

МОДЕЛИРОВАНИЕ РАБОТЫ МЕТОДОВ УПРАВЛЕНИЯ ОЧЕРЕДЯМИ МАРШРУТИЗАТОРА В СЕТЯХ ТСП/IP

Мануйлова Л.В., студент; Батыр С.С., ассистент

(ГВУЗ «Донецкий национальный технический университет», г. Донецк, Украина)

Неравномерный рост скоростей каналов передачи данных неизбежно приводит к возникновению «узких» мест в телекоммуникационной сети и соответственно к возникновению перегрузок, особенно при подключении сетей доступа к транспортной сети. Предметом исследования в данной статье является маршрутизатор, одной из задач которого является предотвращение перегрузок в канале. Именно маршрутизатор с непосредственно подключенным каналом передачи данных обладает необходимой информацией о возникновении перегрузки в канале или состоянии, которое может повлечь за собой ее наличие. Маршрутизатор оценивает текущую степень загруженности выходной очереди и текущий рост или спад интенсивности нагрузки, и сообщает ТСП передатчикам о необходимости снижения окна перегрузки. Для явного уведомления о перегрузке, путем соответствующей маркировки проходящих пакетов, используется протокол ECN (ExplicitCongestionNotification), описанный в документе RFC-3168. Если отсутствует поддержка ECN, то при угрозе перегрузки вместо маркировки пакеты могут сбрасываться в случайном порядке. В противном случае переполнение очереди приведет к множественным потерям пакетов и последующим периодическим чередованиям моментов переполнения и опустошения буфера маршрутизатора.

В условиях конкуренции за полосу пропускания большого числа одновременных ТСП сессий, принадлежащих разным приложениям с различным поведением, на входе агрегирующего порта маршрутизатора возникает трафик со сложной нелинейной динамикой. В данной статье рассматривается случай, когда конкурирующие соединения имеют одинаковый класс обслуживания и попадают в одну очередь, сравниваются различные современные методы управления очередями и проверяется их влияние на параметры качества передачи данных путем имитационного моделирования в симуляторе ns-2.

Наиболее часто в сетях передачи данных используется пассивный механизм сброса излишка пакетов данных при переполнении очереди (DropTail). При передаче ТСП трафика этот механизм приводит к возникновению явления глобальной синхронизации, когда при переполнении буфера маршрутизатора сбрасываются все приходящие пакеты, и все ТСП-передатчики одновременно уменьшают размер ТСП-окна, а потом синхронно его увеличивают, вызывая новую перегрузку. Кроме того, сброс пакетов в данном случае происходит уже в момент перегрузки, и не предпринимается активных действий для её предотвращения. Однако, в современных маршрутизаторах наиболее распространены методы активного управления очередью (AQM), заблаговременно предотвращающие переполнение буфера маршрутизатора.

В данной статье рассматривается механизм раннего обнаружения перегрузок RED (RandomEarlyDetection) [1,2,3], в случайном порядке сбрасывающий пакеты с вероятностью, линейно увеличивающейся с ростом усредненной длины очереди, и его адаптивная модификация ARED (AdaptiveRED) [4].

Функции вероятностисброса/маркировки p для исследуемых алгоритмов описываются следующим образом (q – длина очереди (пакет), C – пропускная способность канала (пакет/с)):

а) DropTail

$$p = \begin{cases} 0 & \text{если } q < q_{\max} \\ 1 & \text{если } q > q_{\max} \end{cases}, \quad (1)$$

где q_{max} – максимально допустимый размер очереди.

б) RED

RED случайно отбрасывает или маркирует поступающие пакеты, когда среднее значение длины очереди avg превышает минимальный порог min_{th} . Вероятность отказа растет с увеличением средней длины очереди вплоть до значения максимальной вероятности сброса пакета max_p . Когда средняя длина очереди достигает значение максимального порога max_{th} , все пакеты получают отказ. Т.к. avg варьируется от min_{th} до max_{th} , вероятность сброса/маркировки изменяется линейно от 0 до max_p :

$$p_b = \frac{avg - min_{th}}{max_{th} - min_{th}} \cdot max_p \quad (2)$$

Финальная вероятность сброса пакетов p_a медленно увеличивается с ростом числа пакетов $count$, которые поступили с момента сброса последнего пакета:

$$p_a = \frac{p_b}{1 - count \cdot p_b} \quad (3)$$

Средняя длина очереди, которая рассчитывается каждый раз по поступлению нового пакета, определяется следующим образом:

$$avg = (1 - w_q) \cdot \overline{avg} + w_q \cdot q, \quad (4)$$

где w_q – весовой коэффициент очереди, \overline{avg} – предыдущее значение средней длины очереди, q – мгновенное значение длины очереди.

Таким образом, функция вероятности сброса пакета принимает следующий вид:

$$p = \begin{cases} \frac{avg - min_{th}}{max_{th} - min_{th}} \cdot max_p & \text{если } min_{th} < avg < max_{th} \\ 0 & \text{если } avg < min_{th} \\ 1 & \text{если } avg > max_{th} \end{cases} \quad (5)$$

На рисунках 1а и 1б представлены вышеописанные функции алгоритмов:

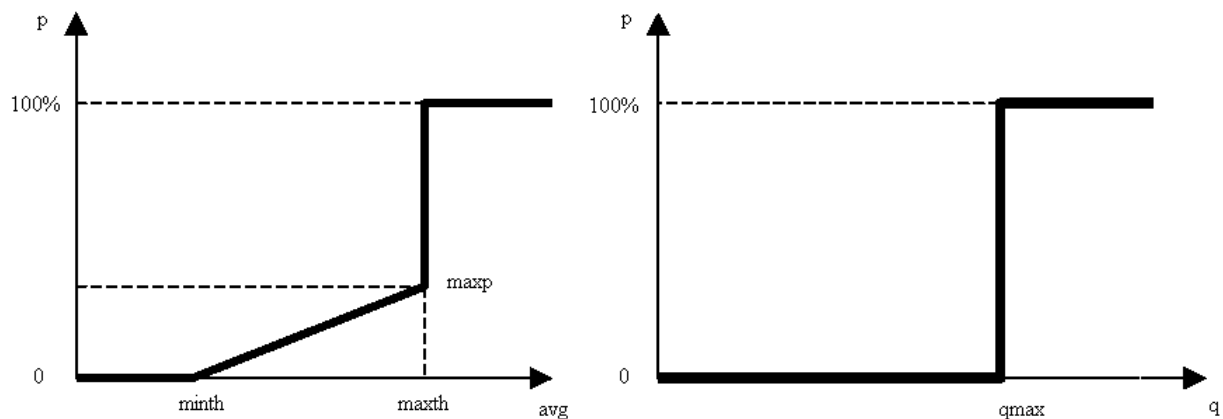


Рисунок 1 – а) функция сброса пакета для алгоритма RED, б) функция сброса пакета для алгоритма DropTail

Однако, недостатком RED является зависимость его поведения от установки фиксированных параметров, которые достаточно сложно подобрать, чтобы они адекватно соответствовали характеру поведения обрабатываемого трафика.

в) ARED

Для адаптивной подстройки параметров был предложен механизм ARED, который путем минимальных изменений исходного алгоритма, решает эту проблему, динамически изменяя max_p в пределах от 1-50% в зависимости от средней загруженности очереди по принципу AIMD:

$$max_p = \begin{cases} max_p + \alpha & \text{если } avg > target \text{ и } max_p \leq 0.5 \\ max_p \cdot \beta & \text{если } avg < target \text{ и } max_p > 0.01 \end{cases}, \quad (6)$$

где

$$target = [min_{th} + 0.4 \cdot (max_{th} - min_{th}), min_{th} + 0.6 \cdot (max_{th} - min_{th})], \quad (7)$$

$$\alpha = \min(0.01; \frac{max_p}{4}), \beta = 0.9 \quad (8)$$

Чтобы уменьшить необходимость настройки других параметров RED, они также рассчитываются автоматически. Определены следующие процедуры для max_{th} и w_q :

$$max_{th} = 3 \cdot min_{th}, \quad (9)$$

$$w_q = 1 - \exp(-\frac{1}{C}) \quad (10)$$

Имитационное моделирование проводилось с использованием NS-2. Для оценки влияния алгоритмов управления очередью, а именно DropTail, RED, ARED, на качество передачи в сети было проведено моделирование перегрузки в канале между двумя маршрутизаторами, через которые передавался мультисервисный трафик 2-х типов: длительные TCP-сессии, создаваемые 100 одновременными FTP приложениями; короткие TCP-сессии, создаваемые HTTP источниками.

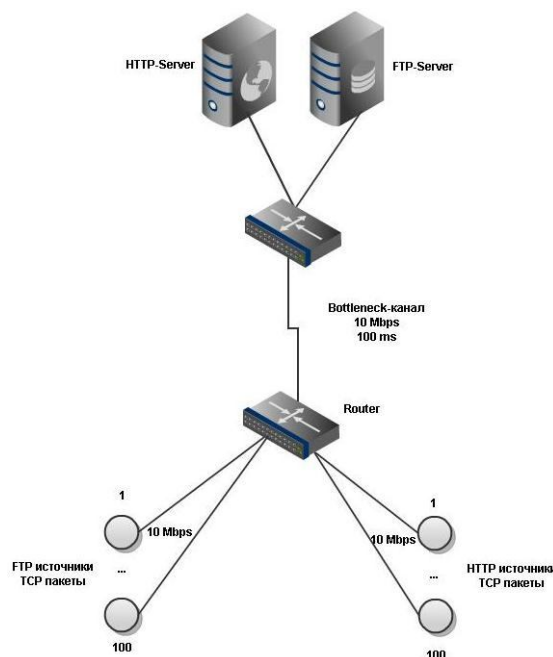
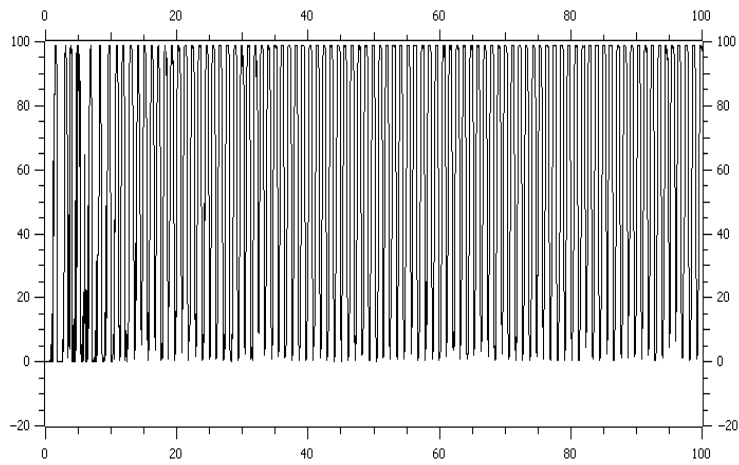


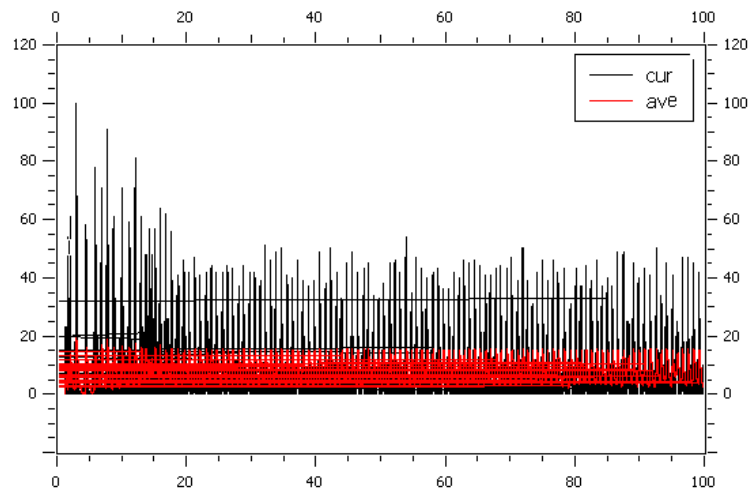
Рисунок 2 – Топология сети для моделирования

Задержка в каналах передачи распределена по равномерному закону в интервале от 1 до 20 мс. Скорость канала между маршрутизаторами ограничена 10 Мбит/с. Использовалась реализация протокола TCP Reno.

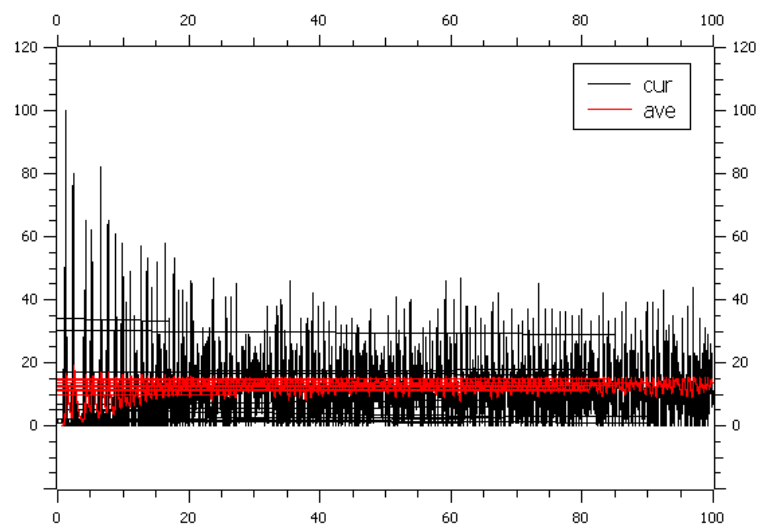
Результаты моделирования представлены на следующих графиках (рис.2-4) и в табл. 1:



а)



б)



в)

Рисунок 3 – а) длина очереди (пакет/с) при DropTail, б) и в) мгновенная и средняя длина очереди (пакет/с) при RED и ARED

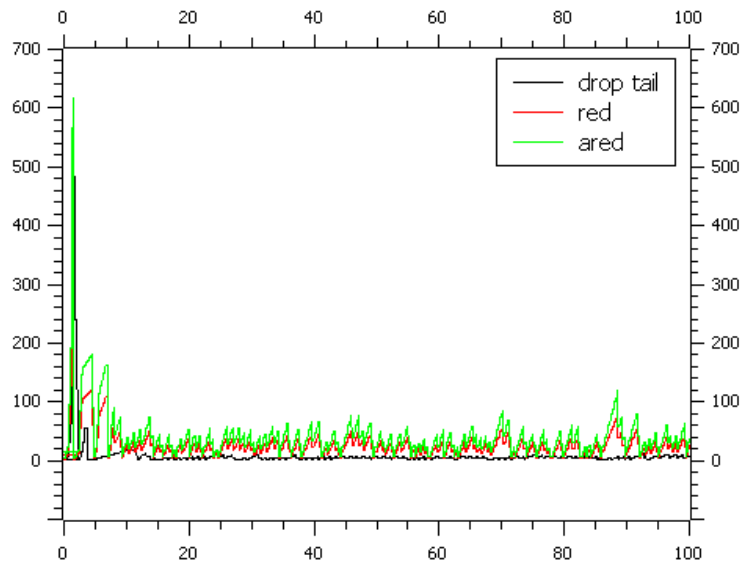


Рисунок 4 – Размер TCP-окна (пакет/с) для исследуемых алгоритмов

Таблица 1 – Параметры исследуемого канала с перегрузкой в зависимости от используемого алгоритма обслуживания очереди при заданной длине очереди 100 пакетов

Алгоритм	Средний размер очереди, пакет	Потери пакетов, %	Коэффициент использования канала
DropTail	88	2,16	0,999
RED	59	0,45	0,998
ARED	47	0,35	0,997

На основании полученных данных моделирования был проведен анализ исследуемых методов управления очередью, который позволяет утверждать, что использование в маршрутизаторах алгоритмов RED и ARED является более рациональным, чем DropTail, т.к. данные алгоритмы позволяют минимизировать дрожание задержки пакетов путем контроля среднего размера очереди, большую стабильность интенсивности трафика и его непредвзятое обслуживание в случае кратковременных всплесков. Также, ARED за счет адаптированной подстройки параметра max_p и автоматической настройки max_{th} и w_q поддерживает предсказуемое значение средней длины очереди и снижает чувствительность параметров RED, уменьшается вероятность превышения верхнего порога и вариативность задержки. За счет чего, соответственно, увеличивается возможный объем переданных данных. Однако остается вопрос выбора диапазона, в пределах которого будет удерживаться очередь, и который должен соответствовать компромиссу между коэффициентом использования и величиной задержки в канале связи.

Перечень ссылок

1. Floyd S. and Jacobson V., Random Early Detection gateways for Congestion Avoidance, IEEE/ACM Transaction on Networking, August 1993.
2. Misra V., Gong W., and Towsley D., Fluid-based Analysis of a network of AQM Routers Supporting TCP Flows with an Application to RED, in Proceedings of ACM/SIGCOMM, 2000.
3. Батыр С.С. Управление длиной очереди на пограничном маршрутизаторе участка сети Интернет / С.С. Батыр - Збірник наукових праць ДонІЗТ, № 30 .- Донецьк, 2012 - С.71-79
4. Floyd S., Gummadi R., and Shenker S., Adaptive RED: An Algorithm for Increasing the Robustness of RED's Active Queue Management, August 2001.