Обзорная статья
УДК 004.7
DOI:10.31854/1813-324X-2023-9-6-42-57

Анализ методов идентификации трафика для управления ресурсами в SDN

Юлия Сергеевна Дмитриева ✉, dmitrieva@sut.ru
Дарина Владимировна Окунева, okuneva.dv@sut.ru
Василий Сергеевич Елагин, v.elagin@sut.ru

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

Аннотация: *Статья посвящена анализу методов классификации трафика в сети SDN. Выполнен обзор аналитических подходов идентификации трафика для выявления применяемых в них решений, а также оценки их применимости в сети SDN. Рассмотрены виды машинного обучения и выполнен анализ входных параметров. Методы интеллектуального анализа, освещенные в научных статьях, систематизированы по следующим критериям: параметры идентификации трафика, модель нейронной сети, точность идентификации. На основании анализа результатов обзора сделан вывод о возможности применения рассмотренных решений, а также о необходимости формирования схемы сети SDN с модулем элементов искусственного интеллекта для балансировки нагрузки.*

Ключевые слова: *Software-Defined Networking (SDN), Программно-конфигурируемая сеть (ПКС), Deep packet inspection (DPI), Machine learning (ML), Deep learning (DL), Convolutional Neural Network (CNN), искусственный интеллект (ИИ)*

Финансирование: Публикация подготовлена в рамках прикладных научных исследований СПбГУТ, регистрационный номер 123060900012-6 в ЕГИСУ НИОКТР.

Ссылка для цитирования: Дмитриева Ю.С., Окунева Д.В., Елагин В.С. Анализ методов идентификации трафика для управления ресурсами в SDN // Труды учебных заведений связи. 2023. Т. 9. № 6. С. 42–57. DOI:10.31854/1813-324X-2023-9-6-42-57

Analyzing Traffic Identification Methods for Resource Management in SDN

Julia Dmitrieva ✉, dmitrieva@sut.ru
Darina Okuneva, okuneva.dv@sut.ru
Vasiliy Elagin, v.elagin@sut.ru

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

Abstract: *The article is devoted to the analysis of traffic classification methods in SDN network. The review of analytical approaches of traffic identification to identify the solutions used in them, as well as assessing their applicability in the SDN network. Types of machine learning are considered and input parameters are analyzed. The methods of intelligent analysis covered in the scientific articles are systematized according to the following criteria: traffic identification parameters, neural network model, identification accuracy. Based on the analysis of the review results, the conclusion is made about the possibility of applying the considered solutions, as well as the need to form a scheme of SDN network with a module of artificial intelligence elements for load balancing.*

Keywords: *Software-Defined Networking (SDN), Deep packet inspection (DPI), Machine learning (ML), Deep learning (DL), Convolutional Neural Network (CNN), Artificial Intelligence (AI)*

Funding: *The scientific article was prepared in the framework of applied scientific research of SPBSUT, registration number 123060900012-6 in EGISU R&D.*

For citation: Dmitrieva J., Okuneva D., Elagin V. Analyzing Traffic Identification Methods for Resource Management in SDN. *Proceedings of Telecommun. Univ.* 2023;9(6):42–57. DOI:10.31854/1813-324X-2023-9-6-42-57

Введение

По мере роста спроса на персонализированные сетевые услуги неизбежно возникает необходимость в детальной классификации сетевых потоков. Управление ресурсами в программно-конфигурируемой сети (SDN, *аббр. от англ.* Software-Defined Networking) является важной задачей, для которой существует большое количество подходов и методов [1–3]. В рамках самых перспективных методов балансировки трафика используется идентификация трафика для предобработки и его дальнейшего использования. Необходимо рассмотреть классификацию потоков для выбора наиболее эффективного метода управления ресурсами. В программно-конфигурируемой сети (ПКС) существует много подходов к управлению ресурсами, наиболее перспективные из них используют классификацию приложений, идентификацию трафика для повышения эффективности своей работы. Рассмотрим методы идентификации трафика, чтобы выявить наиболее подходящий и перспективный метод для управления ресурсами.

Актуальность и важность анализа сетевого трафика обусловлены ростом объема данных, сложностью сетевой инфраструктуры, угрозами безопасности и необходимостью оптимизации ресурсов [4–6].

В ряде задач управления ресурсами в SDN, затронутых в данной статье, невозможно рассматривать методы идентификации трафика отдельно от методов классификации и необходимо учитывать балансировку нагрузки.

Структура статьи представляет из себя следующее: в первом разделе рассматриваются методы идентификации трафика, во втором разделе – виды машинного обучения и проводится анализ входных параметров трафика для идентификации и обучения нейронной сети. В третьем разделе уделяется внимание балансировке нагрузки в SDN, основанной на машинном обучении.

Методы идентификации трафика

Методы идентификации сетевого трафика можно разделить на виды [7]: сигнатурный анализ, классификация на основе блоков данных, машинное обучение и другие. Рассмотрим каждый из методов подробнее.

Сигнатурный анализ – это метод анализа трафика, основанный на сравнении пакетов данных с заранее определенными сигнатурами. Сигнатуры – это уникальные шаблоны пакетов данных, которые соответствуют определенным типам трафика. Например, сигнатура веб-трафика будет содержать информацию о типе запроса (GET или POST), URL-адресе и т. д.

Сигнатуры представляют собой уникальные характеристики или паттерны, которые характерны для определенного типа трафика и могут быть основаны на известных характеристиках протоколов, параметрах портов, последовательностях байтов и других факторах, связанных с определенными видами трафика или атак. Однако этот метод также имеет свои ограничения, например, он может быть неэффективен при обнаружении новых типов трафика. Поэтому, помимо анализа сигнатур, также важно использовать другие методы и подходы для классификации трафика [8].

Для формализации различных подходов сигнатурного анализа авторами предложены следующие соотношения (1-4), используемые для идентификации трафика:

1) анализ шаблонов – заключается в сравнении эталонных образцов пакетов конкретных приложений (шаблонов) с пакетами, передаваемыми в исследуемом потоке трафика (1), где $[e_x \dots e_y]$ – последовательность элементов (поля пакета), S – последовательность переданных элементов (поля пакета), C_z – эталонная подпоследовательность, $Class^C$ – набор классов;

2) анализ протокола/состояния – состоит в определении состояний сетевого потока с использованием соответствующего графа, который представляет собой заранее определенный граф состояний для конкретных приложений или протоколов, позволяющий по последовательности прохождения узлов однозначно производить идентификацию (2), где S – последовательность переданных элементов (поля пакета), $[e_i]$ – элемент (поле пакета); $Index_{e_{k-1}}$, $Index_{e_k}(S)$ – индекс элемента e_{k-1}, e_k.;

3) эвристический и поведенческий анализ – заключаются в поиске заданной последовательности пакетов в потоке трафика (3), где S – последовательность переданных элементов (пакеты), $[e_1 \dots e_N]$ – последовательность элементов (характеристика пакета), T – цель или последний узел графа,

$Model\ Class^c([e_x \dots e_y])$ – процесс моделирования модели, который позволяет проходить все узлы графа $e_x \dots e_y$, e_t – узел, принадлежащий T^c.

4) числовой анализ – вычисляет характерные для отдельных приложений параметры пакетов в соответствующей последовательности и сравнивает их с эталонными (4), где x – индекс пакета, p – индекс характеристики, P_p – характеристика пакета e_x с индексом p, P_p – набор параметров пакета, V_p^c – множество значений, которым принадлежит пакет (множество значений для пакетов), $V_{P_k}^c$ – множество значений сигнатур, принадлежащих какому-либо классу, т. е. для любого k из набора параметров пакета P_k принадлежит множеству $V_{P_k}^c$.

В сигнатурном анализе выделен комплексный подход к классификации трафика – метод глубокого анализа пакетов (DPI, *аббр. от англ.* Deep Packet Inspection), основанный на предварительно вычисленных или самообучающихся сигнатурах. Он позволяет проводить глубокий анализ пакетов данных, передаваемых по сети, и выявлять различные типы трафика, такие как веб-серфинг, потоковое видео, файлы, мессенджеры и т. д. Преимуществом сигнатурного анализа является быстрота обработки трафика и высокая точность выявления типов трафика. Однако этот метод неэффективен для обнаружения новых типов трафика или для обхода сетевых устройств, которые могут изменять пакеты данных.

$$\begin{cases} S \equiv [e_1 \dots e_n] \\ \{Class^c\}: [C_z]^c \Rightarrow Class^c \\ Method_{pa}(S, \{Class^c\}) := \exists_{x,y}: [e_x \dots e_y] \equiv [C_z]^c \Rightarrow S \in Class^c \end{cases} \quad (1)$$

$$\begin{cases} S = [e_i] \\ \{Class^c\}: Class^c \Rightarrow [e_k]^c \\ Method_{p/sa} \equiv (S, \{Class^c\}) := \exists [e_k] = [e_k]^c, e_k \in S: Index_{e_{k-1}}(S) < Index_{e_k}(S) \rightarrow S \in Class^c \end{cases} \quad (2)$$

$$\begin{cases} S \equiv [e_{1\dots} e_N] \\ Method_{pea}(S, \{Class^c\}) := \exists_{x,y}: ModelClass^c([e_x \dots e_y]) \rightarrow e_t \in T^c \Rightarrow S \in Class^c \end{cases} \quad (3)$$

$$\begin{cases} S \equiv [e_1 \dots e_x \dots e_y \dots e_k] \\ Method_{na}(S, \{Class^c\}) := \exists_x, \exists_p: P_{P_k}(e_x) \in V_{P_k}^c \Rightarrow S \in Class^c \end{cases} \quad (4)$$

Классификация на основе блоков данных базируется на анализе открытых полей пакета, таких как порты, IP-адрес отправителя/получателя или MAC отправителя/получателя и т. д. Данный метод является наиболее применимым, но он не работает с зашифрованным и туннелированным трафиком.

До появления DPI и машинного обучения, сопоставление типов портов приемущественно использовалось для идентификации сетевого трафика, но в настоящее время этот метод для таких целей не используется, в основном, соответствует простой концепции обнаружения трафика; большинство приложений P2P (*аббр. от англ.* Peer to Peer, одноранговая сеть) имеют свой порт по умолчанию, например, BitTorrent имеет порт 6881 6889 TCP/UDP. Если у специалиста нет дорогостоящей системы обнаружения вторжений, то без использования каких-либо идентификаторов сетевой администратор может определить тип сетевого трафика. Сейчас P2P-приложения используют динамический порт, в этом случае сопоставление типов портов неэффективно, а также невозможно правильно идентифицировать сетевой трафик и P2P-приложение.

Машинное обучение – это метод анализа трафика, основанный на использовании алгоритмов машинного обучения для выявления типов трафика. Алгоритмы машинного обучения обучаются на основе большого количества примеров пакетов данных различных типов, и затем используются для классификации новых пакетов данных. Преимуществом машинного обучения является его способность обнаруживать новые типы трафика и обходить сетевые устройства, которые изменяют пакеты данных. Кроме того, этот метод более точен, чем анализ поведения. Однако машинное обучение требует больших вычислительных ресурсов для обработки трафика и обучения алгоритмов.

В зависимости от конкретных задач и требований, в системах DPI могут использоваться различные методы анализа трафика. Комбинация нескольких методов может повысить эффективность обнаружения типов трафика и уменьшить количество ложных срабатываний.

Самым популярным методом исследования пакетов в сети на данный момент является DPI. Однако система глубокого анализа пакетов имеет ряд недостатков: большая нагрузка на сеть, невозможность анализировать шифрованный трафик и низ-

кая производительность в работе с новыми видами приложений. Таким образом, система глубокого анализа пакетов теряет свою эффективность в связи с большим ростом трафика в сети Интернет и с резким увеличением количества различных сервисов и приложений. Системы искусственного интеллекта (ИИ) по сравнению с системой DPI более производительны, гибки за счет самостоятельного обучения и лишены вышеуказанных недостатков.

Наиболее часто используемые методы классификации трафика – по типу портов и на основе анализа полезной нагрузки, имеют ряд недостатков и не позволяют классифицировать приложения нового поколения, такие как P2P и онлайн-игры [9], а точность этого метода составляет не более 30–70 % [10].

Метод анализа полезной нагрузки, представляющий собой глубокую проверку сетевых пакетов, дает хорошие результаты, особенно для P2P-сетей, но его нельзя использовать для зашифрованного трафика, и одной из проблем этого метода является отсутствие конфиденциальности пользователей, которая не соблюдается из-за углубленных проверок трафика [11]. Чтобы преодолеть ограничения предыдущих методов и повысить точность классификации трафика, принято решение использовать алгоритмы машинного обучения [12–14].

Виды машинного обучения

Существуют разные виды искусственного интеллекта, из которых самым известным и широко используемым является машинное обучение (ML, *аббр. от англ.* Machine Learning). Основное преимущество применения методов машинного обучения заключается в сокращении времени обработки больших объемов данных и высокой точности результатов.

Машинное обучение позволяет решать ряд задач, среди которых можно выделить задачу классификации трафика, состоящую из трех основных этапов.

На первом этапе происходит обучение модели на основе обучающего набора данных. Обучающий набор данных содержит объекты трафика, представленные в виде векторов признаков, которые помогают модели выделить определенный целевой класс. Кроме того, обучающий набор данных также содержит метки классов, которые указывают, к какому классу каждый объект трафика относится.

На втором этапе происходит проверка работы модели на тестовом наборе данных. Тестовый набор данных представляет собой независимую выборку, которая не использовалась в процессе обучения. Модель применяется к тестовым данным, и их классификация сравнивается с известными метками классов. Это позволяет оценить точность и производительность модели.

На третьем этапе модель применяется к новым данным для определения их принадлежности к целевому классу. Новые данные представляют собой потоки сетевого трафика, представленные последовательностями IP-пакетов. Данные характеризуются различными признаками: количеством пакетов, IP-адресами, номерами портов и статистическими особенностями потоков. Модель использует полученные знания о зависимостях между признаками и классами, чтобы определить, к какому классу относится каждый новый объект трафика.

Таким образом, процесс классификации трафика методами машинного обучения включает обучение модели на обучающем наборе данных, проверку работы модели на тестовом наборе данных и использование модели для классификации новых данных в целевой класс.

Рассмотрим основные виды машинного обучения, которые используются для классификации сетевого трафика:

1) логистическая регрессия используется для бинарной классификации, то есть разделения объектов на два класса (применяется для прогнозирования вероятности принадлежности объекта к определенному классу на основе признаков);

2) деревья решений представляют собой структуру, состоящую из решающих правил, которые помогают принять решение о классификации объекта (основываются на иерархическом разбиении признакового пространства);

3) случайный лес – это ансамбль деревьев решений, где каждое дерево обучается на случайной подвыборке данных и случайном подмножестве признаков, т. е. каждое дерево принимает решение независимо, а итоговое решение определяется голосованием или усреднением результатов всех деревьев;

4) метод опорных векторов (SVM, *аббр. от англ.* Support Vector Machine) осуществляет поиск оптимальной разделяющей гиперплоскости между различными классами (строит модель, которая классифицирует новые данные, исходя из их положения относительно этой гиперплоскости);

5) нейронные сети состоят из набора обрабатывающих нейронов, которые взаимосвязаны и преобразуют набор входных данных в набор требуемых выходов (результат преобразования распознается по характеристикам нейронов и воздействию, связанному с корреляцией между ними, за счет улучшения связей между узлами сеть может получать необходимые результаты);

6) *k*-ближайших соседей (k-NN, *аббр. от англ.* k-Nearest Neighbors) – это метод классификации, основанный на близости объектов, классифицирует новый объект, исходя из классов его *k* ближайших соседей в обучающем наборе данных;

7) градиентный бустинг – это ансамблевый метод, который комбинирует несколько слабых моделей (обычно деревьев решений) для создания более мощной модели, обучает модели последовательно, взвешивая ошибки предыдущих моделей и пытаясь исправить их;

8) статистический подход опирается на статистические характеристики для идентификации приложения и определяет, какие приложения создают определенные потоки трафика, в частности, Байесовский классификатор – ML-алгоритм, предназначенный для многоклассовой классификации данных с независимыми признаками [15].

В работе [16] предложено несколько типов методов интеллектуального анализа данных классификации:

– наивный Байес (*от англ*. Naive Bayes);

– J48 (Java-реализация алгоритма C4.5 с открытым исходным кодом в программном обеспечении WEKA (*аббр. от англ*. Waikato Environment for Knowledge Analysis);

– проективная адаптивная резонансная теория (PART, *аббр. от англ.* Projective Adaptive Resonance Theory);

– сетевой алгоритм радиально-базисных функций (RBF, *аббр. от англ.* Radial Basis Function) для обнаружения вторжений.

Авторы также собрали набор данных из KDD-cup [17] (*аббр. от англ.* Knowledge Discovery and Data Mining Tools), который имеет 24 типа атак. Сравнивая выше перечисленные методы, отмечено, что алгоритм PART показывает лучшие результаты обнаружения вторжений.

Выбор характеристик потока трафика для обучения модели

Классификация трафика может производиться по различным параметрам, включая тип протокола, порты назначения и источника, размер пакета, время отправки и приема пакета, количество пакетов в потоке и т. д.

Сейчас в большинстве систем классификация производится для потоков. Поток идентифицируется по характеристикам Five-Tuple:

– IP-адрес источника;
– порт источника;
– IP-адрес получателя;
– порт получателя;
– протокол транспортного уровня.

По этим пяти составляющим определяется принадлежность пакета к тому или иному потоку. Таким образом, классификация трафика производится для потока, а не для каждого пакета. В потоке выбирается *N* число пакетов, которое используется для классификации, затем весь поток может быть идентифицирован как приложение, которое было распознано в *N* пакетах.

Входные параметры для модели машинного обучения

Тип протокола прикладного уровня, определяет формат и способы обмена данными между приложениями в компьютерной сети. Приложение использует определенный протокол для обмена данными. Например, HTTP используется для передачи веб-страниц и медиа-контента, SMTP – для отправки и получения электронной почты, FTP – для передачи файлов, а SSH – для удаленного управления устройствами.

Порт источника и назначения (порт – это числовой идентификатор, который используется на транспортном уровне модели OSI для определения, к какому протоколу относится пакет). Хотя порты часто связываются с конкретными приложениями, нельзя гарантировать, что порт всегда указывает на определенный тип приложения, однако, существуют их устойчивые комбинации.

IP-адрес источника и IP-адрес назначения. Часто приложения, такие как Google, Whatsapp, Amazon, Twitter, имеют свои серверы. Соответственно, по IP-адресу получателя в Uplink потоке можно определить, к какому ресурсу был направлен запрос.

Использование протокола шифрования. Большая часть приложений в сети шифруется. Это важно для обеспечения конфиденциальности, целостности и аутентификации данных. Наиболее распространенные протоколы шифрования – это SSL (*аббр. от англ.* Secure Sockets Layer, уровень защищённых сокетов) и TLS (*аббр. от англ.* Transport Layer Security, протокол защиты транспортного уровня). Они используются для шифрования трафика во многих приложениях, особенно тех, которые требуют высокой степени безопасности и защиты данных.

Размер пакета в байтах – является одним из основных параметров сетевого трафика, который может отличаться в зависимости от приложения. Например, мессенджеры могут передавать текстовые сообщения меньшего размера, в то время как ВКС (видеоконференцсвязь) может передавать аудио- и видеоданные большего размера. Этот параметр может быть изменен с помощью сжатия.

Количество переданных пакетов в секунду – параметр, который может показать интенсивность использования приложения. Например, при использовании мессенджера количество переданных пакетов в секунду будет ниже, чем при использовании ВКС. Этот параметр может быть изменен с помощью установки ограничений на скорость передачи данных.

Задержка передачи данных – параметр, который может отличаться в зависимости от типа приложения. Например, при использовании мессенджера

задержка передачи данных будет ниже, чем при использовании ВКС.

Длительность потока – в сети это время от начала потока (инициирования сессии доступа к ресурсу) и до его окончания. Разные типы приложений обычно генерируют трафик с разной длительностью потока. Например, потоки, связанные с веб-браузерами, обычно более короткие, потоки медиаплееров – более длительные, потоки файловых менеджеров имеют разную длительность в зависимости от размера передаваемых файлов.

Количество потоков передачи данных – было выбрано как параметр, который может показать количество одновременных соединений с сервером. Например, при использовании мессенджера количество потоков передачи данных будет ниже, чем при использовании ВКС.

Количество пакетов в потоке – например, для видео и аудиопотоков характерно большое количество пакетов в секунду, причем их размер является постоянной величиной. В то же время приложения, связанные с передачей файлов, могут иметь большое количество пакетов с разным размером.

Длина заголовка. Каждый протокол имеет свой уникальный формат заголовка пакета, длина заголовка может быть разной для разных протоколов. Например, заголовок протокола TCP (*аббр. от англ.* Transmission Control Protocol, протокол управления передачей) имеет длину 20 байт, а заголовок протокола UDP (*аббр. от англ.* User Datagram Protocol, протокол пользовательских дейтаграмм) имеет длину 8 байт.

Сигнатуры – набор данных (условий), которые необходимо найти в трафике, проверяют большое количество полей различных протоколов, таких, как исходный адрес, порт назначения или TCP флаги.

Время поступления данных – моменты времени поступления пакетов на приемной стороне, в некоторых случаях, время начала сессии.

Незашифрованные пакеты подтверждения – пакеты, служащие подтверждающими сообщениями для различных протоколов.

Идентификатор протокола – элемент в структуре IP-пакета, следующий за заголовком протокола, занимает один байт и указывает, какому протоколу верхнего уровня принадлежит информация, размещенная в поле данных пакета.

Пакеты данных для доставки и обслуживания сети – трафик служебных протоколов: ICMP (*аббр. от англ.* Internet Control Message Protocol, протокол межсетевых управляющих сообщений), DHCP (*аббр. от англ* Dynamic Host Configuration Protocol, протокол динамической настройки узла).

Временны́е *интервалы между пакетами*.

Для ML-моделей и нейронных сетей используют разные комбинации входных параметров в зависимости от поставленной задачи, типа сети ИИ, необходимой точности классификации трафика. Комбинация нескольких параметров может улучшить точность классификации сетевого трафика.

Некоторые из таких параметров, как порты и протоколы, могут быть недоступны для анализа в случае шифрованного трафика. Это связано с тем, что в этом случае данные отправляются в зашифрованном виде, что делает невозможным идентификацию протоколов в обычном виде. Тогда для анализа шифрованного трафика могут быть использованы другие параметры, в частности, шаблоны трафика, размер пакетов, частота и время передачи данных.

Согласно [18], только от 30 до 70 % текущего интернет-трафика может быть классифицировано с использованием методов на основе анализа типов портов. Поэтому для классификации современных сетевых услуг необходимы более сложные методы идентификации.

Традиционные методы классификации создают ряд проблем для достижения более высокой точности. Во-первых, классификация на основе параметров портов не обеспечивает достаточную точность, когда в сетевой системе используется переназначение, переадресация и случайное назначение портов. Во-вторых, подход, основанный на сигнатуре полезной нагрузки, плохо работает для шифрованных пакетов. В-третьих, подходы, основанные на статистике, весьма чувствительны к динамически изменяющимся условиям сети и приложений, таким как уровень загруженности сети, интенсивность перекрестного трафика и поведение, зависящее от пользователя. Предлагается более обстоятельно рассмотреть альтернативные способы идентификации трафика на основе применения машинного обучения.

Нейронная сеть анализирует информацию и позволяет оценить согласованность анализируемых данных с характеристиками, которые она научена распознавать.

В работе [19] авторы в качестве модели сети выбрали многослойную нейронную сеть прямого распространения. Исследуется эффективность применения нейронной сети для диагностики аномальной сетевой активности. Искусственно моделируется активность сети путем сканирования портов локального компьютера с удаленной машины. Полученные данные представляют собой выборку для обучения нейронной сети и определяют условия протекания нормальной сетевой активности и возникновения аномальной. Нейронная сеть классифицирует входное пространство признаков на два класса сетевой активности: нормальная (0) и аномальная (1).

Исследователи в работе [22] разработали искусственную модель выборки трафика по следующим параметрам:

– тип протокола транспортного уровня;

– номер назначения транспортного порта;

– сигнатура протокола, которая соответствует приложению, работающему с данным портом транспортного уровня.

Для решения задач идентификации и классификации сетевого трафика проведен сравнительный анализ трех основных типов нейронных сетей.

В докладе [24] на Международной конференции представлен фреймворк нейронной сети, который позволяет строить многоклассовые модели сетевого трафика, подходящие для задач генерации потоков и классификации.

В исследовании [26] решается задача мелкогранулярной классификации для потоков чата методом, основанном на сверточной нейронной сети (CNN, *аббр. от англ.* Convolutional Neural Network) для детальной классификации потоков чатов в реальном времени. Методы классификации сетевого трафика, рассмотренные в этой статье, сосредоточены на классификации зашифрованных потоков, в которых известны для каждого сетевого потока только данные из пяти кортежей: время поступления данных, протокол передачи по сети, размер пакета (при условии, что к конкретному содержимому пакету невозможно получить доступ), данные адреса источника и назначения передачи данных.

Авторы работы [29] на Международной конференции предложили новую классификацию зашифрованного трафика, основанную на полезной нагрузке, использующую пакеты подтверждения, которыми обмениваются конечные хосты для установления защищенной связи до того, как будут доставлены зашифрованные потоки данных.

Методы выбора признаков для классификации трафика производятся иранскими учеными в работе [30]. Они разработали фреймворк, который включает в себя две модели нейронной сети: Naive Bayes и SVM.

В [31] исследовали методы машинного обучения для идентификации приложений с помощью классификации сетевого трафика. В качестве классификаторов использовались сети Байеса.

В работе [32] изучается внедрение глубокой нейронной сети (DNN, *аббр. от англ.* Deep Neural Network) для классификации данных сетевого трафика, где DNN используется для автоматической классификации данных сетевого трафика, собранных с контроллера ONOS (*аббр. от англ.* Open Network Operating System) сети SDN. Исследователи подтвердили, что DNN в этом случае должна учитывать не только пакеты данных, предназначенные для доставки, но и пакеты данных, необходимые для поддержки сети в рабочем состоянии, так как производительность классификации DNN зависит от данных сетевого трафика.

В исследовании [6] авторы используют методы машинного обучения в процессе идентификации приложений для смартфонов. Они рассматривают, как поведение трафика приложений меняется со временем на разных устройствах и в разных версиях приложений, и могут ли приложения для смартфонов быть идентифицированы путем анализа исходящего от них зашифрованного сетевого трафика.

Исследователи в статье [33] предлагают сквозной метод классификации зашифрованного трафика с использованием одномерных сверточных нейронных сетей (1D-CNN, *аббр. от англ*. One-Dimensional Convolution Neural Networks) [34]. Метод основан на глубоком обучении. 1D-CNN применяется в качестве алгоритма обучения и используется для автоматического изучения характеристик необработанного трафика. Объединяет разработку, извлечение и выбор признаков в единый фреймворк.

В работе [35] авторы занимались проблемой классификации трафика, которая заключалась в том, что признаки должны быть выделены экспертом. Поиск признаков, которые позволяют классифицировать трафик, очень утомителен и отнимает много времени, поэтому в данной работе входные параметры трафика выбираются с применением метода на основе оценочной модели, используется комбинация CNN, метаэвристического алгоритма ant-lion (ALO, *аббр. от англ.* Ant-Lion Optimization) и самоорганизующейся карты (SOM, *аббр. от англ.* Self Organizing Map) для создания модели классификации трафика, которая может точно идентифицировать типы трафика.

Основные этапы анализируемого метода:

– предварительная обработка данных о трафике с использованием энтропии и дисперсии энтропии;

– автоматическое извлечение признаков из скрытых слоев одномерной сверточной нейронной сети (1D CNN);

– эффективный выбор признаков с высокой точностью классификации с использованием ALO;

– классификация новых экземпляров с использованием кластеризации на основе fuzzy-SOM (*от англ.* fuzzy, нечеткая).

Предлагаемый метод позволяет идентифицировать зашифрованный трафик, динамические протоколы, такие как P2P, различать VPN-трафик и не являющийся VPN, классифицировать различные типы трафика с высокой точностью. Входные параметры трафика для идентификации и обучения нейронной сети представлены в таблице 2.

ТАБЛИЦА 2. Сравнение входных параметров трафика

TABLE 1. Comparison of Input Traffic Parameters

№	Ссылки	Модель нейронной сети	Параметр трафика	Точность идентификации, %
1	[19], [20], [21]	Многослойная нейронная сеть прямого распространения	– Тип протокола (TCP, UDP и др.) – Порт источника/порт назначения – Размер пакета	96,5
2	[22], [23],	Персептрон, многослойный персептрон, сеть со встречным распространением	– Тип протокола транспортного уровня – Номер назначения порта транспортного уровня – Сигнатура протокола, соответствующая приложению, работающему с данным протоколом	97
3	[24], [25],	Авторегрессионная	– Интервалы между пакетами – Размер пакета	99
4	[26], [27], [28]	CNN	– IP-адрес источника и IP-адрес назначения передачи – Время поступления данных – Протокол передачи – Размер пакета – Порт	94
5	[29], [30], [31], [32]	Байесовская нейронная сеть	– IP-адрес источника и IP-адрес назначения – Номера порта источника, номера порта назначения – Идентификатор протокола – Пакеты	99
6	[33], [34]	Одномерная CNN	Зашифрованный поток трафика	90,5
7	[35]	CNN метаэвристического алгоритма ant-lion и SOM	Параметры комбинируются индивидуально с использованием метода на основе оценочной модели	98

Исходя из анализа таблицы 2, по точности идентификации охватываемых параметров трафика представляется актуальным использовать модель CNN, метаэвристический алгоритм ant-lion, и ML-метод SOM, так как параметры комбинируются индивидуально с использованием метода на основе оценочной модели.

С учетом того, что основным элементом управления является ИИ, необходимо сформировать систему на базе ИИ, которая будет регулировать и управлять балансировкой трафика. Исходя из общей структуры SDN, модуль ИИ будет расположен между уровнем управления и уровнем данных, как показано на рисунке 1.

Балансировка нагрузки в SDN на основе искусственного интеллекта

Подходы к балансировке нагрузки в программно-конфигурируемой сети, основанные на машинном обучении, улучшают возможности обучения и позволяют системе принимать решения.

На рисунке 1 представлена предлагаемая схема структуры SDN, которая состоит из пяти уровней. Уровень модуля ИИ отвечает за производство, объединение и распространение правил и политик по сети.

Модуль ИИ позволяет сетевым менеджерам устранять проблемы до их возникновения внутри сети, увеличивать функциональность сети в ответ на меняющиеся требования и принимать превентивные меры для снижения рисков. Данные и/или правила, сгенерированные в модуле ИИ, используются для выявления и устранения сетевых проблем, настройки сети и т. д.

Уровень модуля ИИ состоит из трех подуровней: обработки данных, системы сбора и хранения исходных данных и подуровня генерации команд. Используя данные и информацию, уровень модуля ИИ создает правила и политики, применяя методы, основанные на алгоритмах или искусственном интеллекте. Собранные данные используются для построения правил посредством интеграции с намерениями пользователя.

Уровень обработки данных создает инновационные правила путем сравнения требований приложений с объединенными полученными знаниями о состоянии сети [36]. Принятие решений реализуется как модель, основанная на правилах, с использованием компьютерного языка Java или Lisp [37]. Правила, созданные на подуровне принятия решений, написаны на универсальном языке, чтобы другие уровни могли их понять. Для написания созданных правил используются специализированные языки правил, например, язык разметки правил (RuleML) [38], формат обмена правилами (RIF) [39], язык правил семантической сети (SWRL) [40].

Уровень системы сбора и хранения исходных данных включает общую базу данных для хранения правил, использует протоколы/языки для ввода, изменения, устранения правил/знаний и совместного использования [41]. Таким образом, рекомендации, полученные с помощью модели управления генерации правил и команд, формирование наборов данных и генерация значений данных, сетевые данные и сообщения с уровня управления составляют общую базу данных на этом подуровне.

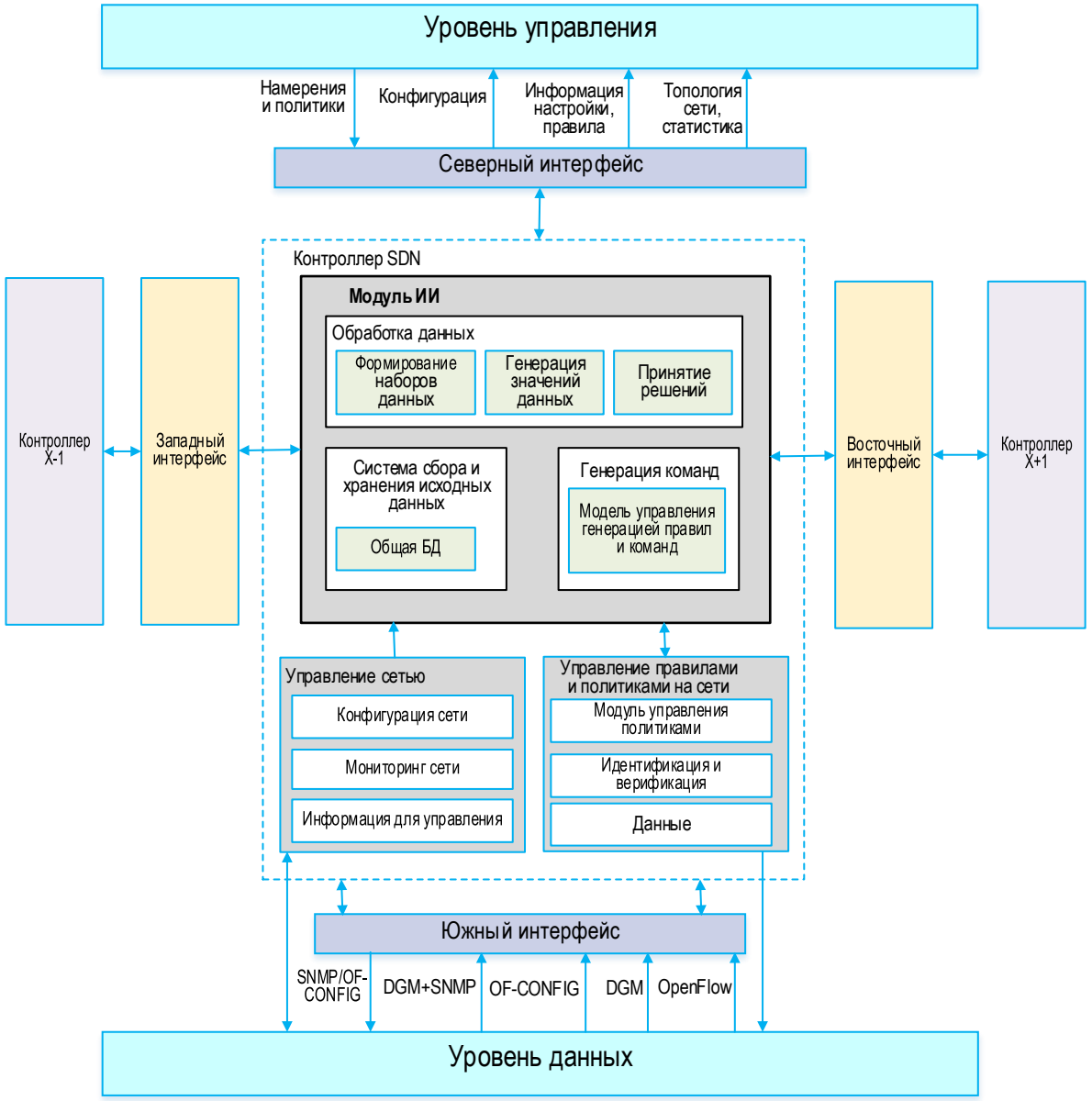

Рис. 1. Предлагаемая структура SDN с модулем ИИ

Fig. 1. Proposed Structure of SDN With AI Module

Механизм управления правилами и политиками на сети применяет правила или делает выводы на основе принятых решений, а затем дает указания в зависимости от того, насколько хорошо были применены правила и знания. Наиболее часто используемым протоколом настройки и управления сетью является протокол сетевой конфигурации (NETCONF, *аббр. от англ.* NETwork-CONFiguration Protocol). Благодаря совместимости с устройствами пересылки протокол OF-CONFIG (аббр. *от англ.* OpenFlow management-and-CONFIGuration Protocol) передает данные через NETCONF [42].

OF-CONFIG или NETCONF могут быть заменены простым протоколом сетевого администрирования (SNMP, *аббр. от англ.* Simple-Network-Management Protocol), разработанным для наблюдения и настройки сетевых устройств в SDN [43].

Следует отметить, что подуровень управления сетью с помощью протокола SNMP/OF-CONFIG собирает данные конфигурации для настройки сети и сетевого наблюдения (статистика трафика, расположение сети, показатели производительности и т. д.). Другие данные могут быть собраны с использованием метода сбора данных DGM (*аббр. от англ.* Data Gathering Method), такого как оптимизация на основе квадратичного целочисленного программирования [44], выборка пакетов [45], адаптивный сбор данных [46] и сбор данных измерений датчиков [47]. Кроме того, уровень управления может собирать различные данные, включая расположение, настройку, схемы движения, записи о событиях, потреблении ресурсов, показатели производительности, показания датчиков и т. д.

Сетевые данные, необходимые для наблюдения за сетью на уровне управления и создания знаний в модуле элементов ИИ, хранятся в модуле, содержащим информацию для управления. Данные внутри подуровня информации для управления могут быть представлены с использованием языка представления данных YANG (*аббр. от англ.* Yet Another Next-Generation) [48] или CIM (*аббр. от англ.* Common-Information Model, общая информационная модель) [49].

Уровень управления и контроллер SDN взаимодействуют друг с другом благодаря «северному» интерфейсу, который может быть реализован с использованием ad hoc [50], RESTful (*от англ.* Representational State Transfer, передача репрезентативного состояния) [51], и сети на основе намерений IBN (*аббр. от англ.* Intent-Based Networking) [52, 53] или языкового API (*от англ.* Application Programming Interface, прикладной программный интерфейс). «Южный» интерфейс служит мостом между уровнем инфраструктуры и уровнем управления при использовании протоколов: OpenFlow, ForCES, OpFlex и других. Он используется для передачи необработанных данных от компонентов пересылки к контроллеру и для передачи правил потока от уровня управления к оборудованию уровня инфраструктуры. Используются интерфейсы, обеспечивающие связь между физически разнесенными контроллерами, ориентированными «с востока на запад»: ALTO [54], Hyperflow [55], ONOS [56], Onix [57] и т. д.

Реактивные подходы к управлению вызывают изменения в сети в ответ на потоки или события. Контроллер предварительно вычисляет компоненты коммутации с помощью набора правил при использовании проактивного управления для управления всеми потенциальными потоками трафика еще до того, как трафик достигнет коммутаторов.

Одной из основных операций контроллера является определение идеального маршрута передачи данных потока (вычисление пути) и оптимизация трафика, которая включает в себя улучшение потоков трафика с целью повышения эффективности сети с помощью собранных данных. Кроме того, уровень управления может собирать необработанные данные: информация о трафике, данные QoS (*аббр. от англ.* Quality of Service, качество обслуживания), правила, инциденты безопасности, протоколы, используемые для маршрутизации, и т. д. [58]. Более того, реализуя механизм политик, уровень управления может инструктировать сетевые устройства, выполнять конкретные задачи для удовлетворения конкретных требований, выполняя политики, как показано на рисунке 1. Сетевые рекомендации преобразуются в правила с помощью подуровня управления правилами и политиками на сети путем учета дополнительной информации, других правил и представлений, основанных на знаниях [59].

Сетевые данные маршрутизируются через последовательность сетевых сервисов с использованием гибкой цепочки, где контроллер выбирает их для включения в цепочку в зависимости от изменяющихся сетевых условий [60]. Кроме того, контроллер также может гибко создавать частные сети, такие как частные виртуальные сети, и масштабировать их в зависимости от требований динамической сети [61].

Разработчики приложений используют прикладной уровень в качестве основы для передачи своих запросов в базовую физическую сеть. Кроме того, это позволяет сетевым менеджерам централизованно устанавливать руководящие принципы настройки сети, которые лучше соответствуют общим бизнес-целям и намерениям в сети IBN [53], при этом функции приложения отделены от аппаратного обеспечения, и определять сетевые политики, уникальные для приложений. Более того, рекомендации по применению могут постоянно изменяться в SDN в зависимости от информации о функционировании сети, что повышает качество предоставления услуг [62].

Прикладной уровень отделяет сервисную функцию от физических компонентов, чтобы централизованно определять желаемые намерения и правила. В результате, при изменении состояния сети принципы и намерения применения могут динамически изменяться. Для достижения задач по определению политики и обновления существуют фреймворки программирования, такие как Procera [63], Nettle [64], Frenetic [65], Kinetic [66] и т. д., которые построены поверх распространенных языков программирования: Python, Haskell и т. д.

В ближайшей перспективе наиболее актуальными прикладными направлениями применения модуля ИИ являются: оптимизация трафика (интеллектуальная пересылка пакетов, оптимизация нагрузки, поддержание и настройка QoS); автономное сетевое администрирование (администрирование действий пользователей; управление мобильностью).

Выводы

На основе обзора научных работ в статье были исследованы аналитические методы классификации трафика и модели машинного обучения на предмет анализа методов идентификации трафика для эффективного управления ресурсами в SDN.

Анализ методов идентификации трафика позволил сделать вывод о том, что аналитические методы классификации трафика имеют ограничения в применении, поэтому принято решение использовать алгоритмы машинного обучения. Был выполнен

анализ входных параметров трафика для идентификации и обучения нейронной сети, который показал, что по точности идентификации охватываемых параметров трафика представляется актуальным использовать CNN-модель, метаэвристический алгоритм ant-lion и ML-метод SOM, так как параметры комбинируются индивидуально с использованием метода на основе оценочной модели.

В статье определено место модуля ИИ в структуре SDN, с учетом особенностей построения таких сетей. Идентификация трафика в сети SDN применяется для балансировки нагрузки, поддержания QoS и эффективного использования сетевого ресурса. Предложена структура внедрения модуля ИИ, в том числе, для идентификации трафика.

В будущих исследованиях необходимо проработать интерфейсы подключения н модуля ИИ к SDN, для минимизации его влияния на задержку при сетевом обмене.

Список источников

1. Дмитриева Ю.С. Сравнительный анализ методов управления сетевыми ресурсами в сетях SDN // Труды учебных заведений связи. 2022. Т. 8. № 1. С. 78–83. DOI:10.31854/1813-324X-2022-8-1-73-83

2. Kirichek R., Vladyko A., Zakharov M, Koucheryavy A. Model networks for Internet of Things and SDN // Proceedings of the 18th International Conference on Advanced Communication Technology (ICACT, PyeongChang, Korea (South), 31 January 2016 – 03 February 2016). IEEE, 2016. PP. 76–79. DOI:10.1109/ICACT.2016.7423280

3. Muhizi S., Shamshin G., Muthanna A., Kirichek R., Vladyko A., Koucheryavy A. Analysis and Performance Evaluation of SDN Queue Model // Proceedings of the 15th IFIP WG 6.2 International Conference on Wired/Wireless Internet Communications (WWIC, St. Petersburg, Russian Federation, 21–23 June 2017. Lecture Notes in Computer Science. Cham: Springer, 2017. Vol. 10372. PP. 26–37. DOI:10.1007/978-3-319-61382-6_3

4. Гетьман А.И., Иконникова М.К. Обзор методов классификации сетевого трафика с использованием машинного обучения // Труды института системного программирования РАН. 2020. Т. 32. № 6. С. 137–154. DOI:10.15514/ISPRAS-2020-32(6)-11

5. Черниговский А.В., Кривов М.В. Нейронные сети как инструмент анализа сетевого трафика // Вестник Ангарского государственного технического университета. 2019. № 13. С. 151–157. DOI:10.36629/2686-777x-2019-1-13-151-157

6. Гетьман А.И., Евстропов Е.Ф., Маркин Ю.В. Анализ сетевого трафика в режиме реального времени: обзор прикладных задач, подходов и решений // Препринт ИСП РАН. 2015. Т. 28. С. 1–52.

7. Ghosh A., Senthilrajan A. Classifying Network Traffic Using DPI And DFI // International Journal of Scientific & Technology Research. Lecture Notes on Data Engineering and Communications Technologies. 2019. Vol. 8. Iss. 11. PP. 3983–3988.

8. Процкая Е.П., Гай В.Е. Программная система анализа сетевого трафика // XXV Международная научно-техническая конференция «Информационные системы и технологии – 2019 (Нижний Новгород, Российская Федерация, 19 апреля 2019). Нижний Новгород: Нижегородский государственный технический университет им. Р.Е. Алексеева, 2019. С. 876–881.

9. Hu L., Zhang L. Real-time internet traffic identification based on decision tree // Proceedings of the World Automation Congress (Puerto Vallarta, Mexico, 24–28 June 2012). IEEE, 2012.

10. Deebalakshmi R., Jyothi V.L. A survey of classification algorithms for network traffic // Proceedings of the Second International Conference on Science Technology Engineering and Management (ICONSTEM, Chennai, India, 30–31 March 2016). IEEE, 2016. PP. 151–156. DOI:10.1109/ICONSTEM.2016.7560941

11. Karagiannis T., Broido A., Brownlee N., Claffy K.C., Faloutsos M. Is P2P dying or just hiding [P2P traffic measurement] // Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM, Dallas, USA, 29 November 2004 – 03 December 2004. IEEE, 2005. DOI:10.1109/GLOCOM.2004.1378239

12. Kohout J., Pevny T. Network Traffic Fingerprinting Based on Approximated Kernel Two-Sample Test // IEEE Transactions on Information Forensics and Security. 2018. Vol. 13. Iss. 3. PP. 788–801. DOI:10.1109/TIFS.2017.2768018

13. Perera P., Tian Y.C., Fidge C., Kelly W. A Comparison of Supervised Machine Learning Algorithms for Classification of Communications Network Traffic // Proceedings of the 24th International Conference on International Conference on Neural Information Processing (ICONIP, Guangzhou, China, 14–18 November 2017). Lecture Notes in Computer Science. Cham: Springer, 2017. Vol. 10634. PP. 445–454. DOI:10.1007/978-3-319-70087-8_47

14. Shi H., Li H., Zhang D., Cheng C., Wu W. Efficient and robust feature extraction and selection for traffic classification // Computer Networks. 2017. Vol. 119. PP. 1–16. DOI:10.1016/j.comnet.2017.03.011

15. Han J., Kamber M., Pie J. Data Mining: Concept and Techniques. Elsever, 2006.

16. Kalyan G., Lakshmi A.J. Performance Assessment of Different Classification Techniques for Intrusion Detection // JORS Journal of Computer Engineering. 2012. Vol. 7. Iss. 5. PP. 2278–8727.

17. Protić D. Review of KDD CUP '99, NSL-KDD and Kyoto 2006+ datasets // Vojnotehnicki glasnik. 2018. Vol. 66. Iss. 3. PP. 580–596 DOI:10.5937/vojtehg66-16670

18. Lotfollahi M., Zade R.S.H., Siavoshani M.J., Saberian M. Deep packet: a novel approach for encrypted traffic classification using deeplearning // Soft Computing. 2020. Vol. 24. PP. 1999–2012. DOI:10.1007/s00500-019-04030-2

19. Катасёв А.С., Катасёва Д.В., Кирпичников А.П. Нейросетевая диагностика аномальной сетевой активности // Вестник технологического университета. 2015. Т. 18. № 6. С. 163–167.

20. Singh K., Agrawal S. Performance Analysis of Back Propagation Neural Network for Internet Traffic Classification // Proceedings of the National Conference on Recent Advances in Electronics and Communication Technologies (RAECT – 2011). 2011.

21. Manju N. Multilayer Feedforward Neural Network for Internet Traffic Classification // Special Issue on Soft Computing. 2023. DOI:10.9781/ijimai.2019.11.002

22. Абдурахманов Р.П., Тожиева Ф.К. Исследование систем управления трафиком на базе моделей нейронных сетей // Наука и мир. 2020. № 4-1;(80):26–32.

23. Ganowicz A., Starosta B., Knapińska A., Walkowiak K. Short-Term Network Traffic Prediction with Multilayer Perceptron // Proceedings of the 6th SLAAI International Conference on Artificial Intelligence (SLAAI-ICAI, Colombo, Sri Lanka, 01–02 December 2022). IEEE, 2022. DOI:10.1109/SLAAI-ICAI56923.2022.10002431

24. Bikmukhamedov R.F., Nadeev A.F. Multi-Class Network Traffic Generators and Classifiers Based on Neural Networks // Systems of Signals Generating and Processing in the Field of on Board Communications (Moscow, Russian Federation, 16–18 March 2021). IEEE, 2021. DOI:10.1109/IEEECONF51389.2021.9416067

25. Azari A., Papapetrou P., Denic S., Peters G. Cellular Traffic Prediction and Classification: A Comparative Evaluation of LSTM and ARIMA // Proceedings of the 22nd International Conference on Discovery Science (DS, Split, Croatia, 28–30 October 2019). Lecture Notes in Computer Science. Cham: Springer, 2019. Vol. 11828. PP. 129–144. DOI:10.1007/978-3-030-33778-0_11

26. Yang L., Wang Z., Feng Y., Yan H. An Effective Real-time Traffic Classification Method Using Convolutional Neural Network // Research Square. 2023. DOI:10.21203/rs.3.rs-3224251/v1

27. Chen X., Wang P., Yu J. CNN based entrypted traffic identification method // Journal of Nanjing University of Posts and Telecommunications (Natural Science). 2018. Vol. 38. PP. 36–41. DOI:10.14132/j.cnki.1673-5439.2018.06.006

28. Guo L., Wu Q., Liu S., Duan M., Li H., Sun J. Deep learning-based real-time VPN encrypted trafc identifcation methods // Journal of Real-Time Image Processing. 2020. Vol. 17. PP. 103–114. DOI:10.1007/s11554-019-00930-6

29. Yang J., Narantuya J., Lim H. Bayesian Neural Network based Encrypted Traffic Classification using Initial Handshake Packets // Proceedings of the 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks – Supplemental Volume (DSN-S, Portland, USA, 24–27 June 2019). IEEE, 2019. DOI:10.1109/DSN-S.2019.00015

30. Izadi S., Ahmadi M., Nikbazm R. Analysis of Feature Selection Methods for Network Traffic Classification // Proceedings of the 8th International Conference on on Advanced Intelligent Systems and Informatics (AISI, Cairo, Egypt, 20–22 November 2022). Lecture Notes on Data Engineering and Communications Technologies. Cham: Springer, 2023. Vol. 152. PP. 65–77. DOI:10.1007/978-3-031-20601-6_6

31. Yamansavascilar B., Guvensan M.A., Yavuz A.G., Karsligil M.E. Application identification via network traffic classification // Proceedings of the International Conference on Computing, Networking and Communications (ICNC, Silicon Valley, USA, 26–29 January 2017). IEEE, 2017. DOI:10.1109/ICCNC.2017.7876241

32. Kwon J., Lee J., Yu M., Park H. Automatic Classification of Network Traffic Data based on Deep Learning in ONOS Platform // Proceedings of the International Conference on Information and Communication Technology Convergence (ICTC, Jeju, Korea (South), 21–23 October 2020). IEEE, 2020. DOI:10.1109/ICTC49870.2020.9289257

33. Wang W., Zeng X., Jinlin W. End-to-end encrypted traffic classification with one-dimensional convolution neural networks // Proceedings of the International Conference on Intelligence and Security Informatics (ISI, Beijing, China, 22–24 July 2017). IEEE, 2018. DOI:10.1109/ISI.2017.8004872

34. Tooke J., Chavula J. Resource-Constrained Real-Time Network Traffic Classification Using One-Dimensional Convolutional Neural Networks // Proceedings of the 13th EAI International Conference on e-Infrastructure and e-Services for Developing Countries (AFRICOMM, Zanzibar, Tanzania, 1–3 December 2021). Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering. Cham: Springer, 2022. Vol. 443. PP. 107–127. DOI:10.1007/978-3-031-06374-9_8

35. Izadi S., Ahmadi M., Nikbazm R. Network traffic classification using convolutional neural network and ant-lion optimization // Computers & Electrical Engineering. 2022. Vol. 101. P. 108024. DOI:10.1016/j.compeleceng.2022.108024

36. Wijesekara P.A.D.S.N., Gunawardena S.A. Comprehensive Survey on Knowledge-Defined Networking // Telecom. 2023. Vol. 4. Iss. 43. PP. 477–596. DOI:10.3390/telecom4030025

37. Jarvis M.P., Nuzzo-Jones G., Heffernan N.T. Applying Machine Learning Techniques to Rule Generation in Intelligent Tutoring Systems // Proceedings of the 7th International Conference on Intelligent Tutoring Systems (ITS, Maceiò, Brazil, 30 August – 3 September 2004). Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2004. Vol. 3220. PP. 541–553. DOI:10.1007/ 978-3-540-30139-4_51

38. Boley H., Tabet S., Wagner G. Design Rationale for RuleML: A Markup Language for Semantic Web Rules // Proceedings of the first Semantic Web Working Symposium (SWWS, Stanford, USA, 30 July – 1 August 2001). Stanford University, 2001. Vol. 1. PP. 381–401.

39. Kifer M. Rule Interchange Format: The Framework // Proceedings of the Second International Conference on Web Reasoning and Rule Systems (RR, Karlsruhe, Germany, 31 October – 1 November 2008). Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2008. Vol. 5341. PP. 1–11. DOI:10.1007/978-3-540-88737-9_1

40. Horrocks I., Patel-Schneider P.F., Boley H., Tabet S., Grosof B., Dean M. SWRL: A Semantic Web Rule Language Combining OWL and RuleML // W3C Member Submission. 2004. PP. 1–31.

41. Wu D., Li Z., Wang J., Zheng Y., Li M., Huang Q. Vision and Challenges for Knowledge Centric Networking // IEEE Wireless Communications. 2019. Vol. 26. Iss. 4. PP. 117–123. DOI:10.1109/MWC.2019.1800323

42. Narisetty R., Dane L., Malishevskiy A., Gurkan D., Bailey S., Narayan S., et al. OpenFlow Configuration Protocol: Implementation for the of Management Plane // Proceedings of the Second GENI Research and Educational Experiment Workshop (Salt Lake City, USA, 20–22 March 2013). IEEE, 2003. PP. 66–67. DOI:10.1109/GREE.2013.21

43. Safrianti E., Sari L.O., Sari N.A. Real-Time Network Device Monitoring System with Simple Network Management Protocol (SNMP) Model // Proceedings of the 3rd International Conference on Research and Academic Community Services (ICRACOS, Surabaya, Indonesia, 9–10 October 2021). IEEE, 2021. PP. 122–127. DOI:10.1109/ICRACOS53680.2021.9701973

44. Wijesekara P.A.D.S.N., Sudheera K.L.K., Sandamali G.G.N., Chong P.H.J. An Optimization Framework for Data Collection in Software Defined Vehicular Networks // Sensors. 2023. Vol. 23. Iss. 3. P. 1600. DOI:10.3390/s23031600

45. Wette P., Karl H. Which flows are hiding behind my wildcard rule? // Proceedings of the conference on SIGCOMM (Hong Kong, China, 12–16 August 2013). New York: ACM, 2013. PP. 541–542. DOI:10.1145/2486001.2491710
46. Zhou D., Yan Z., Liu G. Atiquzzaman, M. An Adaptive Network Data Collection System in SDN // IEEE Transactions on Cognitive Communications and Networking. 2020. Vol. 6. Iss. 2. PP. 562–574. DOI:10.1109/TCCN.2019.2956141
47. Liao W.H., Kuai S.C. An Energy-Efficient SDN-Based Data Collection Strategy for Wireless Sensor Networks // Proceedings of the 7th International Symposium on Cloud and Service Computing (SC2, Kanazawa, Japan, 22–25 November 2017). IEEE, 2017. PP. 91–97. DOI:10.1109/SC2.2017.21
48. Bjorklund M. YANG – A Data Modeling Language for the Network Configuration Protocol (NETCONF). URL: https://www.rfc-editor.org/rfc/rfc6020 (Accessed 19.10.2023)
49. Uslar M., Specht M., Rohjans S., Trefke J., González J.M. The Common Information Model CIM: IEC 61968/61970 and 62325 – A Practical Introduction to the CIM. Berlin, Heidelberg: Springer, 2012. DOI:10.1007/978-3-642-25215-0
50. Gude N., Koponen, T., Pettit J., Pfaff B., Casado M., McKeown N., Shenker S. NOX: towards an operating system for networks // ACM SIGCOMM Computer Communication Review. 2008. Vol. 38. Iss. 3. PP. 105–110. DOI:10.1145/1384609.1384625
51. Rowshanrad S., Abdi V., Keshtgari M. Performance evaluation of SDN controllers: Floodlight and OpenDaylight // IIUM Engineering Journal. 2016. Vol. 17. Iss. 2. PP. 47–57. DOI:10.31436/iiumej.v17i2.615
52. Sanvito D., Moro D., Gulli M., Filippini I., Capone A., Campanella A. ONOS Intent Monitor and Reroute service: Enabling plug&play routing logic // Proceedings of the 4th Conference on Network Softwarization and Workshops (NetSoft, Montreal, Canada, 25–29 June 2018). IEEE, 2018. PP. 272–276. DOI:10.1109/NETSOFT.2018.8460064
53. Дмитриева Ю.С. Управление сетевыми ресурсами на основе намерений // Вестник связи. 2022. № 4. С. 20–26.
54. Rotsos C., King D., Farshad A., Bird J., Fawcett L., Georgalas N., et al. Network service orchestration standardization: A technology survey // Computer Standards & Interfaces. 2017. Vol. 54. Part 4. PP. 203–215. DOI:10.1016/j.csi.2016.12.006
55. Bannour F., Souihi S., Mellouk A. Distributed SDN Control: Survey, Taxonomy, and Challenges // IEEE Communications Surveys & Tutorials. 2017. Vol. 20. Iss. 1. PP. 333–354. DOI:10.1109/COMST.2017.2782482
56. Sanvito D., Moro D., Gulli M., Filippini I., Capone A., Campanella A. ONOS Intent Monitor and Reroute service: Enabling plug&play routing logic // Proceedings of the 4th Conference on Network Softwarization and Workshops (NetSoft, Montreal, Canada, 25–29 June 2018). IEEE, 2018. PP. 272–276. DOI:10.1109/NETSOFT.2018.8460064
57. Koponen T., Casado M., Gude N., Stribling J., Poutievski L., Zhu M., et al. Onix: A Distributed Control Platform for Large-Scale Production Networks // Proceedings of the 9th USENIX Conference on Operating Systems Design and Implementation (OSDI, Vancouver, Canada, 4–6 October 2010). Berkeley: USENIX Association, 2010. PP. 351–364.
58. Zhu M., Cao J., Pang D., He Z., Xu M. SDN-Based Routing for Efficient Message Propagation in VANET // Proceedings of the 10th International Conference on Wireless Algorithms, Systems, and Applications (WASA, Qufu, China, 10–12 August 2015). Lecture Notes in Computer Science). Cham: Springer, 2015. Vol. 9204. PP. 788–797. DOI:10.1007/978-3-319-21837-3_77
59. Moghaddam F.F., Wieder P., Yahyapour R. Policy Engine as a Service (PEaaS): An approach to a Reliable Policy Management Framework in Cloud Computing Environments // Proceedings of the 4th International Conference on Future Internet of Things and Cloud (FiCloud, Vienna, Austria, 22–24 August 2016). IEEE, 2016. PP. 137–144. DOI:10.1109/FiCloud.2016.27
60. Chen Y.J., Wang L.C., Lin F.Y., Lin B.S.P. Deterministic Quality of Service Guarantee for Dynamic Service Chaining in Software Defined Networking // IEEE Transactions on Network and Service Management. 2017. Vol. 14. Iss. 4. PP. 991–1002. DOI:10.1109/TNSM.2017.2758328
61. Yang G., Jin H., Kang M., Moon G.J., Yoo C. Network Monitoring for SDN Virtual Networks // Proceedings of the Conference on Computer Communications (IEEE INFOCOM, Toronto, Canada, 06–09 July 2020). 2020. PP. 1261–1270. DOI:10.1109/INFOCOM41043.2020.9155260
62. Ahvar E., Ahvar S., Raza S.M., Vilchez J. M.S., Lee G.M. Next Generation of SDN in Cloud-Fog for 5G and Beyond-Enabled Applications: Opportunities and Challenges // Network. 2021. Vol. 1. Iss. 1. PP. 28–49. DOI:10.3390/network1010004
63. Voellmy A., Kim H., Feamster N. Procera: a language for high-level reactive network control // Proceedings of the First Workshop on Hot Topics in Software Defined Networks (HotSDN, Helsinki, Finland, 13 August 2012). New York: ACM, 2012. PP. 43–48. DOI:10.1145/2342441.2342451
64. Voellmy A., Hudak P. Nettle: Functional Reactive Programming for OpenFlow Networks. URL: https://pages.cs.wisc.edu/~akella/CS838/F12/838-CloudPapers/Nettle.pdf (Accessed 20.12.2023)
65. Foster N., Freedman M.J., Harrison R., Rexford J., Meola M.L., Walker D. Frenetic: a high-level language for OpenFlow networks // Proceedings of the Workshop on Programmable Routers for Extensible Services of Tomorrow (PRESTO, Philadelphia, USA, 30 November 2010). New York: ACM, 2010. PP. 1–6. DOI:10.1145/1921151.1921160
66. Kim H., Reich J., Gupta A., Shahbaz M., Feamster N., Clark R. Kinetic: Verifiable Dynamic Network Control // Proceedings of the 12th USENIX Symposium on Networked Systems Design and Implementation (NSDI, Oakland, USA, 4–6 May 2015). Berkeley: USENIX Association, 2015. PP. 59–72.

References

1. Dmitrieva J. Comparative Analysis of Network Resource Management Methods in SDN. *Proceedings of Telecommun. Univ*. 2022;8(1):73–83. DOI:10.31854/1813-324X-2022-8-1-73-83
2. Kirichek R., Vladyko A., Zakharov M, Koucheryavy A. Model networks for Internet of Things and SDN. *Proceedings of the 18th International Conference on Advanced Communication Technology, ICACT, 31 January 2016 – 03 February 2016, PyeongChang, Korea (South)*. IEEE; 2016. p.76–79. DOI:10.1109/ICACT.2016.7423280
3. Muhizi S., Shamshin G., Muthanna A., Kirichek R., Vladyko A., Koucheryavy A. Analysis and Performance Evaluation of SDN Queue Model. *Proceedings of the15th IFIP WG 6.2 International Conference on Wired/Wireless Internet Communications,*

WWIC, 21–23 June 2017, St. Petersburg, Russian Federation. Lecture Notes in Computer Science, vol.10372. Cham: Springer; 2017. p.26–37. DOI:10.1007/978-3-319-61382-6_3

4. Getman A.I., Ikonnikov M.K. A survey of network traffic classification methods using machine learning. *Proceedings of ISP RAS*. 2020;32(6):137–154. DOI:10.15514/ISPRAS–2020–32(6)–11
5. Chernigovskiy A.V., Krivov M.V. Neural networks as an instrument of analysis of network traffic. *Bulletin of the Angarsk State Technical University*. 2019;13:151–157. DOI:10.36629/2686-777x-2019-1-13-151-157
6. Getman A.I., Evstropov E.F., Markin Y.V. Wirespeed network traffic analysis: survey of applied problems, approaches and solutions. *ISP RAS preprints*. 2015;28:1–52.
7. Ghosh A., Senthilrajan A. Classifying Network Traffic Using DPI And DFI. *International Journal of Scientific & Technology Research*. 2019;8(11):3983–3988.
8. Prockaya E.P., Gai V.E. Software system analysis of network traffic. *Proceedings of the XXV International Scientific and Technical Conference on Information Systems and Technologies – 2019, 19 April 2019, Nizhny Novgorod, Russian Federation.* Nizhny Novgorod: Nizhny Novgorod State Technical University Publ.; 2019. p.876–881.
9. Hu L., Zhang L. Real-time internet traffic identification based on decision tree. *Proceedings of the World Automation Congress, 24–28 June 2012, Puerto Vallarta, Mexico*. IEEE; 2012.
10. Deebalakshmi R., Jyothi V.L. A survey of classification algorithms for network traffic. *Proceedings of the Second International Conference on Science Technology Engineering and Management, ICONSTEM, 30–31 March 2016, Chennai, India*. IEEE; 2016. p.151–156. DOI:10.1109/ICONSTEM.2016.7560941
11. Karagiannis T., Broido A., Brownlee N., Claffy K.C., Faloutsos M. Is P2P dying or just hiding [P2P traffic measurement]. *Proceedings of the IEEE Global Telecommunications Conference, GLOBECOM, 29 November 2004 – 03 December 2004, Dallas, USA*. IEEE; 2005. DOI:10.1109/GLOCOM.2004.1378239
12. Kohout J., Pevny T. Network Traffic Fingerprinting Based on Approximated Kernel Two-Sample Test. *IEEE Transactions on Information Forensics and Security*. 2018;13(3). DOI:10.1109/TIFS.2017.2768018
13. Perera P., Tian Y.C., Fidge C., Kelly W. A Comparison of Supervised Machine Learning Algorithms for Classification of Communications Network Traffic. *Proceedings of the 24th International Conference on International Conference on Neural Information Processing, ICONIP, 14–18 November 2017, Guangzhou, China. Lecture Notes in Computer Science, vol.10634*. Cham: Springer; 2017. p. 445–454. DOI:10.1007/978-3-319-70087-8_47
14. Shi H., Li H., Zhang D., Cheng C., Wu W. Efficient and robust feature extraction and selection for traffic classification. *Computer Networks.* 2017.119:1–16. DOI:10.1016/j.comnet.2017.03.011
15. Han J., Kamber M., Pie J. *Data Mining: Concept and Techniques*. Elsever: 2006.
16. Kalyan G., Lakshmi A.J. Performance Assessment of Different Classification Techniques for Intrusion Detection. *JORS Journal of Computer Engineering*. 2012;7(5):2278–8727.
17. Protic D. Review of KDD CUP '99, NSL-KDD and Kyoto 2006+ datasets. *Vojnotehnicki glasnik*. 2018;66(3):580–596 DOI:10.5937/vojtehg66-16670
18. Lotfollahi M., Zade R.S.H., Siavoshani M.J., Saberian M. Deep packet: a novel approach for encrypted traffic classification using deeplearning. *Soft Computing*. 2020;24:1999–2012. DOI:10.1007/s00500-019-04030-2
19. Katasev A.S., Kataseva D.V., Kirpichnikov A.P. Neural network diagnosis of abnormal network activity. *Herald of Tekhnological Universite*. 2015;18(6):163–167.
20. Singh K., Agrawal S. Performance Analysis of Back Propagation Neural Network for Internet Traffic Classification. *Proceedings of the National Conference on Recent Advances in Electronics and Communication Technologies, RAECT – 2011*. 2011
21. Manju N. Multilayer Feedforward Neural Network for Internet Traffic Classification. *Special Issue on Soft Computing*. 2023. DOI:10.9781/ijimai.2019.11.002
22. Abdurakhmanov R.P., Tojieva F.Q. The research of traffic management systems based on neural network models. *Science and world*. 2020;4-1(80):26–32.
23. Ganowicz A., Starosta B., Knapińska A., Walkowiak K. Short-Term Network Traffic Prediction with Multilayer Perceptron. *Proceedings of the 6th SLAAI International Conference on Artificial Intelligence, SLAAI-ICAI, 01–02 December 2022, Colombo, Sri Lanka*. IEEE: 2022. DOI:10.1109/SLAAI-ICAI56923.2022.10002431
24. Bikmukhamedov R.F., Nadeev A.F. Multi-Class Network Traffic Generators and Classifiers Based on Neural Networks. *Systems of Signals Generating and Processing in the Field of on Board Communications, 16–18 March 2021, Moscow, Russian Federation*. IEEE; 2021. DOI:10.1109/IEEECONF51389.2021.9416067
25. Azari A., Papapetrou P., Denic S., Peters G. Cellular Traffic Prediction and Classification: A Comparative Evaluation of LSTM and ARIMA. *Proceedings of the 22nd International Conference on Discovery Science, DS, 28–30 October 2019, Split, Croatia. Lecture Notes in Computer Science, vol.11828*. Cham: Springer; 2019. p.129–144. DOI:10.1007/978-3-030-33778-0_11
26. Yang L., Wang Z., Feng Y., Yan H. An Effective Real-time Traffic Classification Method Using Convolutional Neural Network. *Research Square*. 2023. DOI:10.21203/rs.3.rs-3224251/v1
27. Chen X., Wang P., Yu J. CNN based entrypted traffic identification method. *Journal of Nanjing University of Posts and Telecommunications (Natural Science)*. 2018;38:36–41. DOI:10.14132/j.cnki.1673-5439.2018.06.006
28. Guo L., Wu Q., Liu S., Duan M., Li H., Sun J. Deep learning-based real-time VPN encrypted trafc identifcation methods. *Journal of Real-Time Image Processing*. 2020;17:103–114. DOI:10.1007/s11554-019-00930-6
29. Yang J., Narantuya J., Lim H. Bayesian Neural Network based Encrypted Traffic Classification using Initial Handshake Packets. *Proceedings of the 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks – Supplemental Volume, DSN-S, 24–27 June 2019, Portland, USA*. IEEE; 2019. DOI:10.1109/DSN-S.2019.00015
30. Izadi S., Ahmadi M., Nikbazm R. Analysis of Feature Selection Methods for Network Traffic Classificatio. *Proceedings of the 8th International Conference on on Advanced Intelligent Systems and Informatics, AISI, 20–22 November 2022, Cairo, Egypt.*

31. Yamansavascilar B., Guvensan M.A., Yavuz A.G., Karsligil M.E. Application identification via network traffic classification. *Proceedings of the International Conference on Computing, Networking and Communications, ICNC, 26–29 January 2017, Silicon Valley, USA*. IEEE; 2017. DOI:10.1109/ICCNC.2017.7876241

32. Kwon J., Lee J., Yu M., Park H. Automatic Classification of Network Traffic Data based on Deep Learning in ONOS Platform. *Proceedings of the International Conference on Information and Communication Technology Convergence, ICTC, 21–23 October 2020, Jeju, Korea (South)*. IEEE; 2020. DOI:10.1109/ICTC49870.2020.9289257

33. Wang W., Zeng X., Jinlin W. End-to-end encrypted traffic classification with one-dimensional convolution neural networks. *Proceedings of the International Conference on Intelligence and Security Informatics, ISI, 22–24 July 2017, Beijing, China.* IEEE; 2018. DOI:10.1109/ISI.2017.8004872

34. Tooke J., Chavula J. Resource-Constrained Real-Time Network Traffic Classification Using One-Dimensional Convolutional Neural Networks. *Proceedings of the 13th EAI International Conference on e-Infrastructure and e-Services for Developing Countries, AFRICOMM, 1–3 December 2021, Zanzibar, Tanzania. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol.443*. Cham: Springer; 2022. p.107–127. DOI:10.1007/978-3-031-06374-9_8

35. Izadi S., Ahmadi M., Nikbazm R. Network traffic classification using convolutional neural network and ant-lion optimization. *Computers & Electrical Engineering*. 2022;101:108024. DOI:10.1016/j.compeleceng.2022.108024

36. Wijesekara P.A.D.S.N., Gunawardena S.A. Comprehensive Survey on Knowledge-Defined Networking. Telecom. 2023; 4(43):477–596. DOI:10.3390/telecom4030025

37. Jarvis M.P., Nuzzo-Jones G., Heffernan N.T. Applying Machine Learning Techniques to Rule Generation in Intelligent Tutoring Systems. *Proceedings of the 7th International Conference on Intelligent Tutoring Systems, ITS, 30 August – 3 September 2004, Maceiò, Brazil. Lecture Notes in Computer Science, vol.3220*. Berlin, Heidelberg: Springer; 2004. p.541–553. DOI:10.1007/978-3-540-30139-4_51

38. Boley H., Tabet S., Wagner G. Design Rationale for RuleML: A Markup Language for Semantic Web Rules. *Proceedings of the first Semantic Web Working Symposium, SWWS, 30 July – 1 August 2001, Stanford, USA, vol.1.* Stanford University; 2001. p.381–401.

39. Kifer M. Rule Interchange Format: The Framework. *Proceedings of the Second International Conference on Web Reasoning and Rule Systems, RR, 31 October – 1 November 2008, Karlsruhe, Germany. Lecture Notes in Computer Science, vol.5341*. Berlin, Heidelberg: Springer; 2008. p.1–11. DOI:10.1007/978-3-540-88737-9_1

40. Horrocks I., Patel-Schneider P.F., Boley H., Tabet S., Grosof B., Dean M. SWRL: A Semantic Web Rule Language Combining OWL and RuleML. *W3C Member Submission*. 2004:1–31.

41. Wu D., Li Z., Wang J., Zheng Y., Li M., Huang Q. Vision and Challenges for Knowledge Centric Networking. *IEEE Wireless Communications*. 2019;26(4):117–123. DOI:10.1109/MWC.2019.1800323

42. Narisetty R., Dane L., Malishevskiy A., Gurkan D., Bailey S., Narayan S., et al. OpenFlow Configuration Protocol: Implementation for the of Management Plane. *Proceedings of the Second GENI Research and Educational Experiment Workshop, 20–22 March 2013, Salt Lake City, USA*. IEEE; 2003. p.66–67. DOI:10.1109/GREE.2013.21

43. Safrianti E., Sari L.O., Sari N.A. Real-Time Network Device Monitoring System with Simple Network Management Protocol (SNMP) Model. *Proceedings of the 3rd International Conference on Research and Academic Community Services, ICRACOS, 9–10 October 2021, Surabaya, Indonesia*. IEEE; 2021. p.122–127. DOI:10.1109/ICRACOS53680.2021.9701973

44. Wijesekara P.A.D.S.N., Sudheera K.L.K., Sandamali G.G.N., Chong P.H.J. An Optimization Framework for Data Collection in Software Defined Vehicular Networks. *Sensors*. 2023;23(3):1600. DOI:10.3390/s23031600

45. Wette P., Karl H. Which flows are hiding behind my wildcard rule? *Proceedings of the conference on SIGCOMM, 12–16 August 2013, Hong Kong, China*. New York: ACM; 2013. p.541–542. DOI:10.1145/2486001.2491710

46. Zhou D., Yan Z., Liu G. Atiquzzaman, M. An Adaptive Network Data Collection System in SDN. *IEEE Transactions on Cognitive Communications and Networking*. 2020;6(2):562–574. DOI:10.1109/TCCN.2019.2956141

47. Liao W.H., Kuai S.C. An Energy-Efficient SDN-Based Data Collection Strategy for Wireless Sensor Networks. *Proceedings of the 7th International Symposium on Cloud and Service Computing, SC2, 22–25 November 2017, Kanazawa, Japan*. IEEE; 2017. p.91–97. DOI:10.1109/SC2.2017.21

48. Bjorklund M. YANG – A Data Modeling Language for the Network Configuration Protocol (NETCONF). URL: https://www.rfc-editor.org/rfc/rfc6020 [Accessed 19.10.2023]

49. Uslar M., Specht M., Rohjans S., Trefke J., González J.M. *The Common Information Model CIM: IEC 61968/61970 and 62325 – A Practical Introduction to the CIM*. Berlin, Heidelberg: Springer, 2012. DOI:10.1007/978-3-642-25215-0

50. Gude N., Koponen, T., Pettit J., Pfaff B., Casado M., McKeown N., Shenker S. NOX: towards an operating system for networks. *ACM SIGCOMM Computer Communication Review*. 2008;38(3):105–110. DOI:10.1145/1384609.1384625

51. Rowshanrad S., Abdi V., Keshtgari M. Performance evaluation of SDN controllers: Floodlight and OpenDaylight. *IIUM Engineering Journal*. 2016;17(2):47–57. DOI:10.31436/iiumej.v17i2.615

52. Sanvito D., Moro D., Gulli M., Filippini I., Capone A., Campanella A. ONOS Intent Monitor and Reroute service: Enabling plug&play routing logic. *Proceedings of the 4th Conference on Network Softwarization and Workshops, NetSoft, 25–29 June 2018, Montreal, Canada*. IEEE; 2018. p.272–276. DOI:10.1109/NETSOFT.2018.8460064

53. Dmitrieva J. Intent-Based Networking Management. *Communication Bulletin*. 2022;4:20–26.

54. Rotsos C., King D., Farshad A., Bird J., Fawcett L., Georgalas N., et al. Network service orchestration standardization: A technology survey. *Computer Standards & Interfaces*. 2017;54:203–215. DOI:10.1016/j.csi.2016.12.006

55. Bannour F., Souihi S., Mellouk A. Distributed SDN Control: Survey, Taxonomy, and Challenges. *IEEE Communications Surveys & Tutorials*. 2017;20(1):333–354. DOI:10.1109/COMST.2017.2782482

56. Sanvito D., Moro D., Gulli M., Filippini I., Capone A., Campanella A. ONOS Intent Monitor and Reroute service: Enabling plug&play routing logic. *Proceedings of the 4th Conference on Network Softwarization and Workshops, NetSoft, 25–29 June 2018, Montreal, Canada*. IEEE; 2018. p.272–276. DOI:10.1109/NETSOFT.2018.8460064

57. Koponen T., Casado M., Gude N., Stribling J., Poutievski L., Zhu M., et al. Onix: A Distributed Control Platform for Large-Scale Production Networks. *Proceedings of the 9th USENIX Conference on Operating Systems Design and Implementation, OSDI, 4–6 October 2010, Vancouver, Canada*. Berkeley: USENIX Association; 2010. p.351–364.

58. Zhu M., Cao J., Pang D., He Z., Xu M. SDN-Based Routing for Efficient Message Propagation in VANET. *Proceedings of the 10th International Conference on Wireless Algorithms, Systems, and Applications, WASA, 10–12 August 2015, Qufu, China. Lecture Notes in Computer Science, vol.9204.* Cham: Springer; 2015. p.788–797. DOI:10.1007/978-3-319-21837-3_77

59. Moghaddam F.F., Wieder P., Yahyapour R. Policy Engine as a Service (PEaaS): An approach to a Reliable Policy Management Framework in Cloud Computing Environments. *Proceedings of the 4th International Conference on Future Internet of Things and Cloud, FiCloud, 22–24 August 2016, Vienna, Austria*. IEEE; 2016. p.137–144. DOI:10.1109/FiCloud.2016.27

60. Chen Y.J., Wang L.C., Lin F.Y., Lin B.S.P. Deterministic Quality of Service Guarantee for Dynamic Service Chaining in Software Defined Networking. *IEEE Transactions on Network and Service Management*. 2017;14(4):991–1002. DOI:10.1109/TNSM.2017.2758328

61. Yang G., Jin H., Kang M., Moon G.J., Yoo C. Network Monitoring for SDN Virtual Networks. *Proceedings of the Conference on Computer Communications, IEEE INFOCOM, 06–09 July 2020, Toronto, Canada*. IEEE; 2020. p.1261–1270. DOI:10.1109/INFOCOM41043.2020.9155260

62. Ahvar E., Ahvar S., Raza S.M., Vilchez J. M.S., Lee G.M. Next Generation of SDN in Cloud-Fog for 5G and Beyond-Enabled Applications: Opportunities and Challenges. *Network*. 2021;1(1):28–49. DOI:10.3390/network1010004

63. Voellmy A., Kim H., Feamster N. Procera: a language for high-level reactive network control. *Proceedings of the First Workshop on Hot Topics in Software Defined Networks, HotSDN, 13 August 2012, Helsinki, Finland*. New York: ACM; 2012. p.43–48. DOI:10.1145/2342441.2342451

64. Voellmy A., Hudak P. *Nettle: Functional Reactive Programming for Openflow Networks*. URL: https://pages.cs.wisc.edu/~akella/CS838/F12/838-CloudPapers/Nettle.pdf [Accessed 20.12.2023]

65. Foster N., Freedman M.J., Harrison R., Rexford J., Meola M.L., Walker D. Frenetic: a high-level language for OpenFlow networks. *Proceedings of the Workshop on Programmable Routers for Extensible Services of Tomorrow, PRESTO, 30 November 2010, Philadelphia, USA*. New York: ACM; 2010. p.1–6. DOI:10.1145/1921151.1921160

66. Kim H., Reich J., Gupta A., Shahbaz M., Feamster N., Clark R. Kinetic: Verifiable Dynamic Network Control. *Proceedings of the 12th USENIX Symposium on Networked Systems Design and Implementation, NSDI, 4–6 May 2015, Oakland, USA*. Berkeley: USENIX Association; 2015. p.59–72.

Статья поступила в редакцию 01.11.2023; одобрена после рецензирования 24.11.2023; принята к публикации 15.12.2023.
The article was submitted 01.11.2023; approved after reviewing 24.11.2023; accepted for publication 15.12.2023.

Информация об авторах:

ДМИТРИЕВА Юлия Сергеевна
ассистент кафедры инфокоммуникационных сетей и систем Санкт-Петербургского государственного университета телекоммуникаций им. проф. М.А. Бонч-Бруевича
https://orcid.org/0000-0002-7736-7121

ОКУНЕВА Дарина Владимировна
кандидат технических наук, декан факультета инфокоммуникационных сетей и систем Санкт-Петербургского государственного университета телекоммуникаций им. проф. М.А. Бонч-Бруевича
https://orcid.org/0009-0005-4241-8784

ЕЛАГИН Василий Сергеевич
кандидат технических наук, доцент, доцент кафедры инфокоммуникационных систем Санкт-Петербургского государственного университета телекоммуникаций им. проф. М.А. Бонч-Бруевича
https://orcid.org/0000-0003-4077-6869