

Анализ алгоритмов управления очередями для улучшения информационного взаимодействия методом сетевого кодирования

Бритвин Н.В.^{1,3*}, Мешавкин К.В.^{2,1}**

¹*Московский авиационный институт (национальный исследовательский университет), МАИ, Волоколамское шоссе, 4, Москва, 125993, Россия*

²*Лаборатория Касперского, Ленинградское шоссе, 39А, стр. 3, Москва, 125212, Россия*

³*Центр информационных технологий в проектировании Российской академии наук, ул. Маршала Бирюзова, 7а, Одинцово, Московская область, 143000, Россия*

**e-mail: britvin.nickita@yandex.ru*

***e-mail: meshavkin1996@gmail.com*

Статья поступила 21.01.2020

Аннотация

Рассмотрены теоретические аспекты и взаимодействие метода сетевого кодирования и алгоритма управления переполнением очередей для повышения качества информационного взаимодействия. Проанализированы алгоритмы управления очередью Tail Drop и Random Early Detection в системах из Теории Массового Обслуживания, на основе чего был сделан вывод, что RED является достаточным для имитационного моделирования Систем Массового Обслуживания, а также было предложено рассмотреть модификации RED – WRED и CBWRED. Положены основы для создания имитационной модели. Исследовано взаимодействие метода сетевого кодирования и алгоритма управления очередью RED и выяснено, что состояния перегруженности сети возникать не будет. Для моделирования предлагается использовать среду MATLAB, задействовать алгоритм

кодирования и декодирования сетевого кода и симитировать работу буфера при использовании алгоритма управления очередью RED. Планируется получить зависимости воздействия кодовой последовательности пакетов на перегрузки в телекоммуникационных системах и сетях в условиях ограниченной пропускной способности и времени обработки пакета.

Ключевые слова: TCP/IP, AQM, network coded, Tail drop, Random early detection, Mobile ad hoc Network, теория массового обслуживания, сетевой код, перегрузка маршрутизатора, оптимизация сети.

Введение

Стремительное развитие мобильных устройств (здесь и далее будем понимать любые смарт устройства малых габаритов, способных осуществить подключение к телекоммуникационной сети) и наращивание темпов их производства приводит к транспортному коллапсу в телекоммуникационной сети. Уменьшение форм-фактора таких устройств задает новый тренд их производства, нацеленный на беспроводной доступ к всемирной сети. В связи с чем уже наблюдается проблема с частотным диапазоном, что в свою очередь негативно сказывается на информационном взаимодействии [1, 2, 3]. Устройств так много, что на одном и том же канале стандарта 802.11xx работает их большое количество.

Помимо вышесказанного, стек протоколов TCP/IP, как и большинство стандартов и алгоритмов передачи и формирования пакетов, были нацелены на

проводной сегмент, который, в свою очередь, лишен недостатков беспроводного сегмента [4]. В данной статье речь пойдет об алгоритмах Active Queue Management (AQM) [5, 6, 7, 8] для управления переполнением очередей маршрутизаторов. В частности, о двух самых распространенных - Random early detection (RED) и Tail drop. В качестве поля для исследования предлагается рассмотреть беспроводные децентрализованные самоорганизующиеся сети MANET (Mobile ad hoc Network) [9, 10].

Большинство устройств MANET используют алгоритм управления очередью Tail drop [11], в которой пакеты принимаются до тех пор, пока очередь не заполнится, и начинает отбрасывать их, когда размер очереди достигнет максимального уровня. Если буфер будет постоянно заполнен, то сеть окажется перегружена. Повторная передача отброшенных пакетов требует дополнительных ресурсов (расход заряда батареи устройства и пропускную способность канала). В случае множественных коротких TCP сессий в сети наступает перегрузка, а также может произойти так называемая «Глобальная TCP-синхронизация». На основе вышесказанного можно сделать вывод, что Tail drop нерационально использует пространство памяти маршрутизатора.

Алгоритм RED же в зависимости от наполненности очереди отбрасывает пакеты, основываясь на статистической вероятности. Если буфер заполнен меньше определенного порога, то все пакеты пропускаются в обычном режиме. Когда очередь начинает заполняться, вероятность отбрасывания пакетов также повышается. Когда наполненность очереди превышает определенный

максимальный порог, вероятность отбрасывания пакета становится равной единице и все входящие пакеты отбрасываются.

Тем не менее, использование алгоритма RED не избавляет от необходимости повторной передачи пакетов, а лишь снижает нагрузку на сеть. Предполагаемое решение данной проблемы состоит в использовании метода сетевого кодирования, который скрывает факт потерянных пакетов от TCP. Такая связка позволит более эффективно использовать ресурсы сети и устройств, а также сведет к минимуму вероятность повторной передачи пакетов.

Описание алгоритма Tail Drop и RED

Стандартными системами, в которых используются алгоритмы AQM, являются системы обслуживания из ТМО (теории массового обслуживания) с очередью (queue, также называется «буфер»). Существует множество вариантов таких систем, однако одной из наиболее распространенных является система с одним конечным устройством и буфером ограниченного размера (Рисунок 1. Пример распространенной системы массового обслуживания.), где основными параметрами такой системы являются длина очереди K , интенсивность поступления данных λ и время технического обслуживания пакетов μ .

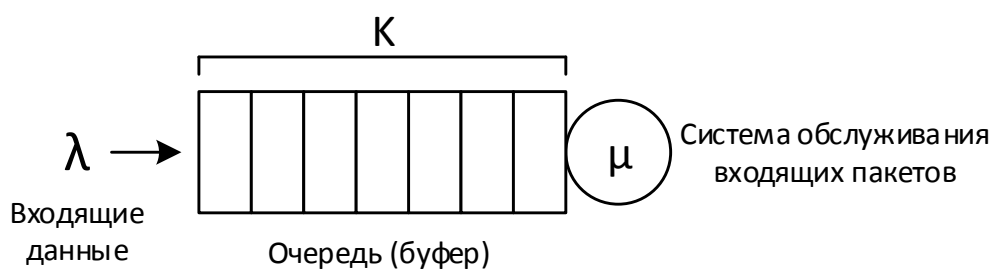


Рисунок 1. Пример распространенной системы массового обслуживания.

Tail drop является простейшим алгоритмом для управления переполнением очередей. При использовании этого алгоритма все входящие пакеты принимаются до тех пор, пока очередь не заполнится, после чего все входящие пакеты начинают отбрасываться. Когда в очереди освобождается место для нового пакета, алгоритм Tail drop перестает отбрасывать пакеты.

На Рисунок 2. Зависимость вероятности отброса пакета алгоритмом Tail Drop от загруженности очереди. представлена зависимость вероятности отброса пакета алгоритмом в зависимости от загруженности очереди.

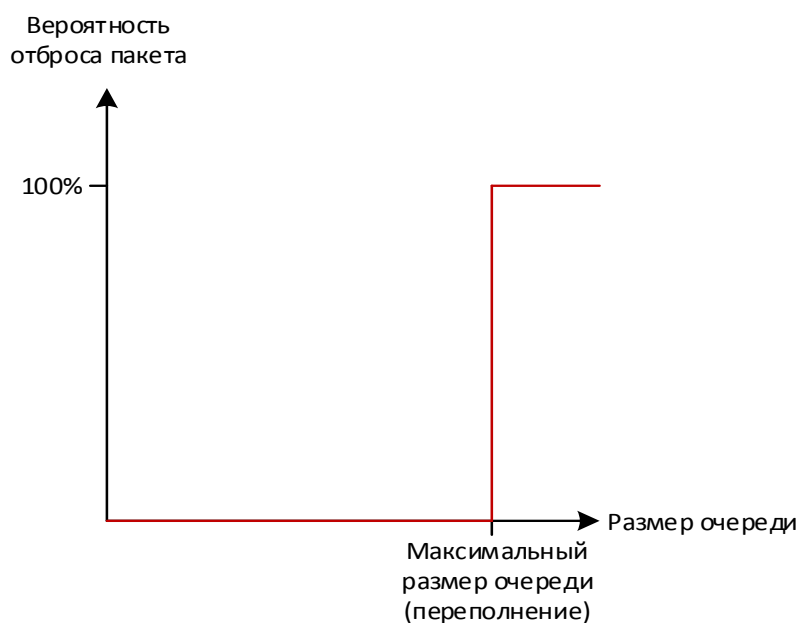


Рисунок 2. Зависимость вероятности отброса пакета алгоритмом Tail Drop от загруженности очереди.

Алгоритм Tail Drop имеет широкое применение в различных областях передачи данных, в том числе в протоколе TCP, но у данного алгоритма существует

ряд проблем [12]. Ниже приведены две основных проблемы, которые не позволяют эффективно использовать алгоритм для управления переполнением очередей Tail drop в сетях, подобных сетям MANET:

- 1) Алгоритм Tail drop не учитывает маркеры приоритетов, которые расставлены в заголовках пакетов, то есть важные для приема пакеты имеют такую же вероятность быть отброшенными, как и менее важные. Данное поведение деструктивно влияет на качество связи в протоколах, которые чувствительны к потерям, например, VoIP.
- 2) «Глобальная TCP-синхронизация» или «TCP global synchronization», которая происходит при одновременном подключении множества источников пакетов: происходит переполнение очереди, алгоритм Tail drop начинает работать и, соответственно, весь входящий трафик одновременно замедляется (такой процесс называют Slow Start), уменьшая величины окон на стороне отправителей. Из-за этого размер очереди и загрузка получателя уменьшается, что приводит к возвращению величин окон на отправителях на начальный уровень, что снова приводит к переполнению очереди, уменьшению окна и замедлению трафика. Такой цикл может продолжаться длительное время и крайне негативно влияет на общую производительность сети и интенсивность передачи данных в ней, так как часть времени система недостаточно эффективно загружена, а часть времени в системе производится отброс пакетов (Рисунок 3. График глобальной TCP-синхронизации., [13]).

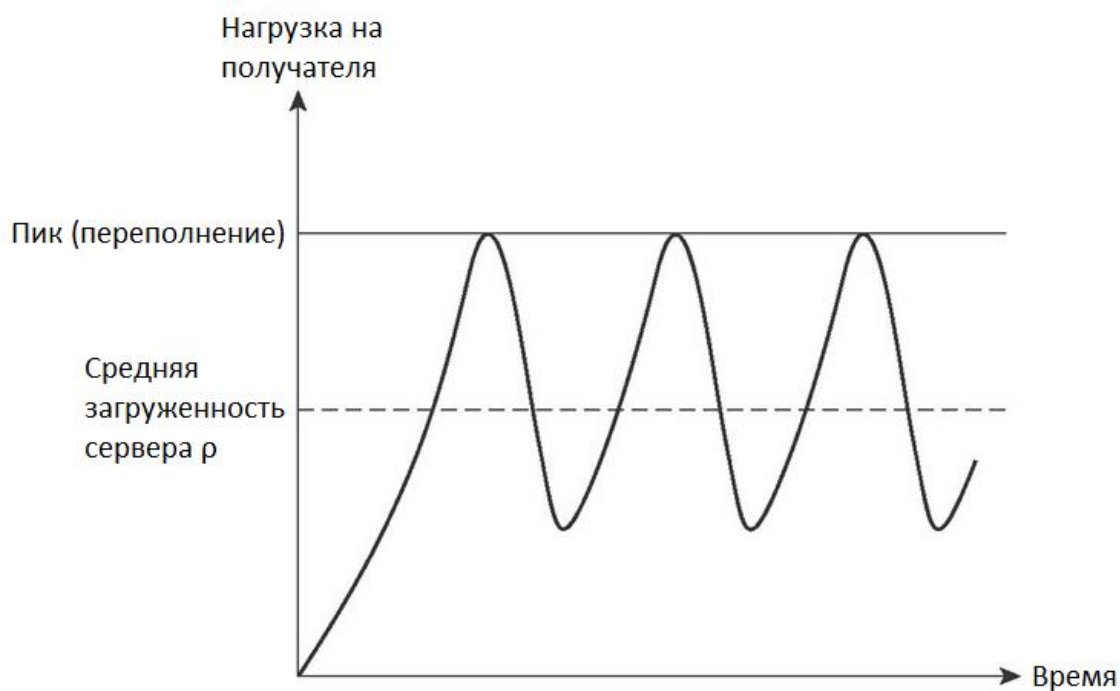


Рисунок 3. График глобальной TCP-синхронизации.

Алгоритм RED является модифицированным алгоритмом Tail Drop, который пытается максимально предотвратить полное отбрасывание пакетов, возникающее при использовании алгоритма Tail Drop. RED, основываясь на статистической вероятности, отбрасывает часть пакетов, приходящих в очередь до того, как она заполнится, при этом чем выше наполненность очереди, тем выше вероятность отбрасывания пакета. Алгоритм RED не делит пакеты по разным потокам, выбирая пакет для отбрасывания без учета того, насколько большой процент трафика занимает тот или иной поток. Тем не менее, ожидается, что пакеты, которые принадлежат к более «агрессивным» потокам (то есть занимающим большую часть потока), будут отбрасываться чаще пакетов менее агрессивных потоков.

Так как для каждого потока алгоритм отбрасывает разную долю пакетов, глобальной синхронизации не происходит. Из-за этого больших скачков нагрузки на получателя не происходит и средний размер очереди уменьшается. На Рисунок 4. Нагрузка на получателя с использованием алгоритма RED. [13] можно увидеть, что разница между пиковой и минимальной нагрузкой на получателя меньше, из чего следует более высокая средняя загруженность сервера.

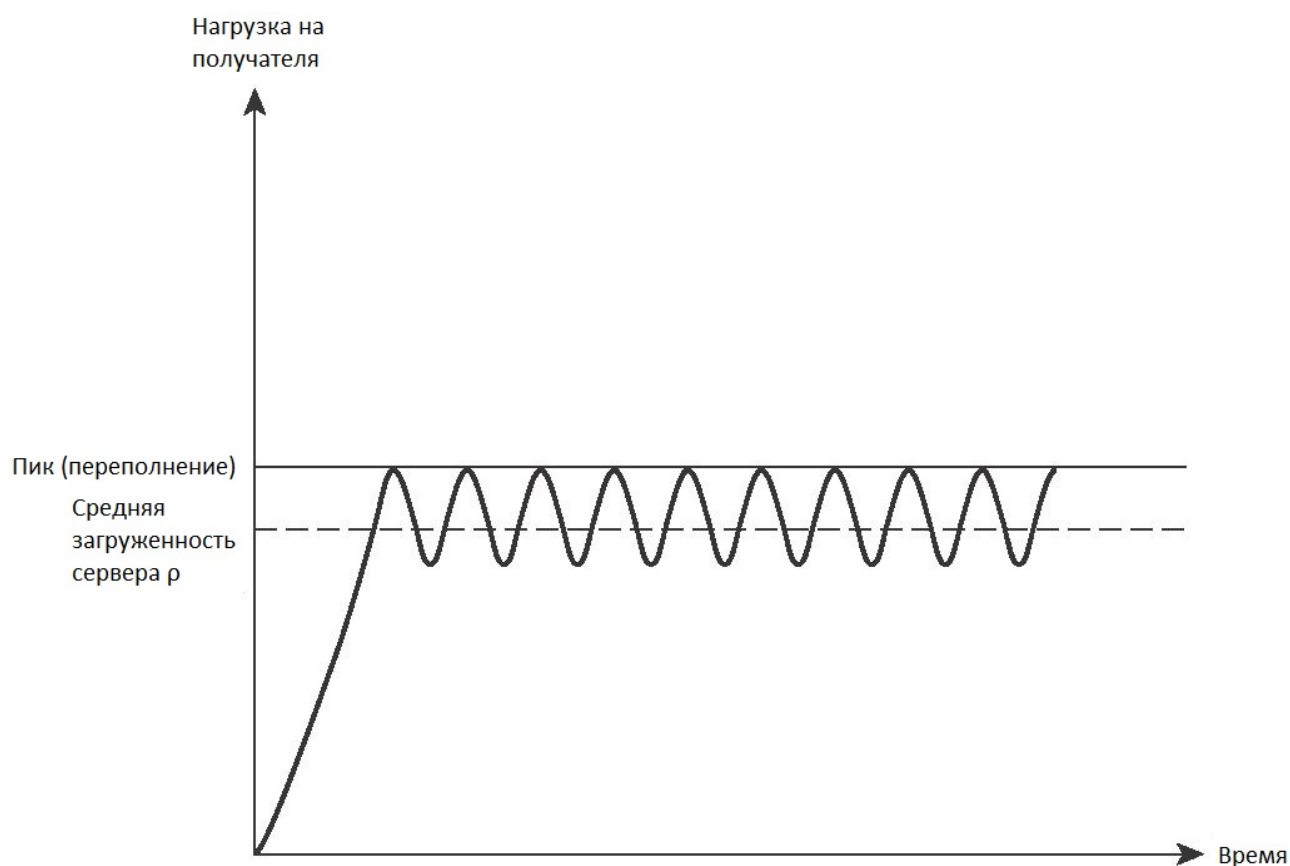


Рисунок 4. Нагрузка на получателя с использованием алгоритма RED.

Алгоритм RED имеет три параметра, влияющих на среднюю вероятность отбрасывания пакета:

1) Минимальный (нижний) порог – минимальная наполненность очереди, после которой алгоритм начинает отбрасывать пакеты. До достижения этой границы отбрасывания пакетов не происходит.

2) Максимальный (верхний) порог – максимальная наполненность очереди, после которой алгоритм будет отбрасывать только часть пакетов (в соответствии с параметром в пункте 3). Превышение этого параметра вызовет переход в режим Tail drop, когда производится отбрасывание всех входящие пакеты.

3) Вероятностный множитель (MPD, Mark Probability Denominator) – величина Z , при которой 1 из Z пакетов будут отброшены до тех пор, пока длина очереди находится между нижним и верхним порогами.

Пример работы данного алгоритма можно видеть на Рисунок 5. Вероятность отброса пакета алгоритмом RED в зависимости от длины очереди.

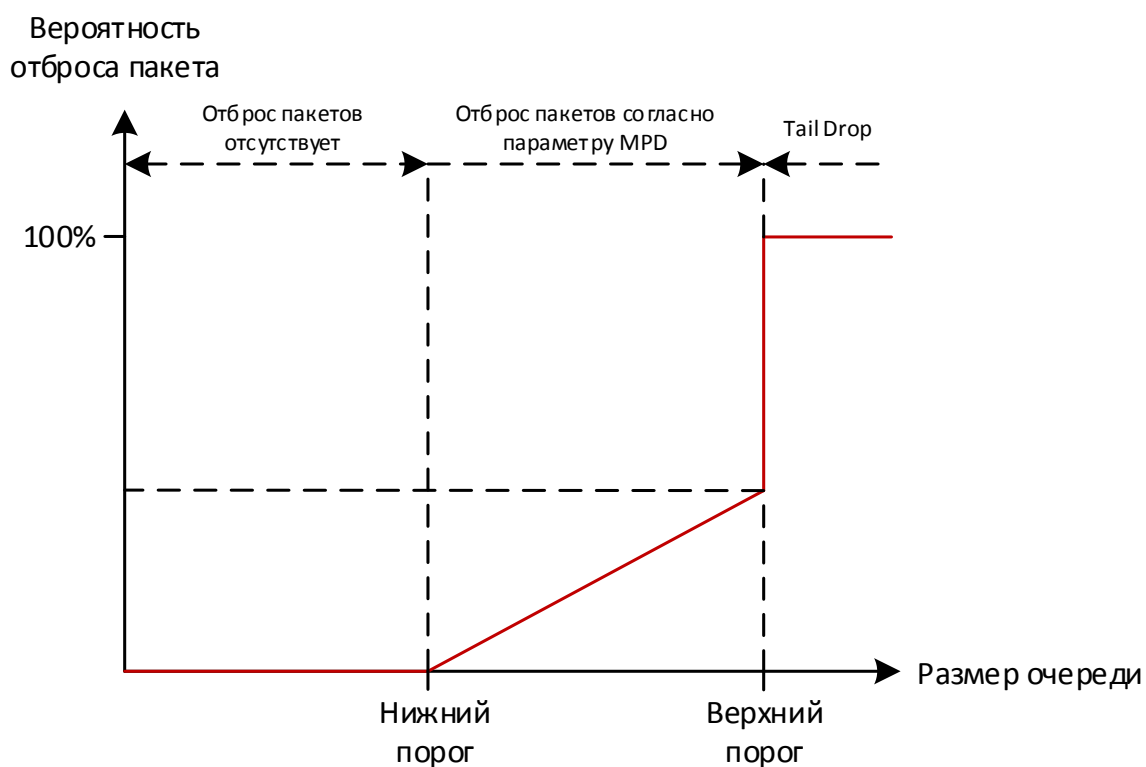


Рисунок 5. Вероятность отброса пакета алгоритмом RED в зависимости от длины очереди.

Также имеет смысл рассмотреть эффективность алгоритмов WRED и CBWRED, которые являются модификациями RED и также применяются в современных системах. В дальнейшем в работе будет рассматриваться потенциальная эффективность совмещения метода сетевого кодирования и алгоритма RED, при этом под алгоритмом RED будут также пониматься его модификации WRED и CBWRED.

Взаимодействие метода сетевого кодирования и алгоритма управления очередью RED

Метод сетевого кодирования (NC) основывается на формировании системы линейно независимых уравнений [14], где в роли неизвестных переменных выступают потерянные сетевые пакеты. Производится подсчет сумм по каждой строке, после чего становится возможным, в случае потери, однозначно восстановить утраченную информацию [15, 16]. Более подробно алгоритм описан в [8]. Метод сетевого кодирования доказал свою эффективность при его моделировании [18]. При взаимодействии с алгоритмом RED метод сетевого кодирования будет поддерживать заполненность буфера в оптимальном состоянии рис. 6, 7.

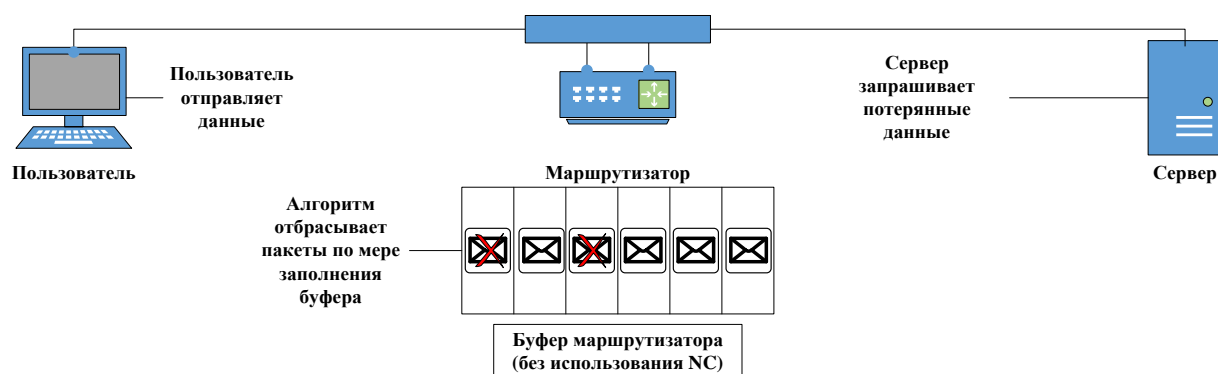


Рисунок 6. Схема передачи информации без использования сетевого кодирования.

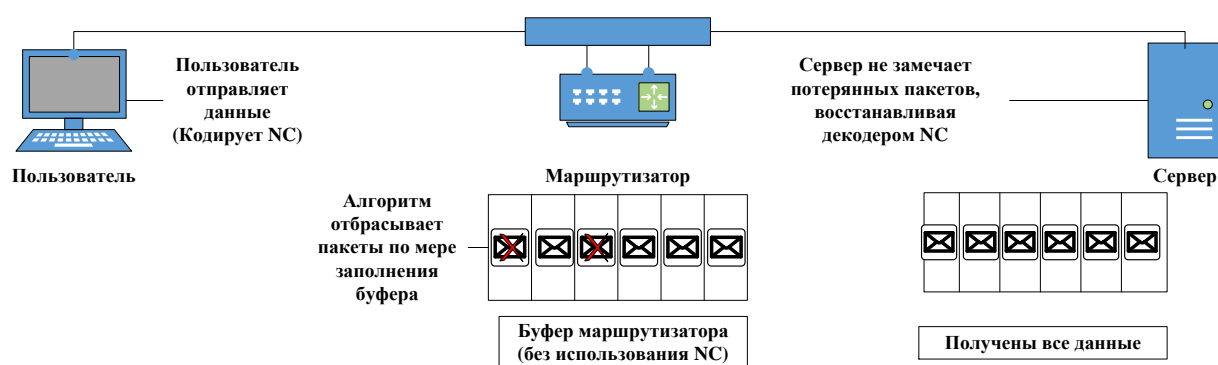


Рисунок 7. Схема передачи информации с использованием сетевого кодирования.

Таким образом не будет возникать состояния перегруженности сети. Характеристики информационного взаимодействия будут оставаться в нормальном состоянии или улучшаться. Для моделирования данного процесса предлагается использовать среду MATLAB, задействовать алгоритм кодирования и декодирования сетевого кода и симитировать работу буфера при использовании алгоритма управления очередью RED [18, 19]. Планируется получить результаты и построить зависимости воздействия кодовой последовательности пакетов на перегрузки в телекоммуникационных системах и сетях в условиях ограниченной пропускной способности и времени обработки пакета.

Заключение

Интегрирование метода сетевого кодирования вместе с алгоритмом RED представляет интерес для будущих исследований. Потенциальные возможности такой интеграции могут качественно сказаться на информационном взаимодействии множества беспроводных устройств. Дальнейшие разработки будут направлены на создание модели такого взаимодействия и апробации теоретических результатов [20]. В дальнейшем будут получены количественные результаты работоспособности этих алгоритмов.

Исследование выполнено в рамках выполнения научных исследований по теме № 0071-2019-0001 «Развитие теории и методов прикладной математики, нейросетевых технологий и систем управления процессами в задачах САД-систем, анализа визуальных данных, защиты информации и прогнозирования»

Библиографический список

1. Balakrishnan H., Seshan S., Katz R.H. Improving Reliable Transport and Handoff Performance in Cellular Wireless Networks, *Wireless Networks*, vol. 1, 1995, pp. 469 – 481.
2. Бородин В.В., Петраков А.М., Шевцов В.А. Анализ эффективности передачи данных в сети связи беспилотных летательных аппаратов // Труды МАИ. 2015. № 81. URL: <http://trudymai.ru/published.php?ID=57894>

3. Романов А.М., Гринголи Ф., Сикора А. Беспроводная синхронизация бортовых вычислительных устройств при помощи WiFi // Труды МАИ. 2019. № 108.
URL: <http://trudymai.ru/published.php?ID=109522>
4. Sundararajan J.K., Shah D., Medard M. et al. Network Coding Meets TCP // Journal of INFOCOM, 2009, pp. 280 - 288.
5. Seungwan R., Christopher R., Chunming Q. Advances in Active Queue Management (AQM) based TCP congestion control // Telecommunication Systems, 2004, vol. 25, no. 3 - 4, pp. 317 - 351.
6. Карпухин Е.О., Мешавкин К.В. Исследование влияния характеристик сети на выбор режима восстановления потерянных пакетов // Вестник компьютерных и информационных технологий. 2017. № 12. С. 39 – 46.
7. Карпухин Е.О., Бритвин Н.В., Мешавкин К.В. Исследование эффективности применения перспективных корректирующих кодов в гибридной ARQ/FEC системе на прикладном уровне // Новые информационные технологии в автоматизированных системах. 2017. № 20. С. 181 - 185.
8. Карпухин Е.О., Бритвин Н.В. Разработка метода гибридного управления доставкой сообщений на основе линейного сетевого кода // Электросвязь. 2017. № 10. С. 30 - 36.
9. Corson S., Macker J. Mobile Ad hoc Networking (MANET), Routing Protocol Performance Issues and Evaluation Considerations (RFC 2501), 1999, available at: <https://tools.ietf.org/html/rfc2501>

10. Волков А.С., Муратчаев С.С., Кульпина Ю.А. Разработка имитационной модели двухранговой сети MANET // Труды МАИ. 2019. № 109. URL: <http://trudymai.ru/published.php?ID=111387>
11. Aamir M., Mustafa A.Z. A Buffer Management Scheme for Packet Queues in MANET // Tsinghua Science and Technology, 2013, vol. 18, no. 6, pp. 543 - 553.
12. Gong Y. The quest for low-latency at both network edges: design, analysis, simulation and experiments. Department of Network and Computer Science, Télécom ParisTech, 2016, available at: <https://pastel.archives-ouvertes.fr/tel-01595947>
13. Ranjbar A.S. CCNP ONT Official Exam Certification Guide. Indianapolis, Cisco Press, 2007, available at: <http://www.ciscopress.com/store/ccnp-ont-official-exam-certification-guide-9781587201769>
14. Пантелеев А.В., Лунева С.Ю. Численный метод решения полностью нечетких систем линейных уравнений // Труды МАИ. 2019. № 109. URL: <http://trudymai.ru/published.php?ID=111433>
15. Вержбицкий В.М. Численные методы. Линейная алгебра и нелинейные уравнения - М.: Изд-во ОНИКС 21 век, 2005. - 430 с.
16. Бритвин Н.В. Проблемы разработки кодера и декодера в методе «Линейного сетевого кода» // 18-я международная конференция «Авиация и космонавтика-2019»: тезисы докладов (Москва, 18-22 ноября 2019). – М.: Логотип, 2019. С 116 – 117.
17. Джон Эриксон. Хакинг: искусство эксплойта. - СПб.: Символ-Плюс, 2010. - 512 с.

18. Madipelli S., Gillella D., Devaraya S. The RED Algorithm – Averaged Queue Weight Modeling for Non Linear Traffic, Blekinge Institute of Technology, 2009. URL: <http://web.archive.org/web/20170329090005/https://www.diva-portal.org/smash/get/diva2:831714/FULLTEXT01.pdf>
19. Rastogi S., Zaheer H. Comparative analysis of queuing mechanisms: droptail, red and nlred // Social network analysis and mining, 2016, vol. 6, no. 1, available at: <https://link.springer.com/article/10.1007%2Fs13278-016-0382-5>
20. Симонов А.С., Семенов А.С., Макагон Д.В. Направления развития высокоскоростной коммуникационной сети для многопроцессорных вычислительных систем // Труды МАИ. 2019. № 108. URL: <http://trudymai.ru/published.php?ID=109525>