

Сравнительный анализ методов управления сетевыми ресурсами в сетях SDN

Ю.С. Дмитриева¹

¹Санкт-Петербургский государственный университет телекоммуникаций им. проф. М.А. Бонч-Бруевича, Санкт-Петербург, 193232, Российская Федерация

*Адрес для переписки: dmitrieva_spbgut@bk.ru

Информация о статье

Поступила в редакцию 08.02.2022

Поступила после рецензирования 02.03.2022

Принята к публикации 05.03.2022

Ссылка для цитирования: Дмитриева Ю.С. Сравнительный анализ методов управления сетевыми ресурсами в сетях SDN // Труды учебных заведений связи. 2022. Т. 8. № 1. С. 73–83. DOI:10.31854/1813-324X-2022-8-1-73-83

Аннотация: Статья посвящена анализу методов управления сетевыми ресурсами в сетях SDN. В интересах этого выполнен обзор научных работ российских и зарубежных авторов для выявления применяемых в них решений, а также их достоинств и недостатков. Методы, освещенные в научных статьях, систематизированы по следующим критериям: год публикации, уровень применимости метода (физический, логический, сетевой), наличие математической модели, режим балансировки нагрузки относительно времени инцидента, решаемый круг задач, степень исследования (теоретическое, экспериментальное, практическое). На основании анализа результатов обзора сделан вывод о неудовлетворительности решений, а также о необходимости рассмотрения новой технологии IBN, охватывающей широкую область применения и работающей превентивно на всех уровнях SDN.

Ключевые слова: SDN, QoS, OpenFlow, IBN, балансировка нагрузки, сравнительный анализ.

Введение

В условиях развития информационно-коммуникационных технологий [1] требуется передача большого объема гетерогенной информации; как результат, растет и нагрузка на сеть [2]. Классическая маршрутизация не справляется с ростом расходования трафика со стороны потребителей информационных услуг. В интересах этого была разработана специальная концепция передачи данных – программно-конфигурируемая сеть (SDN, аббр. от англ. Software-Defined Networking), которая открыла новую эру для разработки стратегий, стандартов и архитектур управления сетевыми ресурсами. Однако на текущий момент и она оказалась недостаточно эффективной, о чем свидетельствует, например, уязвимость сети из-за наличия в ней контроллера SDN – единой точки отказа, при выходе из строя которого происходит потеря работоспособности всей сети [3]. Как результат, совершенствование SDN-решений остается актуальной задачей.

Одним из основных методологических подходов по совершенствованию любых решений является анализ их теоретических и практических реализа-

ций, выявление недостатков, определение достоинств. Как результат, будет возможен синтез новых решений, качественно отличных от существующих. Таким образом, главной причиной проведения данного исследования является необходимость определения слабостей существующих решений и поиска возможностей по созданию новых, более эффективных. Первым же шагом должен стать обзор научных публикаций российских и зарубежных ученых на предмет применяемых техник управления сетевыми ресурсами на базе SDN.

Далее будет произведен обзор статей, посвященных различным способам управления потоками в сетях SDN с систематизацией их характеристик, что позволит произвести категориальное сравнение методов на предмет присущих им слабостей. Затем, в случае невозможности модернизации существующих решений, можно будет сформулировать основные положения для собственного метода, не только лишенного недостатков предыдущих, но и обладающего их преимуществами. Целью же исследования является формирование представления о текущем состоянии дел в области SDN с позиции эффективности решения задач управления сетевыми ресурсами.

Обзор релевантных работ

Для получения представления о проработанности предметной области выполним статистическое исследование количества статей по критерию «балансировка нагрузки в сети SDN» в поисковой системе Google Scholar. Полученная закономерность распределения публикаций по годам представлена на рисунке 1.

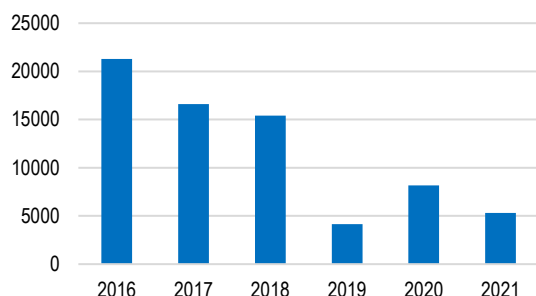


Рис. 1. Количество публикаций по годам (по данным поисковой системы Google Scholar)

Fig.1. Number of Publications by Year According to the Google Scholar Search Engine

Анализ хронологии публикаций статей (см. рисунок 1) позволяет сделать выводы, что распределение имеет условно неизменную тенденцию (поскольку в разные года количество статей имело как высокие, так и низкие значения). Это можно объяснить тем, что интерес к управлению сетевыми ресурсами в сетях SDN в части балансировки нагрузки является переменным. Можно предположить, что это связано с низкой оптимальностью передаваемых решений.

Произведем обзор релевантных работ, посвященных вопросам управления ресурсами в сетях SDN, на предмет используемых для этого авторами методов и алгоритмов балансировки нагрузки.

В работе [4] рассмотрен алгоритм, распределяющий различные типы ресурсов транспортной реконфигурируемой пакетной сети между поступающими на обслуживание потоками пакетов из расчетных процедур, при этом процесс информационного обмена на уровне IP-сети представлен с помощью математической модели системы массового обслуживания (СМО). Общий алгоритм управления ресурсами транспортной SDN состоит из двух этапов: 1) распределение виртуальных канальных ресурсов; 2) формирование требований к виртуальной топологии и распределение оптических ресурсов сети связи. Критерием оптимальности управления ресурсами оптических сетей выбрана минимизация суммарного потребления сетью электроэнергии.

А.С. Волков и соавт. в статье [5] представляют метод управления ресурсами SDN с учетом поочередности. Данный метод основан на математическом аппарате СМО и состоит из следующих этапов: перенаправление пакетов к контроллеру и их обработка, передача OpenFlow обработанных па-

кетов к коммутатору. Приводится модель процесса информационного обмена в виде системы уравнений и неравенств. При введении дополнительных ограничений по требованиям QoS (аббр. от англ. Quality of Service – качество обслуживания) возможно разбиение данной задачи на несколько подзадач верхнего уровня виртуальной топологии IP-сети: 1) распределения сетевыми и вычислительными ресурсами; 2) формирования требований к виртуальной топологии в виде задачи минимизации; 3) формирования виртуальной топологии на основе требований к количеству световых путей между парами коммутаторов и пропускными способностями световых путей.

Алгоритм механизма многоуровневой очереди (MQ, аббр. от англ. Multilevel Queue), обеспечивающий совместное использование ресурсов контроллера с несколькими уровнями очередей, которые могут динамически расширяться и агрегироваться в зависимости от загруженности сети, рассматривается М.Ю. Рытовым, Р.Ю. Калашниковым и А.А. Гореловым в [6]. Предлагаемый авторским коллективом подход оценивается путем сравнения его с базовым контроллером SDN; проведенное ими моделирование обосновывает необходимость его применения для улучшения использования пропускной способности канала при решении широкого круга задач.

Б.М. Данешманд в [7] представил результаты проведенного анализа двух методов QoS. Первый метод основан на протоколе OpenFlow, который объединяет две характеристики: уникальный контроллер и простые устройства, что приводит к проблемам надежности и масштабируемости. Также существует проблема несогласованности между контроллерами (проблема консенсуса), их размещения и протокола связи, планирования нескольких контроллеров и т. д. Второй основан на контроллерах с открытым исходным кодом SDN. Здесь приводится описание QoS трех контроллеров SDN (OpenDaylight, ONOS и Floodlight) и концепция трех общих услуг 5G с характеристиками производительности. Проблема второго метода – параметры программирования. В работе показаны факторы, удовлетворение которых способствует выполнению требований QoS: уменьшенная задержка, повышенная экономическая эффективность, сокращенное время создания сервиса, высокая скорость передачи данных, высокая возможность подключения, повышенная мобильность, снижение потребления энергии, сверхнадежность. Дополнительно проанализированы и параметры для улучшения QoS в сети следующего поколения: пропускная способность, задержка, джиттер и потери.

Задача виртуализации или абстрагирования транспортных ресурсов SDN с помощью разработки драйвера внутри контроллера для адаптации интерфейса к конкретным типам оборудования,

используя протокол NETCONF, решается в работе [8]. Рассмотрена транспортная SDN – подмножество архитектуры SDN, которая включает в себя расширения абстракций, интерфейсов, протоколов и элементов плоскости управления для преодоления особенностей транспортных сетей и ограничений, связанных с использованием OpenFlow.

В статье А.Д. Агеева и соавт. [9] предлагается алгоритм динамической организации и классификации потоков в мультисервисной SDN-сети. Алгоритм состоит из трех этапов: сбор статистических сведений; обработка результатов и классификация потоков; принятие решений по результатам второго этапа и реализация механизмов управления трафиком. Первый этап выполняется с применением протокола OpenFlow, на втором этапе используется приложение для выполнения математических операций расчета и методы Machine Learning и третий этап проводится с использованием существующих протоколов управления сетью.

Доклад Б. Ибрагимова и В. Керимова на Международной конференции [10] посвящен одному из подходов к решению задачи исследования и оценки ресурсов в мультисервисных сетях связи (МСС) будущего поколения FN (*аббр. от англ. Future Network*). Авторы предложили математическую модель для оценки вероятностно-временных характеристик МСС, использующих концепцию единого инфокоммуникационного пространства при оказании мультимедийных услуг. Особенности программно-аппаратных средств сетей SDN, платформы IMS (*аббр. от англ. IP Multimedia Subsystem*) и технологий NFV (*аббр. от англ. Network Functions Virtualization*), которые позволяют более точно учесть телекоммуникационные процессы, протекающие в исследуемой единой многооператорской инфраструктуре МСС будущего поколения FN, были описаны при помощи целевых функций. Оценка показателей математической модели и вероятностно-временных характеристик производилась на основе многолинейных СМО.

В.С. Елагин в [11] представил математическую модель динамического контентозависимого алгоритма балансировки нагрузки в сетях центров обработки данных (ЦОД) с модифицированной обратной связью, учитывающего тип запроса и динамическое состояние сервера. Сеть ЦОД представлена в виде открытой СМО. Для эффективной работы разработанного алгоритма необходимо обеспечить мониторинг состояния серверов с помощью адаптивного алгоритма мониторинга. В результате расчета обеспечивается динамическое распределение нагрузки по серверам в каждом цикле.

Авторский коллектив из СПбГУТ в [12] синтезировал модель сети с применением программного средства mininet и топологии на базе VND (*аббр. от*

англ. Visual Network Description) и проанализировал сетевой трафик при помощи Wireshark. В статье произведена оценка параметра Херста, для чего применяется метод анализа изменения дисперсии. По результатам проведенных экспериментов предлагается использовать поле в IP-заголовке пакетов, а также обоснована быстрая реакция SDN-контроллеров с высокой скоростью (от 100 мкс до 1 мс). Особенностью исследования является проведение тестирования на базе университетской сети Wi-Fi.

Создание SDN-балансировщика нагрузки для устройств защищенных сетей, его архитектура и программное обеспечение описаны в исследовании О.Ю. Гузева и И.В. Чижова [13]. Алгоритм балансировки нагрузки реализован на языке программирования Java в виде модуля для контроллера SDN с открытым исходным кодом (имеющим название Beacon). Авторы провели эксперименты по балансировке нагрузки на устройствах для защищенных сетей: L3-криптошлюз, Transport Layer Security (TLS) – криптошлюз, Intrusion Detection System (IDS). Во всех экспериментах применялся унифицированный алгоритм балансировки, использующий весовые коэффициенты каналов. Анализ трафика при помощи Wireshark после балансировки демонстрирует отсутствие «разорванных» TCP-сессий, что свидетельствует о корректном распределении потоков трафика.

Эти же авторы в работе [14] решают задачу балансировки нагрузки на криптографические маршрутизаторы, предназначенные для защиты каналов связи между центрами обработки данных, объединенными в единый «эластичный» ЦОД. Для решения поставленной задачи они применяют способ, основанный на технологии использования SDN. К достоинствам рассмотренного метода относятся следующие: упрощение настройки криптографического оборудования, отказоустойчивость связи при потере криптографического узла в одном ЦОД, независимость от используемого протокола криптографической защиты сетевого уровня.

В статье [15] разработана имитационная модель с требуемыми параметрами SLA (*аббр. от англ. Service Level Agreement*), рассчитаны новые индикаторы. При выходе показателей SLA за пределы допустимых значений выполняется расчет новых значений индикаторов SLA, для предотвращения изменения их значений реализуется перераспределение ресурсов сети. Применяется методология моделирования динамики дискретных систем, в основе которой лежит граф раскрашенных сетей Петри. Ликвидируется отрицательное влияние инцидентов и проблем на реализацию услуг. Корректировка характеристик сети осуществляется в соответствии с договором об уровне обслуживания SLA.

В работе В.Н. Пашкова [16] рассмотрен основной способ достижения отказоустойчивости – резервирование ресурсов в контуре управления, состоящего из коммутаторов и потоков данных. Автором предложен проактивный алгоритм распределения управления коммутаторами между контроллерами с учетом резервирования и алгоритм балансировки нагрузки между контроллерами и описаны механизмы обнаружения и восстановления управления РПУ (*аббр. от* Распределенная платформа управления) SDN. В случае одиночного отказа контроллера РПУ или перегрузки контроллера управление сетью восстанавливается за счет его перераспределения коммутаторами между оставшимися функционирующими контроллерами. Проактивный алгоритм выбора резервных контроллеров для каждого коммутатора позволяет минимизировать время восстановления в случае сбоя одного контроллера. Алгоритм балансировки нагрузки контроллеров РПУ предотвращает перегрузку контроллеров платформы. Эти алгоритмы являются частью сервиса восстановления для каждого РПУ SDN.

Решение проблемы масштабируемости сети при использовании одиночного контроллера приводится израильскими учеными Х. Суфиевым и Ю. Хаддадом в [17]. Предложено несколько контроллеров для управления сетью WAN (*аббр. от англ.* Wide-Area Network), но балансировка нагрузки между несколькими контроллерами становится проблематичной. Согласно авторской концепции, при обнаружении перегруженного кластера контроллер SC (*аббр. от англ.* SuperController) запускает алгоритм разделения контроллеров RCs (*аббр. от англ.* RegularControllers) на кластеры и обновляет векторы кластеров CV (*аббр. от англ.* ClusterVector) контроллеров RCs.

В работе Х. Суфиева и соавт. [18] описан подход к решению проблемы балансировки нагрузки в плоскости управления SDN. Многоуровневая архитектура плоскости управления, адаптированная к переменной нагрузке трафика, динамически обеспечивает ее балансировку. Основной компонент сети SDN – SC. Для оптимального распределения нагрузки между кластерами необходимо минимизировать нагрузку в наиболее загруженном кластере. Используется архитектура DCF (*аббр. от англ.* Dynamic Cluster Flow), которая разделяет контроллеры на кластеры. В интересах реакции на изменения условий сети авторы предложили состоящий из двух этапов алгоритм повторной кластеризации динамических контроллеров (*от англ.* Dynamic Controllers Clustering). На первом этапе решается проблема К-центра (*от англ.* K-Center problem). На втором, для минимизации разницы нагрузки между кластерами, на основе теории игр разрабатывается правило обмена контроллерами между кластерами. В ходе исследования выполнено моделирование сети с несколькими кластерами

и одним SC с использованием симулятора в среде Visual Studio .Net. Цель моделирования – показать, что алгоритм кластеризации динамических контроллеров обеспечивает лучшие результаты (по показателям: соответствие разностной границе и количеству замен), чем метод фиксированной кластеризации.

На Международной конференции ICUFN '2019 был представлен доклад [19], в котором Aly W.H.F. (исследователь из Американского университета Ближнего Востока в Кувейте) предложил алгоритм балансировки нагрузки с использованием контроллера адаптивной балансировки нагрузки (CALB, *аббр. от англ.* Controller Adaptive Load Balancing): выбора главного контроллера и метода миграции коммутаторов. CALB динамически адаптирует нагрузку на главном контроллере на основе текущей нагрузки на каждом подчиненном контроллере. Используется пул для назначения контроллеров на основе информации об их применении. Эта информация полезна для балансировки нагрузки и снижения энергопотребления путем переноса коммутаторов с перегруженных контроллеров на недогруженные. CALB учитывает задержку между коммутаторами и связанными с ними подчиненными контроллерами, чтобы минимизировать время отклика. CALB предполагает, что к нему подключены главный контроллер и n подчиненных контроллеров. Для моделирования используется эмулятор mininet, контроллер Floodlight, коммутатор OpenFlow.

В статье этого же автора [20] обосновывается применение модели адаптивной балансировки нагрузки универсального контроллера (GCALB, *аббр. от англ.* Generic Controller Adaptive Load Balancing) для SDN; в этом случае GCALB адаптирует нагрузку между подчиненными контроллерами. GCALB сравнивается с двумя эталонными алгоритмами: HyperFlow и повышенной отказоустойчивости контроллера (ECFT, *аббр. от англ.* Enhanced Controller Fault Tolerant).

Первый заключается в том, что подчиненные контроллеры контролируют работоспособность главного. При его отключении ведомый контроллер выбирает ему замену из дополнительного списка, который составляется на основе текущего состояния, измеряемого на каждом ведомом контроллере, и который определяется оставшейся емкостью и скоростью потери данных.

Второй заключается в поддержании механизма отказоустойчивости. Были изучены две архитектуры: один ведущий контроллер с несколькими подчиненными контроллерами и несколько подчиненных контроллеров. Для каждого этапа моделирования экспериментов используется эмулятор mininet. Показатели пропускной способности и время отклика используются для измерения производительности.

Для обнаружения перегрузок, балансировки нагрузки и обеспечения качества обслуживания в статье ученых и специалистов из Мадрасского технологического института (Индия) [21] предлагается идея разделения фреймворка на модули. Введены параметры, определяющие качество сети: 1) пропускная способность; 2) задержки и надежность; 3) эффективная маршрутизация между узлами и каналами связи; 4) обработка различных требований QoS, не игнорируя при этом трафик пользователей с приоритетом. Для гарантии пропускной способности используется алгоритм с несколькими очередями, при котором очереди создаются для каждого интерфейса коммутаторов в сети. В целях обеспечения гарантий задержки используется алгоритм измерения задержки, основанный на аппарате СМО, и предложена процедура поиска подходящего пути с ограниченной задержкой. Чтобы гарантировать надежность обслуживания, требуется создать механизмы восстановления после сбоев. В этой работе предложен упреждающий подход, использующий группу быстрого переключения коммутаторов. Для балансировки нагрузки применяется улучшенный алгоритм с условием непересекающихся узлов. При одновременной передаче трафика по нескольким маршрутам количество потерянных пакетов уменьшается на 98,8 %.

Индийские ученые Б.Р. Килли, С.В. Рао в своем исследовании [22] применяют алгоритмическую модель размещения контроллера с балансировкой нагрузки (LBSP, *аббр. от англ.* Load Balanced Controller Placement), основная идея которого состоит в том, чтобы разделить глобальную сеть (WAN) на несколько доменов и выбрать контроллер для каждого домена, при выполнении условия, что назначенное каждому контроллеру количество коммутаторов должно быть больше минимального значения. Цель LBSP – свести к минимуму максимальный дисбаланс нагрузки. Алгоритм тестировался на сетях Chinanet и Interoute. Результаты показали, что LBSP-алгоритм превосходит существующие решения по размещению контроллеров для WAN по параметрам дисбаланса нагрузки и задержки между контроллерами без влияния на отклонения и среднюю задержку, оказываемую коммутатором на контроллер.

В [23] предложена основанная на OpenFlow стратегия балансировки нагрузки для ЦОД, которая определяет потоки т. н. «Elephant» (сеансы передачи данных, занимающие значительные объемы пропускной способности сети) и реализует схемы маршрутизации с различными типами потоков для достижения высокой пропускной способности сети на основе графов. Предложенный китайскими исследователями Х. Ши, И. Ли, Г. Хе, Т. Янг, Л. Жанг, П. Лю, Г. Жанг и З. Лианг способ балансировки нагрузки сравнивается со стратегией равноценной многопутевой маршрутизации

ECMP (*аббр. от англ.* Equal-Cost Multi-Path). Моделирование, проведенное на испытательном стенде mininet с контроллером POX, показало, что можно эффективно уменьшить перегрузку сети и улучшить ее использование по сравнению с ECMP.

Алгоритм балансировки нагрузки Nayan Seth сети SDN, увеличивающий пропускную способность и скорость передачи, предотвращающий потери пакетов и задержку при выходе из строя коммутатора, описан в статье исследователей из Технологического университета KLE В. Элигара и соавт. [24]. Его тестирование было реализовано на испытательном стенде под Linux mininet, а в качестве контроллера SDN была применена платформа OpenDaylight (ODL) с открытым исходным кодом. Этот алгоритм написан на языке программирования Python. Wireshark и iPerf использовались для тестирования производительности сети до и после его запуска. Применение алгоритма в небольших сетях улучшило показатели сети, а в больших сетях – увеличило потери пакетов и дрожание.

Доклад ученых университета индийского штата Пенджаб С. Каура, К. Кумара, Дж. Сингха, Н.С. Гуммана на международной конференции по информатике [25] посвящен стратегии циклической балансировки нагрузки «Round-Robin» и ее сравнению с уже реализованной случайной стратегией с использованием коммутатора Open-Flow, подключенного к контроллеру POX. Сделан вывод о необходимости распределения потоков между серверами и обеспечивать использование всех серверов при условии одинакового объема клиентских запросов.

Их соотечественники из университета Тапара в своей статье [26] предлагают схему оптимизации нагрузки и обнаружения аномалий (LOADS, *аббр. от англ.* Load Optimization and Anomaly Detection Scheme). В LOADS вероятность выбора коммутатора определяется использованием двух параметров: расстояние от коммутатора до контроллера и отношение потребления ресурсов коммутатора к его контроллеру. Целевая функция алгоритма оптимизации нагрузки и обнаружения аномалий состоит в минимизации времени отклика всех контроллеров на плоскости управления. В LOADS, развернутой на уровне данных для классификации вредоносного и нормального трафика, разработан модуль обнаружения сетевых аномалий на основе потока IP. Чтобы устранить сетевую аномалию, LOADS использует политики контроля доступа поведения пользователя в сети. Предложенная схема оценивается на эмуляторе mininet с контроллером POX.

Стратегия балансировки нагрузки на основе нечеткой логики (LBSFL, *аббр. от англ.* Load Balancing Strategy Based On Fuzzy Logic), в которой сначала анализируется нечеткая функция принадлежности, влияющая на параметры производительности

нагрузки сервера, изложена в докладе китайских ученых Г. Ли, Т. Гао, З. Жанг и И. Чен [27]. Состояние нагрузки виртуального сервера оценивается с помощью нечеткой логики. Затем возможности централизованного управления контроллерами SDN для всей сети используются с целью мониторинга информации виртуального сервера в режиме реального времени и планирования его задач. При низкой общей нагрузке сервер с наименьшей нагрузкой переводится в спящий режим; при высокой общей нагрузке – перезапускается для экономии энергии или повышения производительности. Наконец, реализуется динамический баланс между общей нагрузкой, производительностью и энергопотреблением. Имитационные эксперименты в программном обеспечении с открытым исходным кодом mininet 2.0 показали, что предложенная стратегия повышает общую производительность сети, особенно при решении задач с интенсивной связью и использовании сети с высокой задержкой.

Механизм балансировки веб-нагрузки на основе статистики сервера (SD-WLB, *аббр. от англ. An SDN-aided mechanism for Web Load Balancing based on server statistics*), распределяющий веб-запросы на каждый сервер в соответствии с его временем отклика и объемом трафика соответствующего порта коммутатора, представлен в статье иранских ученых К. Солеиманзадех, М. Ахмади и М. Нассири [28]. Централизованный контроллер SDN периодически собирает эту информацию, чтобы поддерживать актуальное представление о распределении нагрузки между серверами, и входящие запросы пользователей перенаправляются на наиболее подходящий сервер. Моделирование производится с помощью эмулятора mininet.

Исследователи из Турции Б. Бабаигит и Б. Улу предлагают решение задачи балансировки нагрузки между серверами в сетях ЦОД с помощью методов машинного обучения (ML, *аббр. от англ. Machine Learning*) [29]. Формой ML является глубокое обучение (DL, *аббр. от англ. Deep Learning*), которое позволяет прогнозировать, классифицировать и принимать решения, обрабатывая большие объемы данных. Для обучения сети DL используются переменные значения нагрузки между связями. Время отклика для балансировки нагрузки сравнивается с временем отклика различных алгоритмов ML, таких как искусственная нейронная сеть, машина опорных векторов и логистическая регрессия.

Китайские исследователи Г. Жонг, И. Фанг и Й. Цуй в статье [30] рассмотрели эффективную схему балансировки нагрузки, основанную на времени отклика сервера (LBBSRT, *аббр. от англ. Load Balancing Based on Server Response Time*), за счет гибкости SDN. Используя время отклика для выбора сервера с минимальным или наиболее стабильным его значением, получают равномерно сбалансированные нагрузки на сервер.

Авторский коллектив из Аньхойского университета (Китай) описал в своей работе [31] изменение последствия среднего времени отклика клиентских запросов, т. е. времени зондирования (время исследования «серверной фермы» для получения доступа к данным сервера) с использованием концепции теории управления замкнутым контуром (LBBCLCT, *аббр. от англ. Load Balancing Based on Closed Loop Control Theory*). В LBBCLCT эффективность распределения запросов пользователей на наиболее подходящий сервер своевременно меняет сервер, даже когда трафик не согласован.

Индийские ученые Г. Баббар, С. Партгибан, Г. Радгакришнан и С. Раны в [32] синтезировали генетический алгоритм балансировки нагрузки (GLBA, *аббр. от англ. Genetic Load Balancing Algorithm*) в мультимедийных приложениях, реализованный на контроллере POX с помощью эмулятора mininet на языке Python. Алгоритм GLBA используется для расчета эффективности и результативности.

Стратегия надежного развертывания мультиконтроллеров с учетом баланса нагрузки (RLMD, *аббр. от англ. Reliable and Load balance-aware Multi-controller Deployment*) описана в [33]. Специалисты их Китайского национального центра цифровой коммутации разработали модель сети с несколькими контроллерами и определили соответствующие параметры для RLMD. После чего для реализации этой стратегии ими были написаны соответствующие алгоритмы. Алгоритм выбора размещения контроллера (CPS, *аббр. от англ. Controller Placement Selection*) для изучения надежных развертываний контроллеров взвешивает эффективность узлов и качество пути. С точки зрения достижения балансировки нагрузки между контроллерами был улучшен подход кластеризации K-center и разработан алгоритм множественного разделения домена (MDP, *аббр. от англ. Multiple Domain Partition*) для выбора коммутаторов в соответствии с эффективностью узла и скоростью балансировки нагрузки контроллера, что может реализовать разумное планирование домена. Эксперимент развернут с помощью эмулятора mininet.

Коллектив авторов из Бирмингемского городского университета предложил улучшенный алгоритм принятия решений о миграции коммутатора (ISMDA, *аббр. от англ. an Improved Switch Migration Decision Algorithm*), решающий сетевую проблему, когда входящая нагрузка является потоком данных [34]. В этом исследовании использован модуль балансировки платформы миграции коммутаторов, который иницируется на этапе дисбаланса нагрузки контроллера и выполняется на каждом контроллере. В алгоритме ISMDA вычисляется дисперсия и среднее состояние нагрузки контроллера для определения набора недогруженных контроллеров в сети. Построенная модель миграции использовалась одновременно как для определения

стоимости миграции, так и для оптимального выбора контроллера среди множества недогруженных контроллеров при вариации балансировки нагрузки. Сравнительный анализ производительности был реализован на базе таких показателей, как пропускная способность контроллера, время отклика, количество пространства для миграции и потери пакетов.

Произведем аналитический обзор работ по критериям, полученным из следующих соображений. Во-первых, методы должны быть сравнены по уровню их реализации в SDN, а также на теоретической базе (или наличию моделей). Во-вторых, все методы могут действовать с различной скоростью – не только борясь с последствиями нарушений в сетевой передаче, но и действуя превентивно. В-третьих, часть методов может иметь лишь частное применение, что снизит их использование для всех необходимых сценариев и областей. И, в-четвертых, методы могут считаться лишь теоретически существующими, поскольку в рассмотренных исследованиях не упоминаются какие-либо реальные эксперименты.

Таким образом, в качестве критериев выберем следующие:

К_1 – Уровень применимости методов со значениями согласно уровням SDN: физический (Ф), логический (Л), сетевой (С);

К_2 – Наличие математической модели со значениями: тип модели (классическая или авторская) или указание на ее отсутствие (–);

К_3 – Проактивный режим балансировки в виде времени реагирования: до момента инцидента (t_-), во время его наступления (t_0) и после него (t_+);

К_4 – Применимость, как возможность решения большого круга задач, со значениями: широкая (Ш), средняя (С) и узкая (У);

К_5 – Степень исследования со значениями: теория (Т), эксперимент (Э) и практическое решение (П).

Результаты сравнительного анализа представлены в таблице 1, где для К_2 введены следующие обозначения моделей: СМО* – оригинальная авторская модель на базе СМО, CPN (аббр. от англ. Coloured Petri Net) – модель графа раскрашенной сети Петри, К-СР – модель K-Center Problem, ТИ – модели из теории игр, ГМ* – оригинальная авторская модель на основе теории графов, RR – Round-robin.

Результаты сравнительного табличного анализа позволяют сделать следующие выводы.

Во-первых, согласно значению по К_1 при общем количестве исследований (31 работа) уровень применимости методов в основном сетевой (14 работ), меньше физический (11 работ); и лишь малая часть методов работает на логическом уровне (6 работ). Для наглядности вывода, гистограмма применимости методов представлена на рисунке 2 (цвет – желтый).

ТАБЛИЦА 1. Результаты сравнительного анализа методов балансировки трафика

TABLE 1. The Results of Traffic Balancing Methods Comparative Analysis

Ссылка	Год	К_1	К_2	К_3	К_4	К_5
[4]	2020	С	СМО	t_+	Ш	Э
[5]	2021	Ф	СМО*	t_+	Ш	Э
[6]	2021	Ф	MQ	t_+	Ш	Э
[7]	2021	Ф	–	t_0	Ш	Т
[8]	2019	Ф	–	t_-	Ш	Э
[9]	2017	Ф	–	t_+	Ш	Т
[10]	2018	С	СМО	t_+	Ш	Э
[11]	2017	С	СМО	t_0	Ш	Э
[12]	2017	С	–	t_-	Ш	Э
[13]	2018	Ф	–	t_+	У	П
[14]	2018	С	–	t_-	Ш	Э
[15]	2019	С	CPN	t_+	С	Э
[16]	2019	С	Автор.	t_+	У	Э
[17]	2016	Л	–	t_0	Ш	Т
[18]	2019	Л	К-СР, ТИ	t_+	Ш	Э
[19]	2019	Ф	–	t_0	С	Э
[20]	2019	С	GCALB	t_0	Ш	Э
[21]	2021	С	СМО	t_-	С	Э
[22]	2020	Л	LBCP	t_+	Ш	Э
[23]	2020	С	ГМ*	t_0	Ш	Э
[24]	2020	Ф	–	t_0	С	Э
[25]	2015	С	RR	t_+	Ш	Т
[26]	2019	Ф	LOADS	t_+	Ш	Э
[27]	2018	Ф	LBSFL	t_0	Ш	Э
[28]	2019	Ф	SD-WLB	t_+	Ш	Э
[29]	2021	Л	DL	t_-	Ш	Т
[30]	2017	С	LBBSRT	t_0	Ш	Т
[31]	2022	С	LBCLCT	t_0	Ш	Э
[32]	2022	С	GLBA	t_+	У	Э
[33]	2018	Л	RLMD	t_0	Ш	Э
[34]	2020	Л	ISMDA	t_0	Ш	Э

Во-вторых, наиболее часто применяется математическая модель по критерию К_2, основанная на СМО – четыре работы, и отсутствует какая-либо математическая модель в девяти работах; во всех остальных рассмотренных работах авторы используют различные модели, включая оригинальные.

В-третьих, согласно значению по критерию К_3, наиболее часто методы применяются уже после возникновения (14 работ) или во время инцидента с перегрузкой сети (12 работ); реже балансировка нагрузки выполняется до момента наступления (5 работ). Распределение также представлено на рисунке 2 (цвет гистограммы – зеленый).

В-четвертых, по критерию K_4 не все методы могут применяться для большого круга задач. Наиболее актуальные из них охватывают узкий круг задач (23 работы) и средний круг (4 работы). Большая же часть методов используется для решения широкого круга задач – 15 работ (см. рисунок 2, цвет гистограммы – красный).

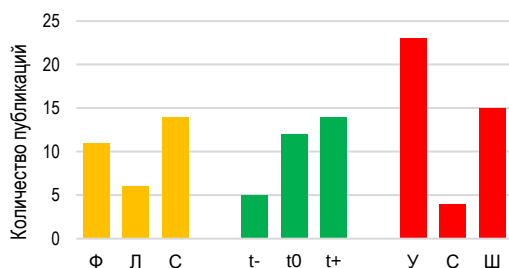


Рис. 2. Гистограмма значений критериев сравнительной оценки релевантных работ

Fig. 2. Histogram of criterion values for comparative evaluation of the corresponding works

И, в-пятых, большинство решений доведены до эксперимента по критерию K_5 (24 работы), но только один способ управления сетевыми ресурсами – до практической реализации, а в 6 работах ограничились лишь теоретическим исследованием. Таким образом, можно предположить, что практически все исследования не могут считаться полноценными, поскольку полученные результаты не были апробированы в рабочих условиях.

Исходя из сделанных выводов, можно говорить не только о недостаточности существующих решений, но и о практической невозможности их требующей модернизации. Таким образом, предлагается рассмотреть и провести исследования сети на основе намерений, обозначающего в зарубежной литературе, как IBN (аббр. от англ. Intent-Based Networks). В IBN реализован подход к автоматизированному управлению в соответствии с сетевыми

политиками. Возможна балансировка до момента наступления инцидента: способность рассчитывать поведение приложений и пользователей, обучаться и перенастраиваться в процессе эксплуатации.

В технологию IBN заложен механизм непрерывного отслеживания с автоматической перенастройкой для согласованности с заданными намерениями, что обеспечивает дополнительный уровень мониторинга сети. Основой этого механизма является система обратной связи, охватывающая автоматизацию на основе политик, сетевой аналитики и машинного обучения. Дополнительная информация, извлекаемая посредством контекстного анализа, помогает ускорить решение проблем в SDN и улучшить планирование ресурсов сети.

Выводы

В исследовании был проведен обзор научных работ на предмет различных методов по балансировке сетей SDN. Как результат, произведено сравнение предлагаемых решений по пяти критериям: уровень применимости, математическая модель, проактивность, круг применения и степень исследования. Исходя из сравнительного анализа работ, а также учитывая достоинства и недостатки описанных в них методов, сделано предположение, что качественное новое решение может быть найдено в сети на основе намерений, работающей на сетевом, логическом и физическом уровнях сети SDN до наступления инцидента. В такой сети можно непрерывно проверять соответствие между фактическим и ожидаемым состояниями сети с помощью мониторинга ключевых показателей эффективности сети в режиме реального времени, что позволит корректировать параметры сети для бесперебойной работы и обеспечения высокого качества передачи.

Список используемых источников

1. Measuring digital development. Facts and figures 2021: ITU. URL: <https://www.itu.int/en/ITU-D/Statistics/Pages/facts/default.aspx> (дата обращения 17.03.2022)
2. Kreutz D., Ramos F.M.V., Verissimo P.E., Rothenberg C.E., Azodolmolky S., Uhlig S. Software-Defined Networking: A Comprehensive Survey // Proceedings of the IEEE. 2015. Vol. 103. Iss. 1. PP. 14–76. DOI:10.1109/JPROC.2014.2371999
3. Панеш А.Х. Достоинства и недостатки программно-конфигурируемых компьютерных сетей // Вестник Адыгейского государственного университета. Серия 4: Естественно-математические и технические науки. 2016. № 3(186). С. 109–113.
4. Баскаков А.Е., Волков А.С. Алгоритм управления ресурсами транспортной программно-конфигурируемой сети связи // Труды МАИ. 2020. № 115. С. 6. DOI:10.34759/trd-2020-115-06
5. Волков А.С., Баскаков А.Е., Бахтин А.А., Горелик А.В. Метод управления ресурсами транспортной ПКС // Системы синхронизации, формирования и обработки сигналов. 2021. Т. 12. № 2. С. 4–10.
6. Рытов М.Ю., Калашников Р.Ю., Горелов А.А. Применение механизма многоуровневой справедливой очереди для снижения ущерба от атак отказа в обслуживании в программно-конфигурируемых сетях // Информация и безопасность. 2021. Т. 24. № 2. С. 253–260. DOI:10.36622/VSTU.2021.24.2.009
7. Данешманд Б.М. Анализ и оценка эффективности методов обеспечения качества обслуживания программно-конфигурируемых сетей стандарта 5G/IMT-2020 // Russian Technological Journal. 2021. Т. 9. № 5. С. 14–25. DOI:10.32362/2500-316X-2021-9-5-14-25
8. Агеева А.Д., Бирюкова Н.В., Мошков В.В., Елагин В.С. Транспортные программно-конфигурируемые сети // Modern Science. 2019. № 12-4. С. 291–301.
9. Маньков В.А., Краснова И.А. Алгоритм динамической классификации потоков в мультисервисной SDN-сети // T-Comm: Телекоммуникации и транспорт. 2017. Т. 11. № 12. С. 37–42.

10. Ибрагимов Б., Керимов В. Об одном подходе к оценке ресурсов в мультисервисных сетях связи на базе единого инфокоммуникационного пространства // XV Ежегодная Международная научно-техническая конференция «IT-Технологии: развитие и приложения» (Владикавказ, Россия, 12–14 декабря 2018). Владикавказ: Северо-Кавказский горно-металлургический институт (Государственный технологический университет), 2018. С. 283–293.
11. Елагин В.С. Динамическое управление нагрузкой в программно-конфигурируемых сетях // Труды учебных заведений связи. 2017. Т. 3. № 3. 60–67.
12. Амельянович А.В., Шпаков М.Н., Мутханна А.С., Буйневич М.В., Владыко А.Г. Централизованное управление потоками трафика в беспроводных локальных сетях на базе концепции SDN // Системы синхронизации, формирования и обработки сигналов. 2017. Т. 8. № 2. С. 31–35.
13. Гузев О.Ю., Чижов И.В. Балансировка нагрузки в защищенных сетях с использованием технологии SDN // Системы и средства информатики. 2018. Т. 28. № 1. С. 123–138. DOI:10.14357/08696527180110
14. Гузев О.Ю., Чижов И.В. SDN-балансировка нагрузки на криптографические маршрутизаторы при объединении центров обработки данных // Системы и средства информатики. 2018. Т. 28. № 1. С. 139–155. DOI:10.14357/08696527180111
15. Братченко Н.Ю., Мочалов В.П., Яковлев С.В. Разработка имитационной модели системы управления качеством инфокоммуникационных услуг сетей SDN // Современная наука и инновации. 2019. № 3(27). С. 44–53. DOI:10.33236/2307-910X-2019-3-27-47-56
16. Пашков В.Н. Распределенная отказоустойчивая платформа управления для программно-конфигурируемых сетей // Моделирование и анализ информационных систем. 2019. Т. 26. № 1(79). С. 101–121. DOI:10.18255/1818-1015-2019-1-101-121
17. Sufiev H., Haddad Y. A dynamic load balancing architecture for SDN // Proceedings of the International Conference on the Science of Electrical Engineering (ICSEE, Eilat, Israel, 16–18 November 2016). IEEE, 2016. DOI:10.1109/ICSEE.2016.7806104
18. Sufiev H., Haddad Y., Barenboim L., Soler J. Dynamic SDN Controller Load Balancing // Future Internet. 2019. Vol. 11. Iss. 3. P. 75. DOI:10.3390/fi11030075
19. Aly W.H.F. Controller Adaptive Load Balancing for SDN Networks // Proceedings of the Eleventh International Conference on Ubiquitous and Future Networks (ICUFN, Zagreb, Croatia, 2–5 July 2019). IEEE, 2019. DOI:10.1109/ICUFN.2019.8805922
20. Aly W.H.F. Generic Controller Adaptive Load Balancing (GCALB) for SDN Networks // Journal of Computer Networks and Communications. 2019. DOI:10.1155/2019/6808693
21. Bharanidharan C., Gandhi S.I., Devapriya R.D. An Enhanced Framework for Traffic Load Balancing and QoS Provisioning in SDN // Wireless Personal Communications. 2021. Vol. 121. PP. 3451–3472. DOI:10.1007/s11277-021-08886-2
22. Killi B.R., Rao S.V. Poly-stable matching based scalable controller placement with balancing constraints in SDN // Computer Communications. 2020. Vol. 154. PP. 82–91. DOI:10.1016/j.comcom.2020.02.053
23. Shi X., Li Y., Xie H., Yang T., Zhang L., Liu P., et al. An OpenFlow-Based Load Balancing Strategy in SDN // Cmc-Computers Materials & Continua. 2020. Vol. 62. Iss. 1. PP. 385–398. DOI:10.32604/cmc.2020.06418
24. Eligar V., Lyer N., Nihal N.D., Hugar N.S., Kumar P.Y., Manjunath M.N. Load balancing using OpenDaylight SDN controller: Case study // International Research Journal on Advanced Science Hub. 2020. Vol. 2. Iss. 9. PP. 59–64. DOI:10.47392/irjash.2020.149
25. Kaur S., Kumar K., Singh J., Ghumman N.S. Round-robin based load balancing in Software Defined Networking // Proceedings of the 2nd International Conference on Computing for Sustainable Global Development (INDIACom, New Delhi, India, 11–13 March 2015). IEEE, 2015. PP. 2136–2139.
26. Chaudhary R., Kumar N. LOADS: Load Optimization and Anomaly Detection Scheme for Software-Defined Networks // Transactions on Vehicular Technology. 2019. Vol. 68. Iss. 12. PP. 12329–12344. DOI:10.1109/TVT.2019.2948222
27. Li G., Gao T., Zhang Z., Chen Y. Fuzzy Logic Load-Balancing Strategy Based on Software-Defined Networking // Proceedings of the 10th International Conference (WiCON 2017, Tianjin, China, 16–17 December 2017). Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering. Vol. 230. Cham: Springer, 2018. PP. 471–482. DOI:10.1007/978-3-319-90802-1_42
28. Soleimanzadeh K., Ahmadi M., Nassiri M. SD-WLB: An SDN-aided mechanism for web load balancing based on server statistics // ETRI Journal. 2019. DOI:10.4218/etrij.2018-0188
29. Babayigit B., Ulu B. Deep learning for load balancing of SDN-based data center networks // International Journal of Communication Systems. 2021. DOI:10.1002/dac.4760
30. Zhong H., Fang Y., Cui J. Reprint of “LBBSRT: An efficient SDN load balancing scheme based on server response time” // Future Generation Computer Systems. 2017. Vol. 80. PP. 409–416. DOI:10.1016/j.future.2017.11.012
31. Malavika R., Valarmathi M.L. Load Balancing Based on Closed Loop Control Theory (LBCLCT): A Software Defined Networking (SDN) powered server load balancing system based on closed loop control theory // Concurrency and Computation: Practice and Experience. 2022. DOI:10.1002/cpe.6854
32. Babbar H., Parthiban S., Radhakrishnan G., Rani S. A genetic load balancing algorithm to improve the QoS metrics for software de-fined networking for multimedia applications // Multimedia Tools and Applications. 2022. DOI:10.1007/s11042-021-11467-x
33. Hu T., Yi P., Zhang J., Lan J. Reliable and load balance-aware multi-controller deployment in SDN // China Communications. 2018. Vol. 15. Iss. 11. PP. 184–198. DOI:10.1109/CC.2018.8543099
34. Adekoya O., Aneiba A., Patwary M. An Improved Switch Migration Decision Algorithm for SDN Load Balancing // Open Journal of the Communications Society. 2020. Vol. 1. PP. 1602–1613. DOI:10.1109/ojcoms.2020.3028971

* * *

Comparative Analysis of Network Resource Management Methods in SDN

J. Dmitrieva¹ 

¹The Bonch-Bruевич Saint Petersburg State University of Telecommunications,
St. Petersburg, 193232, Russian Federation

Article info

DOI:10.31854/1813-324X-2022-8-1-73-83

Received 8th February 2022

Revised 2nd March 2022

Accepted 5th March 2022

For citation: Dmitrieva J. Comparative Analysis of Network Resource Management Methods in SDN. *Proc. of Telecom. Universities*. 2022;8(1):73–83. (in Russ.) DOI:10.31854/1813-324X-2022-8-1-73-83

Abstract: *The article is devoted to the analysis of network resource management methods in the SDN network. In the interests of this, a review of the scientific works was carried out by Russian and foreign authors to identify the solutions used in them, as well as their advantages and disadvantages. The methods covered in scientific articles are systematized according to the following criteria: year of publication, level of applicability of the method (physical, logical, network), availability of a mathematical model, load balancing mode relative to the time of the incident, range of tasks to be solved, degree of research (theoretical, experimental, practical). Based on the analysis of the results of the review, it was concluded that the solutions were unsatisfactory, as well as the need to consider the new IBN technology, covering a wide range of applications and working preventively at all SDN levels.*

Keywords: SDN, QoS, OpenFlow, IBN, load balancing, benchmarking.

References

1. ITU. Measuring digital development. Facts and figures 2021. URL: <https://www.itu.int/en/ITU-D/Statistics/Pages/facts/default.aspx> [Accessed 17 March 2022]
2. Kreutz D., Ramos F.M.V., Verissimo P.E., Rothenberg C.E., Azodolmolky S., Uhlig S. Software-Defined Networking: A Comprehensive Survey. *Proceedings of the IEEE*. 2015;103(1):14–76. DOI:10.1109/JPROC.2014.2371999
3. Panesh A.Kh. The Advantages and Disadvantages of Software-Defined Computer Networks. *The Bulletin of Adyghe State University: Internet Scientific Journal*. 2016;3(186):109–113. (in Russ.)
4. Baskakov A., Volkov A. Resources Managing Algorithm for Transport Software-Defined Communication Network. *Trudy MAI*. 2020;115:6. (in Russ.) DOI:10.34759/trd-2020-115-06
5. Volkov A.S., Baskakov A.E., Bakhtin A.A., Gorelik A.V. Transport SDN Resource Management Method. *Systems of Signal Synchronization, Generating and Processing*. 2021;12(2):4–10. (in Russ.)
6. Rytov M.Y., Kalashnikov R.Y., Gorelov A.A. Application of Multi-Level Fair Queue Mechanism for Mitigating Denial-of-Service Attacks on Software-Defined Networks. *Informatsiia i bezopasnost*. 2021;24(2):253–260. (in Russ.) DOI:10.36622/VSTU.2021.24.2.009
7. Daneshmand B. Analysis and Evaluation of the Effectiveness of Methods for Ensuring the Quality of Service for Software-Defined Networks of the Standard 5G/IMT-2020. *Russian Technological Journal*. 2021;9(5):14–25. (in Russ.) DOI:10.32362/2500-316X-2021-9-5-14-25
8. Ageeva A.D., Biryukova N.V., Moshkov V.V., Elagin V.S. Transport Software Defined Networks. *Modern Science*. 2019;12-4:291–301. (in Russ.)
9. Mankov V.A., Krasnova I.A. Algorithm for dynamic classification of flows in a multiservice software defined network. *T-Comm*. 2017;11(12):37–42. (in Russ.)
10. Ibrahimov B., Kerimov V. About One Approach to the Assessment of Resources in Multiservice Communication Networks on the Basis of A Single Infocommunication Space. *Proceedings of the XV Annual International Scientific and Technical Conference on IT Technologies: Development and Applications, 12–14 December 2018, Vladikavkaz, Russia*. Vladikavkaz: North Caucasian Institute of Mining and Metallurgy Publ.; 2018. p.283–293. (in Russ.)
11. Elagin V. In Software-Defined Network. *Proc. of Telecom. University*. 2017;3(3):60–67. (in Russ.)
12. Amelyanovich A.V., Shpakov M.N., Mutkhanna A.S., Buynevich M.V., Vladko A.G. Centralized Control of Traffic Flows in Wireless Local Area Networks Based on the Concept of SDN. *Systems of Signal Synchronization, Generating and Processing*. 2017;8(2):31–35. (in Russ.)
13. Guzev O.Yu., Chizhov I.V. SDN Load Balancing for Secure Networks. *Systems and Means of Informatics*. 2018;28(1):123–138. DOI:10.14357/08696527180110

14. Guzev O.Yu., Chizhov I.V. SDN Load Balancing on L3-VPN Gateways in Data Centers Interconnection. *Systems and Means of Informatics*. 2018;28(1):139–155. DOI:10.14357/08696527180111
15. Bratchenko N.Y., Mochalov V.P., Yakovlev S.V. The Development of Imitating Model of System of Quality Management of Information and Communication Services SDN. *Sovremennaa nauka i innovacii*. 2019;3(27):44–53. DOI:10.33236/2307-910X-2019-3-27-47-56
16. Pashkov V.N. Fault-Tolerance Distributed Control Plane for Software Defined Networks. *Modeling and Analysis of Information Systems*. 2019;26(1):101–121. DOI:10.18255/1818-1015-2019-1-101-121
17. Sufiev H., Haddad Y. A dynamic load balancing architecture for SDN. *Proceedings of the International Conference on the Science of Electrical Engineering, ICSEE, 16–18 November 2016, Eilat, Israel*. IEEE; 2016. DOI:10.1109/ICSEE.2016.7806104
18. Sufiev H., Haddad Y., Barenboim L., Soler J. Dynamic SDN Controller Load Balancing. *Future Internet*. 2019;11(3):75. DOI:10.3390/fi11030075
19. Aly W.H.F. Controller Adaptive Load Balancing for SDN Networks. *Proceedings of the Eleventh International Conference on Ubiquitous and Future Networks, ICUFN, 2–5 July 2019, Zagreb, Croatia*. IEEE; 2019. DOI:10.1109/ICUFN.2019.8805922
20. Aly W.H.F. Generic Controller Adaptive Load Balancing (GCALB) for SDN Networks. *Journal of Computer Networks and Communications*. 2019. DOI:10.1155/2019/6808693
21. Bharanidharan C., Gandhi S.I., Devapriya R.D. An Enhanced Framework for Traffic Load Balancing and QoS Provisioning in SDN. *Wireless Personal Communications*. 2021;121:3451–3472. DOI:10.1007/s11277-021-08886-2
22. Killi B.R., Rao S.V. Poly-stable matching based scalable controller placement with balancing constraints in SDN. *Computer Communications*. 2020;154:82–91. DOI:10.1016/j.comcom.2020.02.053
23. Shi X., Li Y., Xie H., Yang T., Zhang L., Liu P., et al. An OpenFlow-Based Load Balancing Strategy in SDN. *Cmc-Computers Materials & Continua*. 2020;62(1):385–398. DOI:10.32604/cmc.2020.06418
24. Eligar V., Lyer N., Nihal N.D., Hugar N.S., Kumar P.Y., Manjunath M.N. Load balancing using OpenDaylight SDN controller: Case study. *International Research Journal on Advanced Science Hub*. 2020;2(9):59–64. DOI:10.47392/irjash.2020.149
25. Kaur S., Kumar K., Singh J., Ghumman N.S. Round-robin based load balancing in Software Defined Networking. *Proceedings of the 2nd International Conference on Computing for Sustainable Global Development, INDIACom, 11–13 March 2015, New Delhi, India*. IEEE; 2015. p.2136–2139.
26. Chaudhary R., Kumar N. LOADS: Load Optimization and Anomaly Detection Scheme for Software-Defined Networks. *Transactions on Vehicular Technology*. 2019;68(12):12329–12344. DOI:10.1109/TVT.2019.2948222
27. Li G., Gao T., Zhang Z., Chen Y. Fuzzy Logic Load-Balancing Strategy Based on Software-Defined Networking. *Proceedings of the 10th International Conference, WiCON 2017, 16–17 December 2017, Tianjin, China. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*. vol.230. Cham: Springer; 2018. p.471–482. DOI:10.1007/978-3-319-90802-1_42
28. Soleimanzadeh K., Ahmadi M., Nassiri M. SD-WLB: An SDN-aided mechanism for web load balancing based on server statistics. *ETRI Journal*. 2019. DOI:10.4218/etrij.2018-0188
29. Babayigit B., Ulu B. Deep learning for load balancing of SDN-based data center networks. *International Journal of Communication Systems*. 2021. DOI:10.1002/dac.4760
30. Zhong H., Fang Y., Cui J. Reprint of "LBBSRT: An efficient SDN load balancing scheme based on server response time". *Future Generation Computer Systems*. 2017;80:409–416. DOI:10.1016/j.future.2017.11.012
31. Malavika R., Valarmathi M.L. Load Balancing Based on Closed Loop Control Theory (LBCLCT): A Software Defined Networking (SDN) powered server load balancing system based on closed loop control theory. *Concurrency and Computation: Practice and Experience*. 2022. DOI:10.1002/cpe.6854
32. Babbar H., Parthiban S., Radhakrishnan G., Rani S. A genetic load balancing algorithm to improve the QoS metrics for software de-fined networking for multimedia applications. *Multimedia Tools and Applications*. 2022. DOI:10.1007/s11042-021-11467-x
33. Hu T., Yi P., Zhang J., Lan J. Reliable and load balance-aware multi-controller deployment in SDN. *China Communications*. 2018;15(11):184–198. DOI:10.1109/CC.2018.8543099
34. Adekoya O., Aneiba A., Patwary M. An Improved Switch Migration Decision Algorithm for SDN Load Balancing. *Open Journal of the Communications Society*. 2020;1:1602–1613. DOI:10.1109/ojcoms.2020.3028971

Сведения об авторе:

ДМИТРИЕВА
Юлия Сергеевна

аспирант кафедры инфокоммуникационных систем Санкт-Петербургского государственного университета телекоммуникаций им. проф. М.А. Бонч-Бруевича,
dmitrieva_spbgut@bk.ru
 <https://orcid.org/0000-0002-7736-7121>