

Сравнительный анализ алгоритмов активного управления очередью семейства RED в средстве моделирования сетей NS-2

А. Г. Саргсян*, Б. А. Тагиев*

** Кафедра теории вероятностей и кибербезопасности,
Российский университет дружбы народов,
ул. Миклуто-Маклая, д.6, Москва, Россия, 117198*

Email: 1032201740@rudn.university, 1032200531@rudn.university

Сетевые маршрутизаторы используют алгоритмы управления очередями для обеспечения высокого уровня качества обслуживания (QoS). Это обеспечивает эффективное распределение пропускной способности сети и отвечает требованиям пользователей к скорости передачи данных и надежности. На сегодняшний день использовались различные алгоритмы управления.

В рамках этой работы мы провели анализ производительности системы случайного раннего обнаружения (RED), включая ее различные модификации, с помощью комплексного моделирования. Наш анализ включает сравнение с другими методами управления очередями с использованием таких критериев, как длина очереди, задержка, вариабельность задержки и размер окна для типов протоколов TCP, таких как Reno, NewReno и Vegas. Использование анализа на основе моделирования в симуляторе NS-2 позволяет нам точно оценивать производительность и действенность различных настроек и реализаций алгоритма RED. Мы модифицировали исходный код симулятора NS-2 для проведения анализа на основе моделирования.

Анализ результатов, полученных с использованием программного обеспечения Gnuplot, демонстрирует, что алгоритмы RED обеспечивают значительные улучшения по сравнению с традиционным механизмом drop tail, особенно с точки зрения управления задержкой и ее изменчивостью, а также уменьшения потери пакетов. Эти выводы подтверждаются количественными данными и визуальными представлениями, что делает данное исследование актуальным для разработчиков сетевого оборудования, стремящихся усовершенствовать процедуры управления трафиком.

Ключевые слова: управление очередями, алгоритм RED, NS-2.

1. Введение

Сетевой маршрутизатор поддерживает набор очередей, в которых хранятся пакеты, которые должны достичь указанного адресата. Исторически в этих очередях использовался алгоритм пассивного управления очередью DropTail, который помещает пакет в очередь только в том случае, если он еще не достиг своей максимальной емкости или отбрасывает его в противном случае.

Алгоритмы активного управления очередью семейства RED корректируют отбрасывание или маркировку пакетов до того, как очередь заполнится. Данные алгоритмы работают, поддерживая вероятности отбрасывания или метки отбрасывания для каждого пакета, даже если очередь еще не заполнена.

2. Алгоритм Random Early Detection

2.1. Краткий обзор алгоритма

RED (Random Early Detection, произвольное раннее обнаружение) — алгоритм активного управления очередью (AQM) для управления переполнением очередей маршрутизаторов с возможностью предотвращения перегрузок.

Вероятность p_b маркировки на отбрасывание пакетов представляет собой функцию, линейно зависящую от \hat{q} (средневзвешенное скользящее среднее), 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} . Более подробное описание алгоритма приведено в [3]. Вычисляется вероятность следующим образом (1):

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

Одна из фундаментальных проблем, с которой сталкивается RED, заключается в том, что оценка рабочей нагрузки зависит от длины очереди. Хотя постоянная очередь указывает на перегруженную систему, ее длина дает очень слабое представление о серьезности этой перегрузки.

Впоследствии были реализованы множество вариаций, модифицировавших исходный алгоритм, среди основных модификаций можно выделить GRED, сглаживающий количество потерянных пакетов при достижении максимального порога, DS-RED, NLRED и другие варианты с линейной и нелинейной вероятностной функцией сброса, а также группа адаптивных алгоритмов, при которых параметр p_{\max} меняется в зависимости от нагруженности трафика [1].

2.2. Реализация в NS-2

В качестве эталонной модели была использована топология из 20 TCP источников и 20 приемников в средстве моделирования NS-2 [2]. Получили графики размера окна перегрузки при разных TCP в одинаковых условиях, см. рис. 1. Значительно отличающийся вид Vegas обуславливается тем, что в то время, как Reno и NewReno основаны на принципе уменьшения скорости передачи данных при обнаружении потерь пакетов, Vegas использует измерение задержки для обнаружения и предотвращения перегрузки, стремясь избежать потерь пакетов, регулируя скорость передачи до их возникновения.

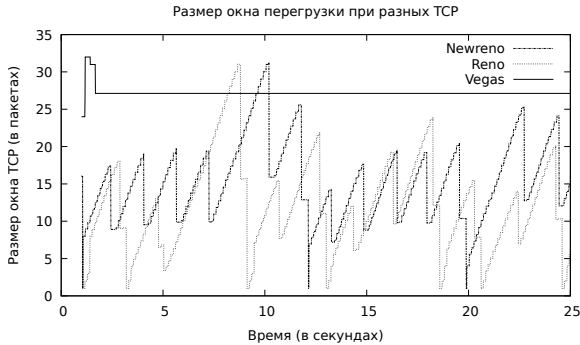


Рис. 1. Размер окна перегрузки на 1 источнике при разных TCP

Для проведения сравнительного анализа различных алгоритмов, относящихся к семейству RED, было выполнено моделирование аналогичной топологии. Получили следующие значения размера окна перегрузки, см. рис. 2. В данном примере все алгоритмы, кроме DS-RED, показали крайне схожие результаты обработки очереди. DS-RED начинает отбрасывать пакеты более агрессивно, т.к. имеет промещуточное значение q_{mid} , причем $q_{mid} = 0.5(q_{\max} - q_{\min})$.

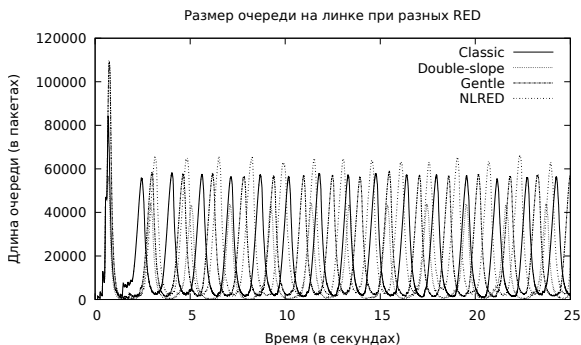


Рис. 2. Длина очереди при разных алгоритмах RED

Кроме того, были изучены результаты, полученные для адаптивных вариантов алгоритмов, детализированные на рисунке 3. Адаптивные алгоритмы позволяют избежать минусов классического алгоритма RED, путем:

- Автоматической установки минимального порога q_{\min} . Он устанавливается в зависимости от пропускной способности канала и задержки целевой очереди.
- Автоматической установки максимального порога q_{\max} . Он устанавливается в зависимости от значения q_{\min} , причем $q_{\max} = 3 * q_{\min}$.
- Адаптивной настройки p_{\max} . Он адаптирован в соответствии с текущей средней длиной очереди.

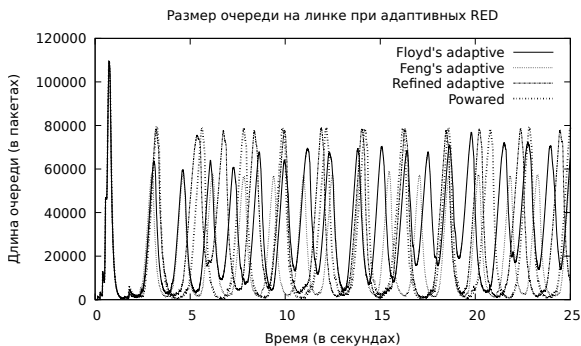


Рис. 3. Адаптивные варианты RED

3. Заключение

В результате моделирования работы алгоритмов RED в рамках эталонной модели можно сделать вывод, что адаптивные алгоритмы RED справляются лучше с перегрузкой и обработкой очередью, что благоприятно сказывается на общем состоянии сети и передачи данных. В рамках нашей эталонной модели выбор адаптивного алгоритма является более выгодным, однако не стоит брать данный пример как правило и для иной ситуации может показать преимущество другой алгоритм.

Литература

1. А. В. Королькова, Д. С. Кулябов, А. И. Черноиванов, “К вопросу о классификации алгоритмов RED,” *Вестник РУДН. Серия «Математика. Информатика. Физика»*, стр. 34-46, янв. 2009.
2. David X. Wei, Pei Cao, “NS-2 TCP-Linux,” in *Proceeding from the 2006 workshop on ns-2: the IP network simulator - WNS2 '06*. ACM Press, 2006. DOI: 10.1145/1190455.1190463.
3. S. Floyd, V. Jacobson, “Random early detection gateways for congestion avoidance,” *IEEE/ACM Transactions on Networking*, vol. 1, no. 4, стр. 397–413, 1993. DOI: 10.1109/90.251892.

UDC 004.94

Comparative analysis of active queue management algorithms of the RED family in the NS-2 network modeling tool

A. G. Sargsyan*, B. T. Tagiev*

* *Department of Probability Theory and Cyber Security
Peoples' Friendship University of Russia
Miklukho-Maklaya str. 6, Moscow, 117198, Russia*

Email: 1032201740@rudn.university, 1032200531@rudn.university

Network routers use queue management algorithms to ensure a high level of Quality of Service (QoS). This ensures efficient allocation of network bandwidth and meets user requirements for data transfer speeds and reliability. Various control algorithms have been used to date.

Within the scope of this work, we have conducted a performance analysis of Random Early Detection (RED), including its various modifications, through complex modeling. Our analysis includes a comparison to other queue management techniques using criteria such as queue length, latency, variability of latency, and window sizing for TCP protocol types such as Reno and Vegas. The use of simulation-based analysis on the NS-2 simulator allows us to accurately evaluate the performance and efficacy of various RED algorithm settings and implementations. We modified the source code of the NS-2 simulator to conduct simulation-based analyses.

An analysis of the results generated using the Gnuplot software demonstrates that the RED algorithms provide significant improvements over the traditional drop tail mechanism, particularly in terms of latency management and its variability, as well as the reduction of packet loss. These findings are supported by quantifiable data and visual representations, making this study relevant to network equipment developers seeking to enhance traffic management procedures.

Key words and phrases: active queue management, RED algorithm, NS-2.