

Обзор и сравнение алгоритмов активного управления трафиком семейства RED с линейной функцией сброса

Г.А. Легиньких^{1,*}, И. С. Зарядов¹

¹Российский университет дружбы народов имени Патриса Лумумбы, ул. Миклухо-Маклая, д.6, Москва, Россия, 117198

Аннотация

В статье рассматриваются алгоритмы активного управления трафиком семейства RED с линейной функцией сброса и с одной очередью сброса. Описаны их основные принципы работы. В форме таблицы представлены ключевые формулы, используемые для вычисления вероятности сброса пакетов в зависимости от заданных параметров, таких как минимальный и максимальный порог загрузки сети и др. Для проведения сравнительного анализа был разработан программный инструмент на языке Python, автоматизирующий тестирование и анализ работы алгоритмов при различных введенных условиях. Программа позволяет оценить влияние параметров на производительность при определенном алгоритме. Результаты показали, что такие модификации RED, как ARED, Re-ARED, FARED и др. могут при определенных обстоятельствах существенно снизить потери пакетов. Полученные данные могут быть полезны для оптимизации сетевых ресурсов в нынешних системах телекоммуникации.

Ключевые слова

активное управление очередями, перегрузка сетей, алгоритм RED, линейная функция сброса, алгоритмы семейства RED

1. Введение

Растущий объем трафика вызвал проблему перегрузки в современных сетях передачи данных. С целью решения проблемы перегрузки были разработаны алгоритмы активного управления очередями (AQM), позволяющие управлять сетевыми ресурсами и предотвращать перегрузки на ранних стадиях. Одним из первых таких алгоритмов был алгоритм RED. Алгоритмы семейства RED (Random Early Detection) позволяют минимизировать задержки и потери пакетов, тем самым обеспечивая стабильную сеть. Тем не менее выбор и настройка подходящего алгоритма остается сложной задачей из-за разнообразия сетевых параметров и условий трафика.

Для улучшения производительности RED были разработаны его модификации. Каждый этот алгоритм направлен на устранение недостатков классического алгоритма. Задачей этой работы является рассмотрение некоторых модификаций алгоритма RED с линейной функцией сброса и с одной очередью сброса, включая Adaptive RED (ARED) [1], Refined Adaptive RED (Re-ARED) [2], Fast Adapting RED (FARED) [3], Gentle RED (GRED) [4], Random early dynamic detection (REDD) [5], Double Slope RED (DSRED) [6] and Adaptive GRED (AGRED) [7]. Эти алгоритмы используют различные подходы для управления сбросом, такие как адаптивное изменение параметров, а так же динамическое регулирование порогов.

Основное внимание уделяется сравнению их подходов и эффективности в условиях различных сетевых нагрузок. Для автоматизации процесса анализа и сравнения алгоритмов был разработан программный инструмент на языке Python, позволяющий строить таблицу и сравнивать алгоритмы в разнообразных сетевых сценариях.

Information and Telecommunication Technologies and Mathematical Modeling of High-Tech Systems 2025 (ITTMM 2025), Moscow, April 07–11, 2025

*Автор, отвечающий за публикацию.

✉ galina.leginkikh@gmail.com (Г.А. Легиньких); izaryadov@gmail.com (И. С. Зарядов)



© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

Данный подход помогает детально изучить поведение того или иного алгоритма при различных условиях. Проведенная работа имеет практическую ценность для оптимизации сетевых ресурсов.

2. Основная часть

Для более подробного анализа и понимания механизмов работы алгоритмов в таблице представлены ключевые формулы, которые используются для их функционирования. Это позволяет наглядно увидеть различия в подходах. (см. табл. 1)

Таблица 1: Описание алгоритмов

Алгоритм	Формула
RED	$P_b = \begin{cases} 0, & 0 < \text{avg} \leq \text{min_th} \\ \frac{\text{avg} - \text{min_th}}{\text{max_th} - \text{min_th}} \cdot \text{max_p}, & \text{min_th} < \text{avg} \leq \text{max_th} \\ 1, & \text{avg} > \text{max_th} \end{cases}$
ARED	$\text{max_p} = \begin{cases} \text{max_p} + \alpha, & \text{avg_new} > \text{target} \\ \text{max_p} \times \beta, & \text{avg_new} < \text{target} \end{cases}$ $\text{max_p} = \max(0.01, \min(\text{max_p}, 0.5))$ $\alpha = \min(0.01, \frac{\text{max_p}}{4}), \quad \beta = 0.9$ $\text{target} \in [\text{min_th} + 0.4 \cdot (\text{max_th} - \text{min_th}), \text{min_th} + 0.6 \cdot (\text{max_th} - \text{min_th})]$ <p>Формула вероятности аналогична RED.</p>
Re-ARED	<p>max_p аналогично ARED, но другие коэффициенты:</p> $\alpha = 0.25 \cdot \text{max_p} \cdot \frac{\text{avg} - \text{target}}{\text{target}}$ $\beta = 1 - 0.17 \cdot \frac{\text{target} - \text{avg}}{\text{target} - \text{min_th}}$ $\text{target} \in [\text{min_th} + 0.48 \cdot (\text{max_th} - \text{min_th}), \text{min_th} + 0.52 \cdot (\text{max_th} - \text{min_th})]$ <p>Формула вероятности аналогична RED.</p>
FARED	<p>max_p аналогично ARED, но другие коэффициенты</p> $\alpha = 0.0412 \cdot \text{max_p} \cdot \frac{\text{avg} - \text{target}}{\text{target}}$ $\beta = 1 - 0.0385 \cdot \frac{\text{target} - \text{avg}}{\text{target} - \text{min_th}}$ $\text{target} \in [\text{min_th} + 0.48 \cdot (\text{max_th} - \text{min_th}), \text{min_th} + 0.52 \cdot (\text{max_th} - \text{min_th})]$ <p>Формула вероятности аналогична RED.</p>
GRES	$P_b = \begin{cases} 0, & \text{avg} \leq \text{min_th} \\ \frac{\text{avg} - \text{min_th}}{\text{max_th} - \text{min_th}} \cdot \text{max_p}, & \text{min_th} < \text{avg} \leq \text{max_th} \\ \frac{\text{avg} - \text{max_th}}{\text{max_th}} \cdot (1 - \text{max_p}) + \text{max_p}, & \text{max_th} < \text{avg} \leq 2\text{max_th} \\ 1, & \text{avg} > 2\text{max_th} \end{cases}$
REDD	$\text{max_th} = \begin{cases} \text{max_th} - 2, & \text{avg} < \text{target_avg} \text{ и } \text{max_th} \geq 2 \\ \text{max_th} + 2, & \text{avg} > \text{target_avg} \text{ и } \text{max_th} \leq (K - \text{min_th}) \end{cases}$ <p>Формула вероятности аналогична RED.</p>

Продолжение на следующей странице

Продолжение таблицы 1

Алгоритм	Формула
DSRED	$P_b = \begin{cases} 0, & \text{avg} < \text{min_th} \\ \alpha(\text{avg} - \text{min_th}), & \text{min_th} \leq \text{avg} < \text{mid_th} \\ 1 - \gamma + \beta(\text{avg} - \text{mid_th}), & \text{mid_th} \leq \text{avg} < \text{max_th} \\ 1, & \text{avg} \geq \text{max_th} \end{cases}$ $\alpha = \frac{2(1-\gamma)}{\text{max_th} - \text{min_th}}$ $\beta = \frac{2\gamma}{\text{max_th} - \text{min_th}}$ $\text{mid_th} = 0.5(\text{max_th} + \text{min_th})$
AGRED	$P_b = \begin{cases} 0, & 0 < \text{avg} \leq \text{min_th} \\ \frac{\text{avg} - \text{min_th}}{\text{max_th} - \text{min_th}} \cdot \text{max_p}, & \text{min_th} < \text{avg} \leq \text{max_th} \\ \frac{\text{avg} - \text{max_th}}{\text{max_th}} \cdot \frac{(1 - \text{max_p})}{2} + \text{max_p}, & \text{max_th} < \text{avg} \leq 2\text{max_th} \\ 1, & \text{avg} > 2\text{max_th} \end{cases}$

Эти формулы показывают, как различные модификации алгоритмов семейства RED управляют вероятностью сброса пакетов в зависимости от текущего состояния сети. Все эти алгоритмы добавляют к классическому RED адаптивные параметры, которые позволяют более гибко реагировать на изменения.

Для дальнейшего сравнения алгоритмов был разработан инструмент на языке Python, который позволяет автоматизировать процесс тестирования и анализа поведения алгоритмов при любых заданных параметрах. Вывод программы включает в себя таблицу с результатами для каждого алгоритма. Сравнила такие показатели как, изменение базовых параметров, уровень потерь пакетов.

Пример работы программы при данных параметрах: avg_old = 10, min_th = 5, max_th = 20, queue_size = 100, w_q = 0.002, max_p = 0.5, gamma = 0.8. (см. рис. 1).

Алгоритм	avg	min_th	max_th	max_p	gamma	Вероятность сброса	Уровень потерь пакетов
0 RED	10.18	5	20	0.5	нет	0.1726667	17.2 %
1 ARED	10.18	5	20	0.45	нет	0.1554000	15.5 %
2 Re-ARED	10.18	5	20	0.473707	нет	0.1635867	16.3 %
3 FARED	10.18	5	20	0.494045	нет	0.1706103	17.0 %
4 GRED	10.18	5	20	0.5	нет	0.1726667	17.2 %
5 REDD	10.18	5	18	0.5	нет	0.1992308	19.9 %
6 DSRED	10.18	5	20	нет	0.8	0.1381333	13.8 %
7 AGRED	10.18	5	20	0.5	нет	0.1726667	17.2 %

Рис. 1: Алгоритмы

RED: Базовый алгоритм с вероятностью сброса 17.2%. Это отправная точка для сравнения, показывающая средний уровень потерь при заданных параметрах.

ARED: Благодаря адаптивной настройке максимальной вероятности сброса, этот алгоритм снижает вероятность сброса до 15.5%, что указывает на лучшую способность адаптироваться к изменениям условий сети.

Re-ARED: Показал немного большую вероятность сброса, а точнее 16.3%, по сравнению с ARED из-за другого подхода в расчетах дополнительных параметров.

FARED: Вероятность сброса почти такая же, как у RED, что показывает, что модификация мало влияет на общую производительность.

GRED: Поскольку его показатели идентичны RED, GRED не показывает значительных преимуществ при данной нагрузке, но может быть полезен в других условиях.

REDD: Этот алгоритм показал увеличенную вероятность сброса, указывая на более агрессивное управление очередью, что может быть полезно в условиях высокой перегрузки, но приводит к большому количеству потерь пакетов.

DSRED: Показал наименьшую вероятность сброса (13.8%), демонстрируя более гибкую функцию сброса за счет новых параметров.

AGRED: Показатели идентичны RED при данных условиях.

3. Заключение

В ходе работы было исследовано несколько модификаций алгоритма RED, включая ARED, Re-ARED, FARED и других. Сравнение их эффективности показало, что рассмотренные алгоритмы улучшают производительность сети по сравнению с RED, в некоторых случаях снижая потери пакетов. Разработанная программа на Python предоставляет инструмент для автоматизированного тестирования и анализа этих алгоритмов.

Финансирование: Данное исследование не получало внешнего финансирования.

Список литературы

1. Floyd, S., Gummadi, R., Shenker, S. и др. *Adaptive RED: An algorithm for increasing the robustness of RED's active queue management* 2001.
2. Kim, T.-h. & Lee, K.-h. *Refined Adaptive RED in TCP/IP Networks* в *2006 SICE-ICASE International Joint Conference* (2006), 3722—3725. doi:10.1109/SICE.2006.314633.
3. Tahiliani, M. P., Shet, K. C. & Basavaraju, T. G. *FARED: Fast Adapting RED Gateways for TCP/IP Networks* в *Advanced Computing, Networking and Security* (ред. Thilagam, P. S., Pais, A. R., Chandrasekaran, K. & Balakrishnan, N.) (Springer Berlin Heidelberg, Berlin, Heidelberg, 2012), 435—443.
4. Eguchi, T., Ohsaki, H. & Murata, M. *On control parameters tuning for active queue management mechanisms using multivariate analysis* в *2003 Symposium on Applications and the Internet, 2003. Proceedings.* (2003), 120—127. doi:10.1109/SAINT.2003.1183040.
5. Abdel-jaber, H., Thabtah, F. A., Woodward, M. E., Jaffar, A. D. & Bazar, H. A. A. *Random Early Dynamic Detection Approach for Congestion Control* 2014.
6. Zheng, B. & Atiquzzaman, M. *DSRED: improving performance of active queue management over heterogeneous networks* в *ICC 2001. IEEE International Conference on Communications. Conference Record (Cat. No.01CH37240)* **8** (2001), 2375—2379 vol.8. doi:10.1109/ICC.2001.936557.
7. Baklizi, M., Abdel-Jaber, H., Ramadass, S., Abdullah, L. & Anbar, M. Performance assessment of AGRED, RED and GRED congestion control algorithms. *Information Technology Journal* **11**, 255—261. doi:10.3923/itj.2012.255.261 (февр. 2012).