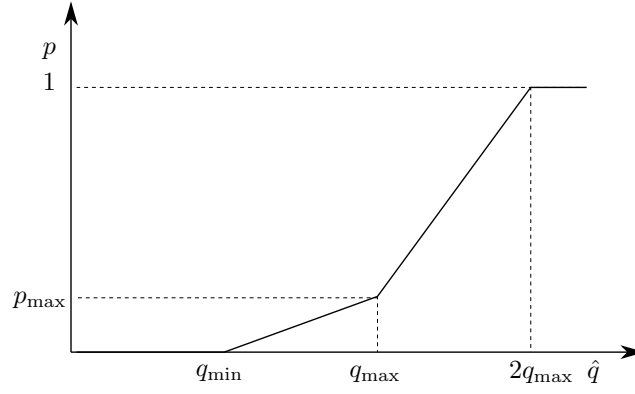


Рис. 9. Вид функции сброса в алгоритме GRED

### 5.5. SDRED

Из четырех параметров RED в SDRED изменяются максимальный порог  $q_{\max}$  и вес  $w$  очереди.

Функция вероятности сброса пакетов в алгоритме SDRED (рис. 10) вычисляется следующим образом [14]:

$$p(\hat{q}) = \begin{cases} 0, & \hat{q} < q_{\min} \\ \frac{\hat{q} - q_{\min}}{q_{\max} - q_{\min}} p_{\max}, & q_{\min} \leq \hat{q} < q_{\max}, \\ \frac{\hat{q} - q_{\min}}{(q_{\max} + 0,1R) - q_{\min}} p_{\max}, & q_{\max} \leq \hat{q} < 0,7R, \\ \frac{\hat{q} - q_{\min}}{(q_{\max} + 0,2R) - q_{\min}} p_{\max}, & 0,7R \leq \hat{q} < 0,8R, \\ \frac{\hat{q} - q_{\min}}{(q_{\max} + 0,3R) - q_{\min}} p_{\max}, & 0,8R \leq \hat{q} < 0,9R, \\ 1, & 0,9R \leq \hat{q} \leq R. \end{cases} \quad (7)$$

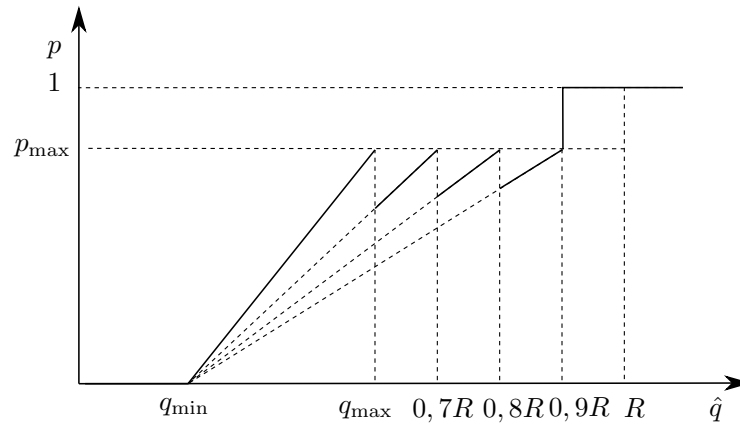


Рис. 10. Вид функции сброса в алгоритме SDRED

Значение экспоненциально взвешенной скользящей средней длины очереди в алгоритме SDRED имеет вид:

$$\hat{q}_{i+1} = \begin{cases} (1-w)\hat{q}_i + wq_i, & \hat{q}_i < q_{\max}, \quad 0,9R \leq \hat{q}_i \leq R, \\ (1-kw)\hat{q}_i + kwq_i, & q_{\max} \leq \hat{q}_i < 0,7R, \\ (1-k^2w)\hat{q}_i + k^2wq_i, & 0,7R \leq \hat{q}_i < 0,8R, \\ (1-k^3w)\hat{q}_i + k^3wq_i, & 0,8R \leq \hat{q}_i < 0,9R. \end{cases} \quad (8)$$

где  $k$  — константа для изменения  $w$  в зависимости от коэффициента занятости очереди.

В данном алгоритме применяются сразу два подхода: изменение функции очереди и функции сброса. Оба эти подхода направлены на ускорение возврата  $\hat{q}$  в рабочую область. Классифицируем данный алгоритм как QP(RS).

## 5.6. SRED (Stabilized RED)

Алгоритм SRED [15] использует дополнительную вероятность сброса, зависящую от числа активных соединений или потоков. Фактически с помощью данного алгоритма выявляются потоки, ведущие себя неадекватно (например, требующие неоправданно много ширины полосы пропускания). Эта схема позволяет держать занятость буфера на заранее определённых уровнях в независимости от количества активных соединений.

Результирующая вероятность представляет собой произведение полученной дополнительной вероятности сброса и собственно вероятности сброса RED, то есть алгоритм относится к классу P(PM).

Отметим, что когда вероятность сброса RED представлена ступенчатой функцией и  $w = 1$ ,  $\hat{q}$  становится эквивалентной  $q$ .

## 5.7. WRED

Взвешенный алгоритм произвольного раннего обнаружения (Weighted Random Early Detection, WRED) предоставляет различные уровни обслуживания пакетов в зависимости от вероятности их отбрасывания и обеспечивает избирательную установку параметров механизма RED на основании типа трафика.

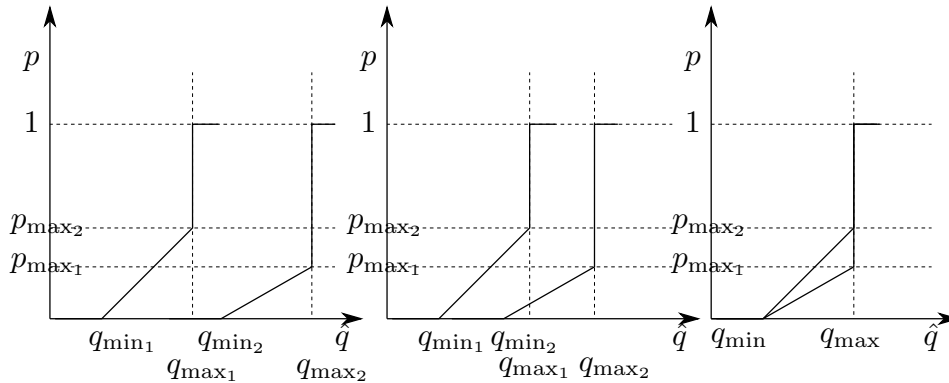


Рис. 11. Вид функции сброса в алгоритме WRED

Алгоритм WRED работает с единой очередью пакетов, для которой, как и в RED, по формуле (1) рассчитывается экспоненциально взвешенное скользящее среднее. Для каждого типа трафика задаются собственные параметры (пороговые значения, максимальный уровень сброса) и вычисляется вероятность сброса.

При этом пороговые значения длины очереди для различных уровней сброса пакетов могут быть полностью не совпадающими, частично перекрывающимися или полностью совпадающими.

Алгоритм относится к классу  $M(CS)$ .

### 5.8. Семейство алгоритмов RED with In / Out

В алгоритме RED with In / Out (RIO) (случайное раннее обнаружение с профильными / непрофильными пакетами) [16] поступающие пакеты делятся на профильные (IN) и непрофильные (OUT). Пакеты поступающего трафика определяются как IN-пакеты, если трафик находится в пределах заданной политики, и как OUT-пакеты, если трафик вышел за пределы заданной политики.

Для принятия решения о сбросе OUT-пакетов используется алгоритм RED относительно экспоненциально взвешенной скользящей средней длины общей очереди, а для принятия решения о сбросе IN-пакетов используется алгоритм RED относительно экспоненциально взвешенной скользящей средней длины виртуальной очереди только из IN-пакетов.

Алгоритм RED with In / Out and Coupled Virtual Queues (RIO-C) — случайное раннее обнаружение с профильными / непрофильными пакетами и парными виртуальными очередями так же, как и алгоритм RIO, оперирует понятиями IN- и OUT-пакетов. В данном алгоритме для пакетов с различным приоритетом используются разные ограничения, что позволяет пакетам с более высоким приоритетом обслуживаться быстрее. Кроме того, в алгоритме используется задание вероятности сброса пакета, которая может увеличиваться при росте интенсивности трафика какого-либо приоритета.

Принятие решения о сбросе IN-пакетов с приоритетом  $j$ ,  $j < n$  зависит от значения экспоненциально взвешенной скользящей средней длины виртуальной очереди, состоящей только из IN-пакетов с приоритетом, меньшим или равным  $j$ . Принятие решения о сбросе OUT-пакетов с приоритетом  $n$  зависит от средней занятости общей (физической) очереди.

Алгоритм RED with In/Out and De-Coupled Queues (RIO-DC) — случайное раннее обнаружение с профильными / непрофильными пакетами и непарными виртуальными очередями так же, как и алгоритм RIO, оперирует понятиями IN- и OUT-пакетов. В этом алгоритме экспоненциально взвешенный скользящий средний размер очереди для пакетов определённого типа рассчитывается на основе их числа в очереди. Остальные параметры RED выбираются в зависимости от режима, заданного для различных уровней сброса пакетов.

Алгоритмы этого семейства относятся к классу  $M(CP)$ .

### 5.9. SF-RED

Алгоритм SF-RED [17] использует одну физическую и  $i$  (в соответствии с числом классов трафика) виртуальных очередей, для каждой из которых заданы свои пороговые значения  $q_{\min i}$  и  $q_{\max i}$ , а функция сброса вычисляется по формуле

$$p_i = \begin{cases} 0, & \hat{q}_i < q_{\min i}, \\ \frac{\hat{q}_i - q_{\min i}}{q_{\max i}}, & q_{\min i} \leq \hat{q}_i < q_{\max i}, \\ 1 & \hat{q}_i > q_{\max i}, \end{cases} \quad (9)$$

где  $\hat{q}_i$  — экспоненциально взвешенная скользящая средняя длина виртуальной очереди  $i$ , зависящая от изменяемого значения веса  $w_i$ .

Для каждого интервала времени и класса  $i$

$$q_{\min i} = \begin{cases} q_{\min i} - \alpha, & \hat{q}_i > q_{tr} \\ q_{\min i} + \alpha, & \hat{q}_i < q_{tr} \end{cases} \quad (10)$$

$$p_{\max_i} = \begin{cases} p_{\max_i} + \beta, & \frac{\hat{q}_i}{\hat{q}_1} < w_{ir}, \quad i \neq 1 \\ p_{\max_i} - \beta, & \frac{\hat{q}_i}{\hat{q}_1} > w_{ir}, \quad i \neq 1 \end{cases} \quad (11)$$

где  $\hat{q}_t$  — целевое значение экспоненциально взвешенной скользящей длины физической очереди,  $\alpha = 1$ ,  $\beta = 0, 1$ ,  $q_{tr} = [(1 + 10\%) \hat{q}_t, (1 - 10\%) \hat{q}_t]$ ,  $w_{ir} = [(1 + 10\%) w_i, (1 - 10\%) w_i]$

## 6. Заключение

В заключение приведём сводную табл. 1 распределения рассмотренных в предыдущем разделе алгоритмов семейства RED по классам.

Распределение алгоритмов семейства RED по классам

Таблица 1

Класс	M		Q	P				
Алгоритм	CS	CP		NL	NLA	RS	PA	PM
WRED	+	—	—	—	—	—	—	—
RIO	—	+	—	—	—	—	—	—
SF-RED	—	+	—	—	+	—	—	—
ARED	—	—	—	+	—	—	—	—
RARED	—	—	—	+	—	—	—	—
POWARED	—	—	—	+	—	—	—	—
ERED	—	—	+	—	—	—	—	—
DSRED	—	—	—	—	+	—	—	—
GREED	—	—	—	—	—	+	—	—
SDRED	—	—	+	—	—	+	—	—
SRED	—	—	—	—	—	—	—	+

## Литература

1. *Floyd S., Jacobson V.* Random Early Detection Gateways for Congestion Avoidance // IEEE/ACM Transactions on Networking. — 1993. — No 1(4). — Pp. 397–413.
2. *Королькова А. В., Кулябов Д. С.* Предварительная классификация алгоритмов семейства RED // Материалы XII научной конференции МГТУ «Станкин» и «Учебно-научного центра математического моделирования МГТУ «Станкин» – ИММ РАН» по математическому моделированию и информатике: Программа. Сборник докладов. / под ред. О. Казакова. — М.: ИЦ ГОУ ВПО МГТУ «Станкин», 2009. — С. 125–128.
3. *Floyd S., Gummadi R., Shenker S.* Adaptive RED: An Algorithm for Increasing the Robustness of RED's Active Queue Management. — 2001. — <http://www.icir.org/floyd/papers/adaptiveRed.pdf>.
4. *Floyd S.* Explicit Congestion Notification (ECN) mechanism in the TCP/IP protocol // ACM Computer Communications Review. — 1994. — Vol. 24.
5. *Floyd S.* RED: Discussions of Setting Parameters. — 1997. — <http://www.aciri.org/floyd/REDparameters.txt>.
6. *Que D., Chen Z., Chen B.* An Improvement Algorithm Based on RED and Its Performance Analysis // IEEE. — 2008. — Pp. 2005–2008.
7. Techniques for Eliminating Packet Loss in Congested TCP/IP Network: Techrep CSE-TR-349-97 / W. Feng, D. Kandlur, D. Saha, K. Shin / U. Michigan. — 1997.

8. A Self-Configuring RED Gateway / W. Feng, D. Kandlur, D. Saha, K. Shin // Infocom. — 1999.
9. Floyd S., Gummadi R., Shenker S. Adaptive RED: An Algorithm for Increasing the Robustness of RED's Active Queue Management. — 2001. — [www.icir.org/floyd/papers/adaptiveRed.pdf](http://www.icir.org/floyd/papers/adaptiveRed.pdf).
10. Kim T.-H., Lee K.-H. Refined Adaptive RED in TCP/IP Networks // SICE-ICASE International Joint Conference. — Bexco, Busan, Korea: 2006.
11. POWARED for Non-Linear Adaptive RED / B. K. Ng, M. S. Uddin, A. A. Y. M. Abusin, D. Chieng // Asia-Pacific Conference on Communications, Perth, Western Australia, 3 - 5 October 2005. — 2005.
12. Zheng B., Atiquzzaman M. DSRED: A New Queue Management Scheme for the Next Generation Internet // IEICE TRANS. COMMUN. — 2006. — Vol. E89-B, No 3.
13. Iannaccone G., May M., Diot C. Aggregate traffic performance with active queue management and drop from tail // SIGCOMM Comput. Commun. Rev. — 2001. — Vol. 31, No 3. — Pp. 4–13. — ISSN 0146-4833.
14. Ryoo I., Yang M. A State Dependent RED: An Enhanced Active Queue Management Scheme for Real-Time Internet Services // IEICE Trans. Commun. — 2006. — Vol. E89-B, No 2. — Pp. 614–617.
15. Ott T., Lakshman T., Wong L. SRED: Stabilized RED // Proceedings of INFOCOM'99. — 1999. — Pp. 1346–1355.
16. Clark D. D., Fang W. Explicit Allocation of Best-Effort Packet Delivery Service // IEEE/ACM Transactions on Networking. — 1998. — Vol. 6, No 4. — Pp. 362–373. — <http://nms.lcs.mit.edu/6829-papers/p362-clark.pdf>.
17. Wu H.-M., Wu C.-C., Lin W. SF-RED — a Novel Server-based AQM to Provide Inter-server Fairness Service // Proceedings of the 12th International Conference on Parallel and Distributed Systems (ICPADS'06). — 2006.

UDC 004.021, 519.2

## On the Classification of RED Algorithms

A. V. Korolkova, D. S. Kulyabov, A. I. Tchernov

*Telecommunication Systems Department  
Peoples' Friendship University of Russia  
Miklukho-Maklaya str., 6, Moscow, Russia, 117198*

The RED (Random Early Detection) based algorithm classification is suggested in the article. Nearly all variants of RED modifications are orthogonal and can be compound.

**Key words and phrases:** Random Early Detection (RED), Weighted RED (WRED), Enhanced RED (ERED), Adaptive RED (ARED), Refined ARED, POWARED, Gentle RED (GRED), Double Slope RED (DSRED), State Dependent RED (SDRED), Stabilized RED (SRED), RED with In/Out (RIO), SF-RED.