

Обзорная статья

УДК 004.75

<https://doi.org/10.31854/1813-324X-2025-11-2-67-82>

EDN:QCIUMV



Перспективная архитектура сетей, определяемых знаниями (KDN)

Филип Сергеевич Блан, blan.fs@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

В данной статье рассматривается концепция и архитектура сетей, определяемых знаниями – новой парадигмы управления сетями, которая интегрирует искусственный интеллект и машинное обучение для обеспечения интеллектуального и адаптивного поведения сети.

Актуальность исследования обусловлена ограничениями традиционных и программно-определяемых сетей в условиях современных вызовов, таких как экспоненциальный рост трафика, динамичные условия и увеличение операционных затрат. Рассматриваемые в исследовании сети вводят плоскость знаний, что позволяет оптимизировать распределение ресурсов, автоматизировать принятие решений и повышать безопасность в режиме реального времени. Несмотря на то, что сегодня большой популярностью пользуется технология SDN (Software-Defined Network), в которой централизованная функция управления позволяет обзирать все процессы, происходящие в сети. В свое время ее появление действительно оказалось прорывом, и сейчас некоторые эксперты склоняются к тому, что следующим этапом эволюции сетей станет Knowledge-Defined Network – сеть, определяемая знаниями, действующая на основе алгоритмов машинного обучения. Маршрутизация, распределение ресурсов, виртуализация сетевых функций (Network Functions Virtualization, NFV), цепочка сервисных функций (Service Function Chaining, SFC), обнаружение аномалий, анализ загруженности сети – все эти пункты способна взять на себя KDN.

Цель исследования заключается в изучении структурных и функциональных особенностей сетей, определяемых знаниями, а также – в анализе взаимодействия пяти логических плоскостей: данных, управления, мониторинга, знаний и приложений – для достижения высокой степени автоматизации и адаптации.

Методы включают анализ научной литературы, концептуальное моделирование и сравнительную оценку архитектур определяемой знаниями сети и программно-определяемой сети.

Результаты. В ходе исследования была проанализирована архитектура сетей, определяемых знаниями, и определено, что интеграция плоскости знаний в сеть позволяет добиться значительного повышения автоматизации и адаптивности.

Новизна. Проведенное исследование является одной из первых попыток провести системный анализ концепции сетей, определяемых знаниями, в контексте русскоязычной научной литературы. Работа восполняет существующий пробел в отечественной науке, предлагая уникальный взгляд на возможности сетей, определяемых знаниями, с учетом специфики локальных условий и применения

Теоретическая значимость работы заключается в создании основы для изучения и интеграции методов машинного обучения в системы управления сетями.

Ключевые слова: сети, определяемые знаниями, программно-определяемые сети, автоматизация управления, машинное обучение, интеллектуальное управление

Ссылка для цитирования: Блан Ф.С., Елагин В.С. Перспективная архитектура сетей, определяемых знаниями (KDN) // Труды учебных заведений связи. 2025. Т. 11. № 2. С. 67–82. DOI:10.31854/1813-324X-2025-11-2-67-82. EDN:QCIUMV

Review research

<https://doi.org/10.31854/1813-324X-2025-11-2-67-82>

EDN:QCIUMV

Future Architecture of Knowledge-Defined Networking (KDN)

✉ Filip S. Blan, blan.fs@sut.ru

✉ Vasily S. Elagin, v.elagin@sut.ru

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

Annotation

In this paper, the concept and architecture of Knowledge-Defined Networking (KDN) are explored as a new paradigm of network management that integrates artificial intelligence and machine learning to enable intelligent and adaptive network behavior.

The relevance of the research is driven by the limitations of traditional and Software-Defined Networking (SDN) systems in the face of modern challenges such as exponential traffic growth, dynamic conditions, and rising operational costs. KDN introduces a knowledge plane that optimizes resource allocation, automates decision-making, and enhances security in real-time. Despite the fact that today the SDN (Software-Defined Network) technology is very popular, in which the centralized control function allows to review all processes occurring in the network. At the time, its appearance really turned out to be a breakthrough, and now some experts are inclined to believe that the next stage of network evolution will be the Knowledge-Defined Network - a network defined by knowledge, operating on the basis of machine learning algorithms. Routing, resource allocation, network function virtualization (NFV), service function chaining (Service Function Chaining, SFC), anomaly detection, network load analysis - all these points can be taken on by KDN. **The study aims** to examine the structural and functional features of KDN and analyze the interaction of its five logical planes – data, control, monitoring, knowledge, and applications – to achieve a high degree of automation and adaptability. The research **methods** include literature analysis, conceptual modeling, and a comparative evaluation of KDN and SDN architectures.

The results. The study analyzed the architecture of KDN, comprising five logical planes: data, control, monitoring, knowledge, and applications. The findings demonstrate that integrating the knowledge plane significantly enhances automation and adaptability within the network.

The novelty of this work lies in being one of the first attempts to conduct a systematic analysis of the Knowledge-Defined Networking (KDN) concept in the context of Russian-language scientific literature. The research addresses an existing gap in domestic science, offering a unique perspective on KDN capabilities considering local conditions and applications.

The theoretical significance of the work lies in establishing a foundation for the study and integration of machine learning methods into network management systems.

Keywords: Knowledge-Defined Networking, Software-Defined Networking, management automation, machine learning, intelligent management

For citation: Blan F.S., Elagin V.S. Future Architecture of Knowledge-Defined Networking (KDN). *Proceedings of Telecommunication Universities*. 2025;11(2):67–82. (in Russ.) DOI:10.31854/1813-324X-2025-11-2-67-82. EDN:QCIUMV

Введение

Развитие сетевых технологий прошло несколько ключевых этапов, начиная с традиционных фиксированных архитектур, где управление сетью требовало ручной настройки каждого устройства, до по-

луавтоматических систем, предоставляющих централизованное управление. Первые системы отличались низкой гибкостью и высокой сложностью обслуживания, что делало их малоэффективными для масштабных и динамичных сценариев. С появлением концепции программно-определяемых се-

тей (SDN, *аббр. от англ.* Software-Defined Networking) в начале 2000-х гг. произошло отделение плоскости управления от плоскости данных, что позволило повысить уровень автоматизации и упростить эксплуатацию [1]. Однако в современных реалиях и SDN оказываются ограничены в своей способности адаптироваться к быстро меняющимся условиям и резкому росту трафика, поэтому встал вопрос о переходе от традиционных архитектур к более интеллектуальным и автоматизированным системам [2].

В настоящее время значительную часть сетевой нагрузки составляют запросы на передачу медиа-файлов (аудио-, фото-, видео-) в высоком разрешении, что приводит к формированию огромных объемов передаваемых и хранимых данных. Недостаточные возможности автоматизации в существующих решениях, особенно в динамичных и высоконагруженных средах, приводят к росту операционных затрат, снижению эффективности обслуживания сети и увеличению рисков эксплуатации.

В условиях возросших требований к скорости, точности и гибкости обработки данных, традиционные сети с фиксированной конфигурацией сталкиваются с серьезными ограничениями, поэтому концепция сетей, определяемых знаниями (KDN, *аббр. от англ.* Knowledge-Defined Networking), становится одним из наиболее перспективных направлений эволюции сетевых технологий. Возможности машинного (ML, *аббр. от англ.* Machine Learning) и глубокого обучения (DL, *аббр. от англ.* Deep Learning) в интеллектуальном слое, известном как плоскость знаний, позволяют эффективно влиять на все части системы [3]. Эта плоскость, являющаяся операционным центром сети, обрабатывает данные для автономного принятия решений и адаптации работы в режиме реального времени.

Цель данной статьи – предоставить обзор сетей, определяемых знаниями, рассмотрев их архитектуру, потенциальные преимущества, недостатки и направления дальнейшего развития. В первую очередь, анализируются архитектурные особенности KDN и их ключевые достоинства, включая автоматизацию и адаптивность, которые снижают затраты, нагрузку на сетевых администраторов и повышают эффективность эксплуатации сети. Также обсуждаются ограничения, связанные с необходимостью значительных вычислительных ресурсов и проблемами интеграции с существующими сетевыми инфраструктурами. Наконец, в статье представлены перспективы внедрения KDN в различные сетевые сценарии.

Обзор сетей, определяемых знаниями

Главным отличием KDN является их способность использовать знания для автоматизации и интеллектуального управления сетью, что делает такие сети более гибкими и адаптивными по сравнению

с традиционными подходами. Если же в традиционных сетях управление основывается на жестко заданных правилах, определяемых администраторами, в KDN подход смещается от информационно-ориентированного к знание-ориентированному. Знание в контексте KDN представляет собой обработанную с помощью машинного обучения информацию, полученную на основе данных, собираемых из сети. Знания представляют собой более высокий уровень абстракции, который формируется через анализ данных, накопленный опыт и обучение [4].

Архитектура KDN организована таким образом, чтобы на каждом уровне, от сбора данных до их анализа и применения, обеспечивать непрерывный цикл получения, обработки и использования знаний. Особенно важна интеграция знаний в плоскости мониторинга и контроля, которые позволяют KDN адаптироваться к изменяющимся условиям и автономно оптимизировать свою работу.

Архитектура KDN включает в себя пять логических плоскостей (рисунок 1), расширяя типичную структуру SDN.

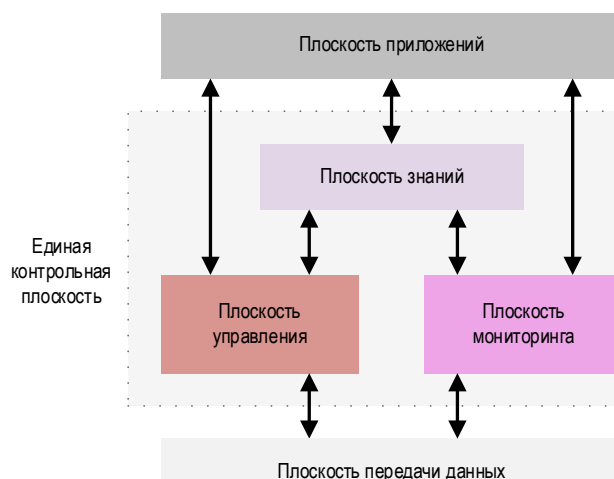


Рис. 1. Схематическая блок-диаграмма архитектуры KDN

Fig. 1. Schematic Block Diagram of the KDN Architecture

Плоскость передачи данных (Data Plane) отвечает за обработку и передачу данных, следуя правилам, заданным *плоскостью управления* (Control Plane). Здесь осуществляется управление маршрутизацией и настройкой сетевых устройств, передавая правила и инструкции в *плоскость передачи данных*. *Плоскость мониторинга* (Management Plane, встречается так же Measurement Plane) обеспечивает сбор данных, контроль состояния сети и конфигурацию сетевых устройств. *Плоскость знаний* (Knowledge Plane) генерирует, обрабатывает и предоставляет знания на основе анализа данных для интеллектуального управления сетью. Далее *плоскость приложений* (Application Plane) предоставляет платформу для взаимодействия сетевых приложений с инфраструктурой и определения высокоуровневых политик.

Следует отметить, что плоскости управления, мониторинга и знаний могут быть функционально интегрированы в единую контрольную плоскость. Такое объединение позволяет проводить параллели с архитектурой SDN, где логически централизованный уровень выполняет схожие задачи координации, наблюдения и регулирования состояния сети [5].

Архитектура KDN отличается высокой степенью согласованности взаимодействия всех пяти плоскостей, что обеспечивает способность системы мгновенно адаптироваться к изменениям внешних и внутренних условий. Это позволяет достичь значительного уровня автоматизации и внедрения интеллектуальных механизмов управления.

Далее каждая из составляющих плоскостей рассмотрена подробно.

Плоскость передачи данных

В KDN плоскость передачи данных, обеспечивая передачу информации между конечными устройствами, выполняет функции, аналогичные традиционным сетям и SDN. Однако существует и несколько ключевых отличий. Прежде всего, настройка и отслеживание состояния плоскости передачи данных в KDN выполняются плоскостью мониторинга, тогда как реализация политик и пра-

вил находится под управлением плоскости контроля. Из-за этого в KDN, по сравнению с SDN, значительно больший объем данных передается в единую контрольную плоскость [6].

Эта увеличенная нагрузка требует от устройств маршрутизации в KDN (физических или виртуальных коммутаторов, маршрутизаторов, точек доступа и т. д.) больше пропускной способности и вычислительной мощности, нежели чем от схожих устройств в SDN. Базовая структура такого устройства, а также взаимодействия плоскости данных с другими элементами системы, представлены на рисунке 2.

Следует отметить, что устройства маршрутизации в идеальных KDN не обладают автономным интеллектом для принятия решений, так как все правила формируются и задаются логически централизованным контроллером. Несмотря на то, что данный подход представляется оптимальным с теоретической точки зрения, его реализация в легаси-системах (унаследованных системах) с устаревшей инфраструктурой вызывает значительные трудности.

Гибридная система, которая упрощает интеграцию KDN в существующие системы, должна содержать дополнительный локальный агент управления. Функциональная структура такого узла в KDN показана на рисунке 3.

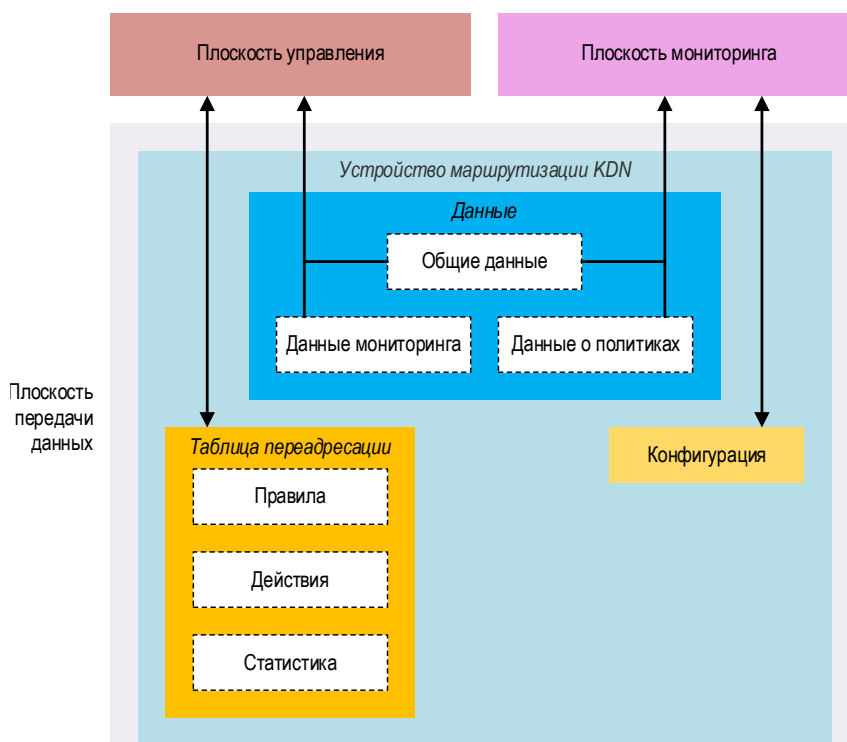


Рис. 2. Схематическая блок-диаграмма плоскости передачи данных и устройства маршрутизации KDN

Fig. 2. Schematic Block Diagram of the Data Plane and KDN Forwarding Device

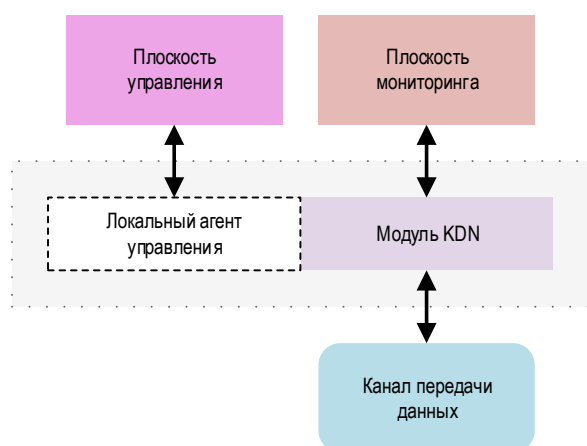


Рис. 3. Функциональная структура гибридного узла в KDN

Fig. 3. The Structure of a Hybrid Node in KDN

Локальный агент управления осуществляет обработку пакетов с использованием управляющих сообщений, полученных от плоскости управления, в то время как модуль KDN настраивает коммутатор на основе конфигураций, поступающих из плоскости мониторинга. Кроме того, модуль KDN собирает и передает данные обратно в упомянутые выше плоскости.

Если управляющий канал (канал, соединяющий гибридный узел и плоскость управления) доступен, тогда локальный агент находится в пассивном состоянии и перенаправляет полученные управляющие сообщения от внешнего контроллера в модуль KDN внутри узла. А в случае потери связи с централизованным контроллером, локальный агент управления берет на себя роль активного резервного контроллера, который переключается на использование заранее определенных политик и правил. В таком случае нормальная работа узла, включая отправку и прием данных, а также перенаправление пакетов, поступающих от других узлов, может быть обеспечена через использование канала передачи данных.

Плоскость управления

Плоскость управления обеспечивает маршрутизацию и обработку данных с использованием знаний, полученных из сетевых данных. В отличие от традиционных сетей, где контроль децентрализован и ограничен физической инфраструктурой, или SDN, где управление логически централизовано, контрольная плоскость KDN интегрирует элементы ML и анализа данных для создания интеллектуальных моделей управления сети.

Плоскость управления, состоящая из одного или более SDN-контроллера, координирует передачу информации между плоскостью данных и другими элементами, опираясь на глобальное представление о топологии и состоянии сети для эффективного управления потоками информации. Данные,

поступающие в плоскость управления, через плоскость знаний подвергаются высокоуровневому анализу с применением методов ML, что позволяет более точно прогнозировать поведение сети и оптимизировать ее производительность. Плоскость знаний анализирует телеметрию, исторические данные и текущие сетевые параметры для формирования стратегий и политик управления. Эти политики передаются обратно в плоскость управления, которая применяет их для обновления правил маршрутизации, управления трафиком и распределения ресурсов.

Плоскость управления KDN может быть реализована в рамках различных архитектур: централизованной, распределенной или гибридной. Следует отметить, что все узлы в представленных архитектурах должны соответствовать ранее описанной гибридной структуре узлов (см. рисунок 3).

Централизованная архитектура. Данная модель, представленная на рисунке 4а, сохраняет основную концепцию архитектуры SDN с логически централизованной плоскостью управления, где все действия узлов полностью управляются SDN-контроллером [7]. Такая архитектура отличается простотой, удобством управления и высокой гибкостью. Однако ее недостатком является повышенная задержка, обусловленная удаленной связью между узлами и контроллером, логически и физически отделенным от плоскости данных. Хотя такая схема эффективна для небольших сетей, ее производительность снижается по мере увеличения масштаба и нагрузки инфраструктуры. Контроллер может столкнуться с нехваткой ресурсов при обработке большого числа запросов от устройств, что приведет к ухудшению показателей. Кроме того, наличие единой точки отказа делает модель уязвимой и ограничивает ее надежность.

Распределенная архитектура. Традиционная архитектура SDN с логически централизованной плоскостью управления, разделяющей плоскости управления и данных, демонстрирует низкую производительность в междоменных, особенно в гетерогенных сетях. Для улучшения управления в таких сетях была предложена архитектура с логически распределенной плоскостью управления, где контроль распределяется между несколькими контроллерами [8], как показано на рисунке 4b. В данной модели, вместо централизованного SDN-контроллера, элементы передачи взаимодействуют с локальным контроллером, который обеспечивает управление в пределах своей зоны ответственности. В отличие от логически централизованной архитектуры изменения в сети не транслируются автоматически всем контроллерам – обновления передаются только при необходимости, например, при междоменных сервисах, когда контроллеры взаимодействуют друг с другом напрямую для обмена информацией.

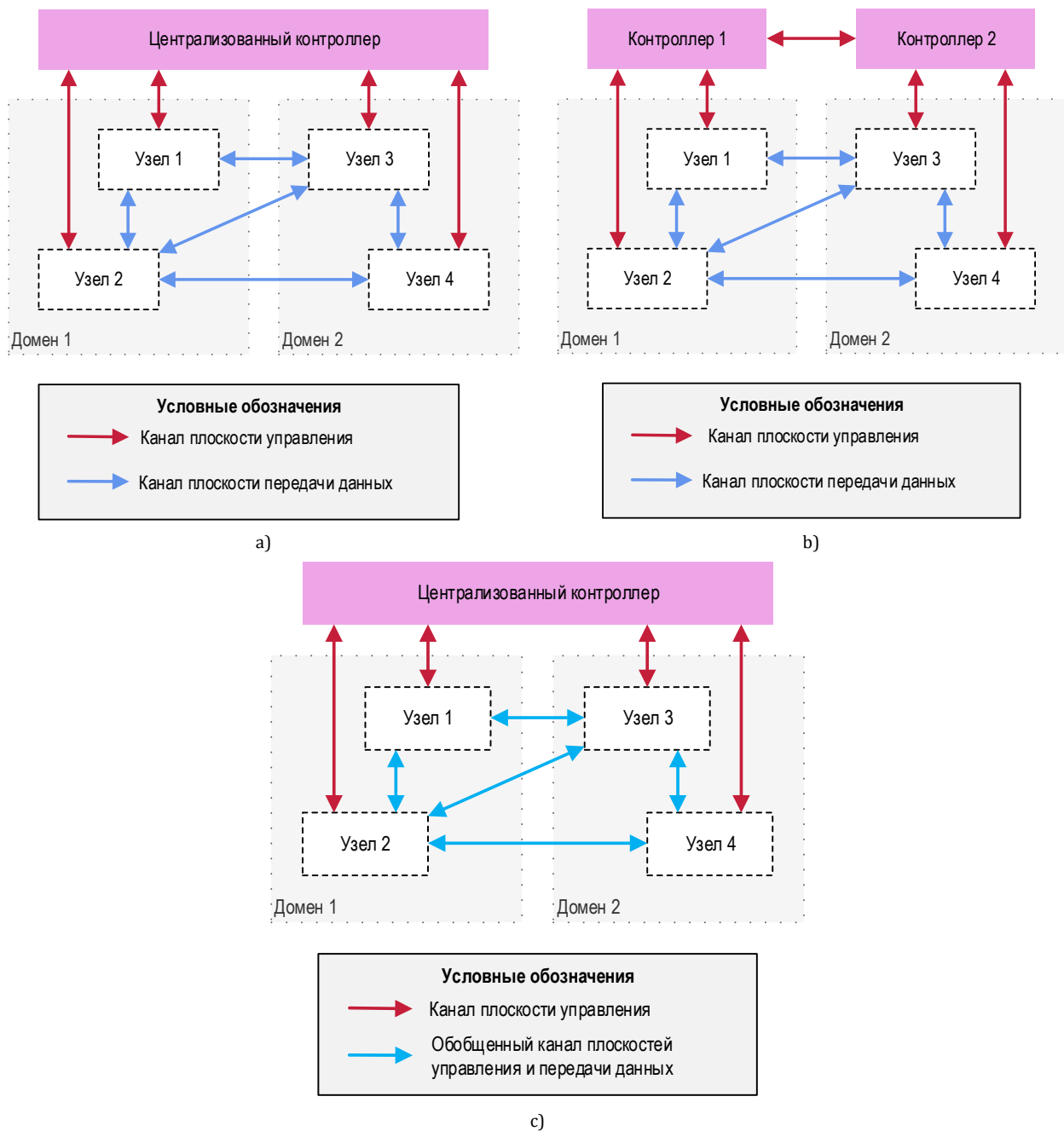


Рис. 4. Архитектура плоскости управления KDN: централизованная (а), распределенная (б) и гибридная (с)

Fig. 4. Architecture of the KDN Control Plane: Centralized (a), Distributed (b) and Hybrid (c)

Гибридная архитектура. Гибридная архитектура управления сочетает в себе черты централизованной и распределенной архитектуры, а контроль над сетью осуществляется совместно с плоскостями данных и управления [9]. В зависимости от условий сети, гибридная архитектура может гибко регулировать тип управления, автоматически переключая узлы и контроллеры (рисунок 4с). Коммуникация плоскости управления включает как прием управляющих сообщений от централизованного контроллера, так и обмен сообщениями

между узлами. Элементы плоскости передачи данных также участвуют в принятии решений и управлении сетью, что обеспечивает более высокую масштабируемость и устойчивость к сбоям.

Наряду с централизованной, распределенной и гибридной архитектурами существуют и альтернативные подходы к реализации управляющей плоскости KDN. Такие архитектуры могут включать многоуровневую организацию управления или механизмы контекстно-ориентированного взаимо-

действия, что позволяет более эффективно координировать обмен данными и поддерживать высокую производительность сети в условиях масштабирования и динамических изменений топологии.

Плоскость мониторинга

Основной целью плоскости мониторинга, которая функционирует параллельно с контроллером (управляющей плоскостью) в архитектуре KDN, является наблюдение за сетевыми устройствами, сбор данных и информации, обнаружение и

настройка параметров качества обслуживания (QoS, аббр. от англ. Quality of Service) в сети.

Как уже упоминалось ранее, в KDN функции мониторинга отделены от логики управления, и для стандартных сценариев (сбора данных, сетевого мониторинга и настройки) используются отдельные протоколы. Это упрощает устранение неисправностей в случае сбоев по сравнению с традиционной архитектурой SDN.

Схема плоскости мониторинга в архитектуре KDN представлена на рисунке 5.

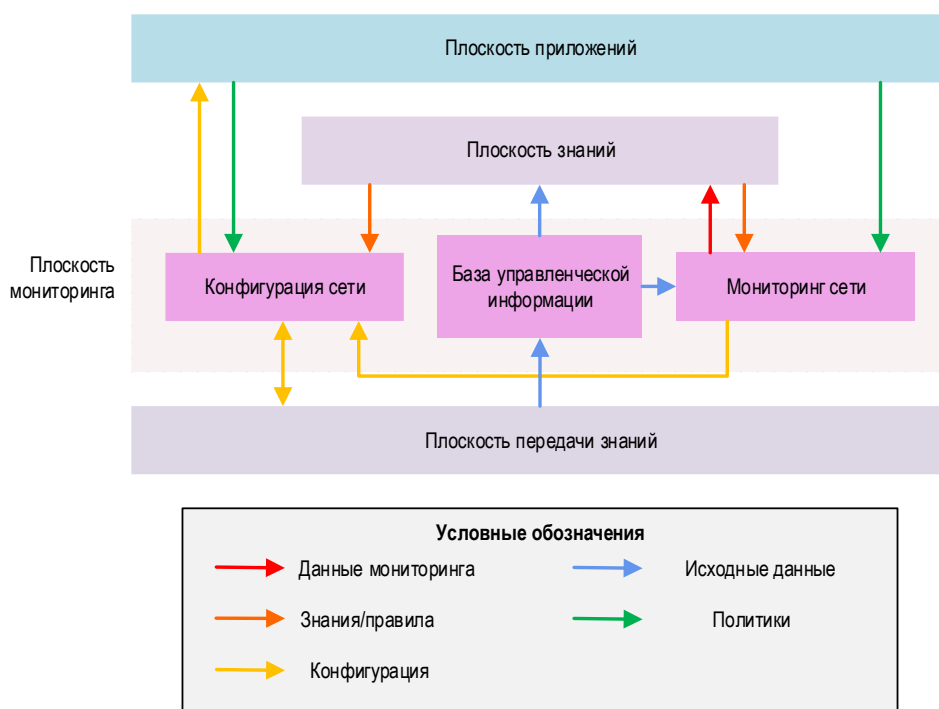


Рис. 5. Схематическая блок-диаграмма плоскости мониторинга KDN

Fig. 5. Schematic Block Diagram of the KDN Management Plane

В архитектуре KDN плоскость управления предоставляет унифицированный интерфейс для сбора данных, конфигурирования устройств и мониторинга сети. База управленческой информации отвечает за сбор и управление сетевыми данными, необходимыми для выполнения задач мониторинга и генерации знаний. Исходные данные из плоскости данных могут либо использоваться непосредственно в модуле мониторинга, либо передаваться в плоскость знаний для их дальнейшей обработки и генерации знаний.

Модуль сетевого мониторинга генерирует конфигурационные правила, сравнивая имеющиеся данные или знания с политиками сети. Анализ состояния сети может осуществляться как по запросу, так и непрерывно, а его результаты также могут направляться в плоскость знаний для генерации новых правил или знаний. Важно отметить, что, хотя это не показано на схеме, если ре-

зультаты мониторинга передаются в плоскость знаний, то могут быть созданы дополнительные знания или правила, которые, в свою очередь, могут повлиять на политики плоскости приложений.

Сгенерированные модулем мониторинга конфигурационные правила передаются в модуль конфигурации сети для непосредственной настройки элементов плоскости данных. Эти правила также могут использоваться для принятия решений по настройке сети, с учетом правил или знаний, полученных из плоскости знаний, и политик конфигурации, поступивших из прикладной плоскости.

Плоскость знаний

Плоскость знаний представляет собой логический уровень, отвечающий за генерацию, обработку и распространение знаний. Он помогает сетевым операторам (людям-администраторам, про-

граммным системам) эффективнее управлять поведением сети и точнее адаптироваться к изменяющимся условиям. В рамках данной плоскости используются декларативные (правила), процедурные (процессы) и контекстуальные знания (информация об окружающей среде и условиях работы сети) [10]. Они поступают из разных источников (сетевых устройств, сенсоров и т. д.), и анализируются для получения знаний и принятия интеллектуальных решений. Плоскость знаний помогает сетевым операторам заранее выявлять и устранять проблемы, оптимизировать поведение сети под изменяющиеся требования и внедрять усовершенствованные меры безопасности для защиты от угроз.

Плоскость знаний из-за постоянной обработки потока данных имеет наиболее сложную архитектуру среди всех плоскостей в KDN. Как показано на рисунке 6, плоскость знаний состоит из трех подуровней:

1) подуровень генерации знаний – генерирует описательные знания из исходных данных с использованием моделей генерации знаний;

2) подуровень композиции знаний – выявляет взаимосвязи между различными фрагментами знаний и создает комплексные знания с помощью редактора онтологий, которые после могут использоваться и для формирования правил, учитывающих, в том числе, и намерения пользователя;

3) подуровень управления и распределения знаний – отвечает за хранение информации, управление и обмен необработанными данными в пределах плоскости знаний, а также сформированными и сгенерированными знаниями, правилами и управляющими сообщениями между другими плоскостями.

Подуровни плоскости знаний детально описаны в следующих подразделах.

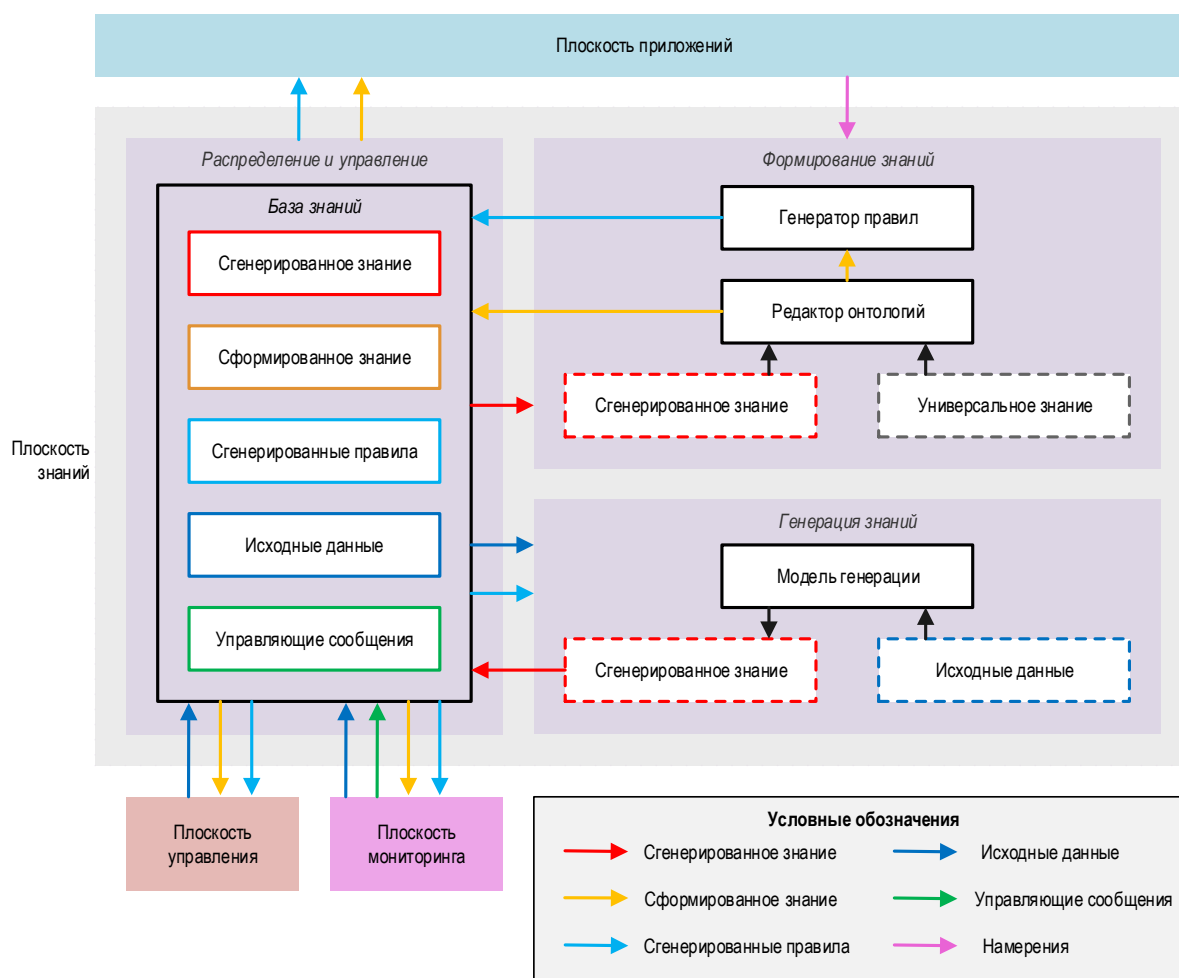


Рис. 6. Схематическая блок-диаграмма плоскости знаний KDN

Fig. 6. Schematic Block Diagram of the KDN Knowledge Plane

Подуровень генерации знаний

Подуровень генерации знаний выполняет извлечение описательных знаний из необработанных данных с использованием моделей генерации знаний. Этот процесс позволяет значительно сократить объем информации, передаваемой контроллеру, так как описательные знания являются более компактной и структурированной формой исходных данных. Исходные данные, собираемые на плоскостях управления и мониторинга, подаются в качестве входных данных для генераторов, в роли которых могут применяться как эвристические модели, так и методы ML.

Эвристические модели базируются на математических методах, позволяющих описывать исходные данные и выявлять внутренние корреляции между ними [11]. Они отличаются простотой реализации, низкой вычислительной сложностью и пригодны для задач, где требуется оперативная обработка информации в реальном времени. Однако возможности этих моделей ограничены, поскольку они плохо справляются с нелинейными и сложными зависимостями в данных, требуют высокой точности исходных данных и ручной настройки параметров. Такие модели не адаптированы для работы в условиях динамичных и изменчивых сетевых сред, но, несмотря на эти ограничения, эвристические методы могут быть полезны в рамках конвейерной обработки данных, где они дополняют более сложные методы ML.

Методы ML позволяют автономно анализировать данные и генерировать знания, что делает их особенно эффективными для сложных и динамичных сетевых сред. Для повышения точности алгоритмов, перед обучением таких моделей данные предварительно обрабатываются для выделения высокоуровневых признаков [12]. Основным преимуществом методов ML является их способность адаптироваться к изменениям в поведении сети, выявлять сложные закономерности и работать с большими объемами многомерных данных. Эти алгоритмы обеспечивают более высокую точность анализа и прогнозирования по сравнению с эвристическими подходами. В рамках генерации знаний могут применяться следующие типы ML.

1) *Методы с учителем (Supervised Learning)*. Этот подход заключается в обучении алгоритмов на размеченных данных, что позволяет делать прогнозирование на основе уже известных примеров [13]. Для достижения точных результатов входные данные должны быть четко размечены и отражать правильные ответы. Алгоритмы с учителем делятся на две основные категории: классификационные и регрессионные модели. Классификация используется для задач с дискретными выходными переменными, которые можно легко категоризо-

вать. Регрессия применяется для предсказания непрерывных числовых значений, основываясь на наблюдениях из прошлого. В контексте KDN наиболее распространенные примеры применения методов с учителем включают прогнозирование отказов соединений, определение требований к пропускной способности и оценку QoS.

2) *Методы без учителя (Unsupervised Learning)*. Этот подход обучает алгоритмы на неразмеченных данных, не обеспечивая четкими примерами для обучения. Основная идея состоит в том, чтобы предоставить системе большие объемы разнообразных данных и позволить ей самостоятельно извлекать полезные знания и выявлять скрытые закономерности [14]. Одним из наиболее популярных методов является кластеризация, которая группирует неразмеченные данные в схожие группы (кластеры). Другим примером является обнаружение аномалий – метод идентификации редких событий или наблюдений, которые значительно отличаются от большинства данных. В KDN эти методы применяются для кластеризации трафика и базовых станций, а также для мониторинга поведения устройств и сетей.

3) *Глубокое обучение (DL)* – раздел ML, основанный на использовании искусственных нейронных сетей с множеством скрытых слоев. Эти сети способны автоматически извлекать иерархические представления данных, что позволяет им решать комплексные и нелинейные задачи [15]. Алгоритмы DL обучаются путем настройки весов связей между нейронами через обратное распространение ошибки, таким образом DL может в процессе итераций самостоятельно вырабатывать условия для принятия корректных решений. Наиболее распространенные алгоритмы DL и области их применения представлены в таблице 1.

ТАБЛИЦА 1. Алгоритмы DL, применяемые в KDN

TABLE 1. Deep Learning Algorithms Used in KDN

Алгоритм	Ключевое применение
Долгая краткосрочная память (LSTM, аббр. от англ. Long Short-Term Memory)	Прогнозирование трафика, отслеживание мобильности, обнаружение неисправностей
Генеративно-состязательная сеть (GAN, аббр. от англ. Generative Adversarial Network)	Генерация синтетического трафика, противостояние угрозам безопасности, оптимизация QoS
Метод опорных векторов (SVM, аббр. от англ. Support Vector Machine)	Обнаружение аномалий, оптимизация маршрутизации, прогнозирование QoS

Тем не менее, описанные методы ML и DL зачастую требуют значительных вычислительных ресурсов и высококачественных размеченных наборов данных для обучения, которые в хаотичной сетевой системе могут быть недоступны. Кроме того, алгоритмы могут быть непрозрачными (проблема

«черного ящика», особенно характерная для DL), что усложняет интерпретацию процесса принятия решений.

Во многих случаях оптимальным решением может стать комбинирование обоих подходов. Эвристические модели могут выступать в качестве базовой или резервной системы, тогда как методы ML берет на себя более сложные задачи. Например, эвристическая модель может заниматься постоянной классификацией трафика, в то время как алгоритмы ML занимаются обнаружением аномалий или прогнозированием маршрутизации. Гибридные подходы объединяют сильные стороны обоих методов, что позволяет достичь баланса между эффективностью и адаптивностью.

Подуровень композиции знаний

В подуровне композиции знания, сгенерированные на подуровне генерации, проходят дальнейший анализ или объединяются для создания новых, комплексных знаний. Центральным элементом композиции знаний является редактор онтологий, который комбинирует универсальные, уже существующие, знания с множеством новых, сгенерированных знаний. Редактор онтологий представляет собой программное приложение, предназначенное для создания и поддержания онтологий, которые используются для определения свойств и иной информации с помощью онтологического языка.

Кроме того, подуровень композиции знаний отвечает на запросы и намерения плоскости приложений, сравнивая их с уже составленными правилами и политиками. Эта информация используется генератором правил для создания новых правил, которые применяются другими плоскостями сети.

Одним из динамичных методов генерации правил является обучение с подкреплением (Reinforcement Learning). Алгоритмы этого типа используют агентов (программные модули), которые самостоятельно обучаются выполнению задач через многократные пробные взаимодействия с динамичной средой. Основная цель агента – без явного программирования максимизировать вознаграждения и минимизировать штрафы, анализируя свои действия в конкретных ситуациях. Основные области применения этого метода в KDN включают маршрутизацию с учетом QoS, адаптивное распределение ресурсов, а также мониторинг и управление в сетях [16].

Подуровень управления и распределения знаний

Управление знаниями в контексте KDN представляет собой процесс сбора, хранения, обмена и использования знаний и правил в сетевой среде. Основная цель управления знаниями связана с по-

вышением производительности сети за счет оптимизации процесса обеспечения необходимых знаний и правил заинтересованным сторонам. Данная плоскость состоит из базы знаний, которая отвечает за хранение и распространение следующих элементов:

- знаний, сгенерированных на основе моделей генерации знаний;
- знаний, сформированных редакторами онтологий;
- правил, созданных генераторами правил;
- необработанных данных и управляющих сообщений, поступающих из сети.

Поскольку подуровень управления и распределения знаний собирает данные из всех частей сети, чтобы обеспечить совместимость различных систем KDN, необходимо наличие стандартов для их хранения, обмена и использования. Кроме того, в условиях ограниченных ресурсов сети должны быть установлены правила передачи трафика, связанного со знаниями (сгенерированные, сформированные знания или правила, созданные на их основе). Одной из задач подуровня распределения знаний является формирование правил передачи такого трафика как внутри плоскости знаний, так и его маршрутизация плоскостям управления, мониторинга и приложений.

Важно отметить, что плоскость знаний является центральным компонентом KDN, играющим ключевую роль во всей архитектуре. Учитывая ее критическую функцию, избыточность плоскости знаний имеет первостепенное значение для обеспечения устойчивости и надежности всей сети. Это подчеркивает преимущества гибридной архитектуры (см. рисунок 4с), где наличие избыточных узлов плоскости знаний на разных уровнях позволяет обеспечить плавное переключение и непрерывность работы даже в случае аппаратных или программных сбоев. Такой подход предотвращает возникновение единой точки отказа, которая могла бы нарушить критически важные функции, такие как маршрутизация трафика, распределение ресурсов и устранение неполадок в сети.

Плоскость приложений

Плоскость приложений является верхним уровнем в архитектуре KDN и отвечает за предоставление высокоуровневого представления сети, которое тесно связано с бизнес-целями и задачами пользователей, которые выражаются в форме намерений.

Плоскость приложений в KDN спроектирована для понимания требований приложений и динамической оптимизации их политик на основе знаний о состоянии сети. Это позволяет учитывать измене-

ния в сетевых условиях и эффективно адаптировать работу приложений к текущим потребностям и ограничениям. Кроме того, плоскость приложений предоставляет централизованную точку управления сетевыми политиками – администраторы могут задавать политики (которые относятся как к одному приложению, так и к их группе) и одновременно применять эти политики по всей сети.

Архитектура плоскости приложений и ее связь с другими элементами архитектуры KDN представлены на рисунке 7. Как видно из рисунка, плоскость приложений состоит из двух основных подуровней: *подуровня определения* намерений и политик, которая вложена в *подуровень обновления* намерений и политик.

Цель подуровня определения заключается в том, чтобы администраторы сети могли задавать намерения и политики вручную. После того как намерения и политики определены, их можно динамически развивать и изменять с использованием подуровня обновления намерений и политик. Для этого используются правила и знания, поступающие из плоскости знаний, конфигурации из плоскости мониторинга и сетевая информация (данные о топологии и статистика сети) полученная из плоскости управления.

Следует отметить, что в подуровне обновления намерений и политик имеется механизм правил, который обрабатывает данные из других плоскостей для принятия решений или создания инструкций по обновлению политик.

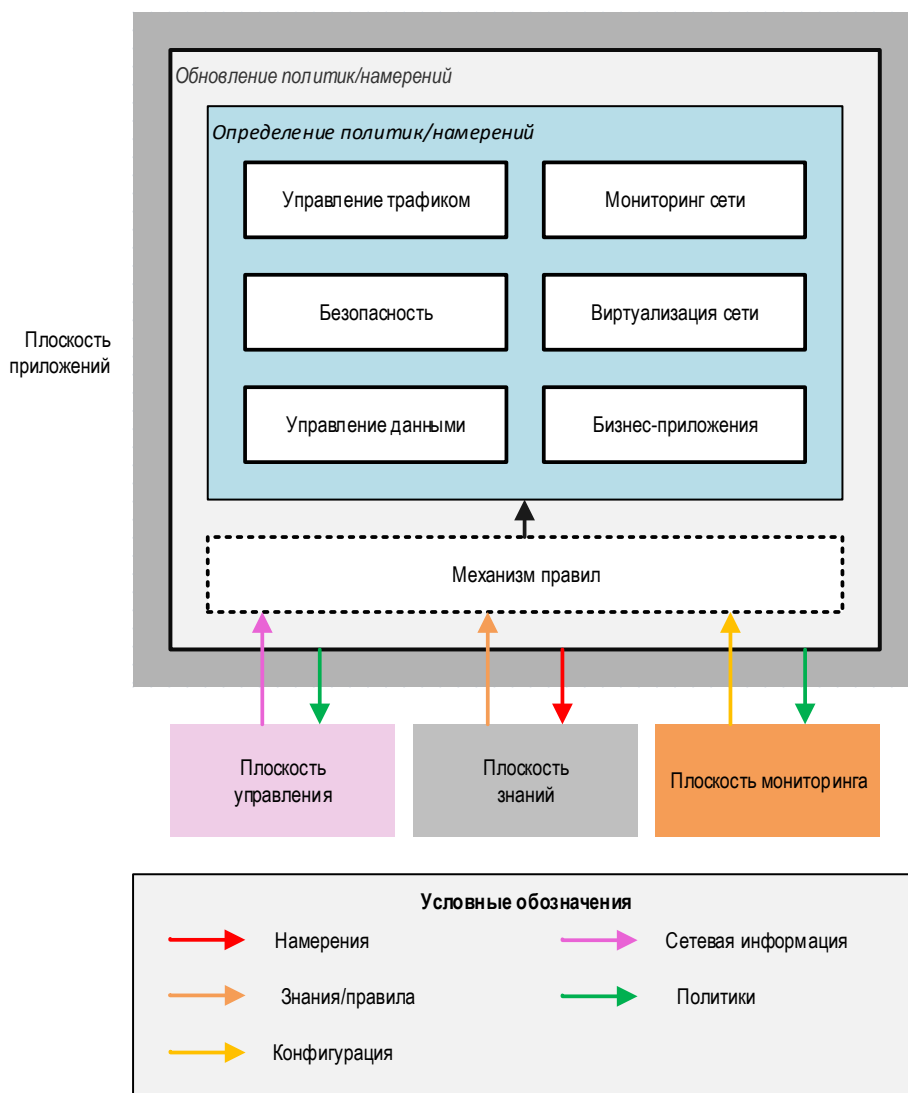


Рис. 7. Схематическая блок-диаграмма плоскости приложений KDN

Fig. 7. Schematic Block Diagram of the KDN Application Plane

В KDN возможно выделить шесть основных типов приложений.

1) *Управление трафиком*. Приложения отвечают за процесс оптимизации производительности и использования сетевых ресурсов для обеспечения эффективного потока данных и могут достигать этой цели различными способами [17], включая использование политик инженерии трафика, маршрутизацию, балансировку нагрузки и предоставление QoS.

2) *Мониторинг сети*. Приложения определяют политики для различных задач мониторинга, таких как управление отказами, управление мобильностью и управление энергопотреблением. Эти задачи реализуются в плоскости управления и для принятия решений в различных сценариях и учитывают и политики приложений, и актуальные данные о состоянии сети.

3) *Приложения безопасности*. Позволяют задавать политики для контроля доступа, шифрования, настройки межсетевых экранов, обнаружения и предотвращения атак, аномалий и вторжений. Такие приложения обеспечивают быстрое и эффективное выполнение политик безопасности, помогая гарантировать целостность, доступность и конфиденциальность сетевых ресурсов.

4) *Виртуализация сети*. Процесс заключается в создании нескольких логических сетей на физической инфраструктуре с использованием технологии разделения сети (Network Slicing) [18]. Это позволяет нескольким сетям сосуществовать и использовать одни и те же физические ресурсы, при этом каждая логическая сеть функционирует как отдельная сущность со своими политиками и конфигурациями. Виртуализация сети может осуществляться на уровне всей сети (Network-Level Slicing) или на уровне потоков (Flow-Level Slicing).

5) *Управление данными*. Эта категория включает приложения, работающие с объемами данных:

- *большие данные (Big Data)* работают с крупными и сложными наборами данных (по объему, скорости поступления и разнообразию), которые сложно обрабатывать традиционными методами [19]; KDN может решать проблемы с доставкой и обработкой данных в облачных центрах;

- *облачные вычисления* обеспечивают предоставление вычислительных ресурсов, включая серверы, хранилища, приложения и сервисы, по запросу через интернет; политики виртуализации сетевых функций позволяют эффективно использовать облачные ресурсы;

- *управление дата-центрами* (практика подключения множества серверов, хранилищ и других вычислительных ресурсов внутри одного центра обработки данных); такие приложения помогают минимизировать задержки, оптимально использовать ресурсы и повышать общую эффективность.

6) *Бизнес-приложения*. Приложения, ориентированные на получение прибыли и определяют политики, которые помогают организациям достигать бизнес-цели.

Дальнейшее развитие сетей управления на основе знаний

KDN на текущий момент находятся на начальном этапе своего развития, поэтому большая часть их концепции базируется на теоретических основах, а не на полностью реализованных практических системах. Это означает, что область KDN обладает значительным потенциалом для дальнейших исследований. Однако даже на данном этапе KDN представляют собой перспективную парадигму, вводящую в управление сетями знание-ориентированный подход. В отличие от традиционных сетей и SDN, KDN акцентирует внимание на интеллектуальности и автоматизации. Этот этап становления предоставляет исследователям уникальную возможность анализировать и формировать направление развития KDN, которое в перспективе станет неотъемлемой частью сетей следующего поколения.

По мере развития KDN возникают перспективы как для выявления преимуществ, так и для понимания возможных ограничений. Это позволяет уже на ранних этапах оценить потенциал технологии и повлиять на ее развитие. Так, например, *основными достоинствами KDN* являются следующие характеристики.

1) *Повышенная автоматизация и интеллектуальность*. KDN трансформирует управление сетями за счет внедрения ML и искусственного интеллекта (AI, аббр. от англ. Artificial Intelligence), что обеспечивает высокий уровень автоматизации и интеллектуальное принятие решений. Это позволяет в реальном времени адаптировать сеть, проводить предиктивный анализ и устранять неисправности, значительно снижая необходимость в ручном управлении. Используя знания, полученные из сетевых данных, системы KDN так же могут самоорганизовываться, адаптироваться к изменениям условий и внедрять эффективные решения, закладывая основу для полностью автономных сетей.

2) *Упрощенное управление сетью*. KDN упрощает управление сетью благодаря разделению плоскостей управления, данных и знаний, что создает единую модульную архитектуру для мониторинга, настройки и оптимизации сетевых операций. Такой подход позволяет операторам эффективно управлять сложными системами, минимизируя потребность в детальной ручной настройке. Плоскость знаний дополнительно автоматизирует рутинные задачи управления и динамически адаптируется к изменяющимся требованиям сети.

3) *Улучшение производительности сети и QoS.* Возможности KDN по оптимизации распределения ресурсов и управления трафиком напрямую влияют на повышение производительности сети. Используя знания в реальном времени и предиктивную аналитику, KDN обеспечивает снижение задержек, улучшение качества обслуживания и повышение надежности. Эти улучшения способствуют оптимизации пользовательского опыта, делая сеть более отзывчивой к требованиям конечных пользователей при сохранении высокой производительности.

4) *Снижение затрат на рабочую силу.* Автоматизация в KDN значительно снижает затраты на управление сетью, так как маршрутизация трафика, распределение ресурсов и обнаружение неисправностей выполняются автономно. Это уменьшает потребность в высококвалифицированных сетевых инженерах и позволяет организациям более эффективно распределять свои ресурсы.

5) *Повышение безопасности.* Интеграция AI и ML в KDN усиливает сетевую безопасность за счет способности в реальном времени обнаруживать и устранять опасности. Используя продвинутую аналитику, системы KDN могут проактивно выявлять уязвимости и прогнозировать потенциальные угрозы, создавая мощный механизм защиты. Централизованная плоскость знаний обеспечивает согласованность мер безопасности во всей сети, снижая риск изолированных уязвимостей.

С другой стороны, уже можно выделить и *недостатки KDN.*

1) *Проблемы с данными.* KDN в значительной степени зависит от больших объемов высококачественных данных для обучения моделей ML. Это создает серьезные сложности, связанные с хранением, обработкой и управлением информацией. Обеспечение однообразности и целостности знаний в сети может быть сложной задачей, особенно при работе с разнообразными источниками и форматами данных. Необходимость анализа данных в реальном времени, требуя устойчивой инфраструктуры для поддержки таких операций, так же усложняет процесс.

2) *Интеграция с существующими системами.* Переход от традиционных сетей или SDN к KDN создает проблемы совместимости с устаревшими системами. Во многих существующих сетях отсутствует инфраструктура, необходимая для интеграции плоскости знаний, что приводит к значительным затратам на модернизацию. Эти препятствия могут задерживать внедрение и увеличивать сложность развертывания решений KDN в реальных условиях.

3) *Интероперабельность и сложности внедрения.* Обеспечение интероперабельности устройств

и систем в KDN (как внутри одной сети, так и между двумя сетями) затруднено из-за многоплоскостной архитектуры и множества протоколов. С ростом сети поддержание согласованности и обмен данными становятся сложнее, а вычислительные требования ML-решений возрастают, что создает проблемы с масштабируемостью. Внедрение KDN требует значительных финансовых вложений в инфраструктуру, программное обеспечение и квалифицированный персонал. Постоянное обновление моделей ML и адаптация к технологиям повышают расходы, что затрудняет доступ для компаний с ограниченным бюджетом.

4) *Недостаток прозрачности.* Зависимость KDN от ML и AI часто приводит к недостаточной прозрачности процессов принятия решений. Операторам сети может быть сложно понять или проверить логику некоторых автоматических действий, особенно в случаях применения комплексных алгоритмов ML. Такая непрозрачность снижает уровень доверия и усложняет отладку или аудит, что создает серьезные препятствия для широкого внедрения.

Несмотря на это, траектория развития KDN предполагает действительно возможный переход к более адаптивному, предиктивному и контекстно-ориентированному сетевым средам. Развитие распределенного интеллекта в архитектурах KDN может обеспечить возможность совместной работы в реальном времени между периферийными и центральными системами, создавая сети, которые не только реагируют на текущие запросы, но и предвосхищают их. Кроме того, интеграция квантовых вычислений и более совершенных AI / ML-архитектур обещает значительно расширить возможности KDN, позволяя обрабатывать огромные объемы данных с беспрецедентной скоростью и точностью.

В развитии KDN выделяют *три ключевых направления.*

1) *Оптимизация существующих алгоритмов ML.* Развитие KDN напрямую зависит от совершенствования существующих алгоритмов ML, а также от их адаптации к специфическим требованиям динамичных сетевых сред. Индивидуально настроенные модели ML, приспособленные к различным сценариям, обеспечат более быстрое принятие решений и точное распределение ресурсов. Разработка новых алгоритмов с использованием таких передовых методов, как федеративное обучение [20] (децентрализованное / распределенное обучение на множестве физических платформ) или объяснимый AI [21] (системы и модели, способные объяснять свои действия), повысит адаптивность и прозрачность систем KDN, обеспечивая соответствие изменяющимся требованиям к производительности и этическим стандартам.

2) *Интеграция с существующими системами.* Критически важным направлением дальнейшего развития KDN является обеспечение плавной интеграции с устаревшими системами для минимизации сбоев и сокращения затрат на модернизацию. Инновации в гибридных архитектурах, связывающих традиционные сетевые устройства с плоскостью знаний KDN, могут сохранить существующую инфраструктуру при добавлении новых функций. Это позволит организациям внедрять KDN, сохраняя при этом непрерывность операций, обеспечивая экономичность модернизационных процессов.

4) *Оптимизация для самоорганизующихся сетей.* Будущее KDN так же тесно связано с инновациями в создании самоорганизующимися сетями, которые могут автономно организовываться, переконфигурироваться и оптимизироваться в ответ на изменения условий. Улучшение алгоритмов самообучения и использование принципов роевого интеллекта могут позволить сетям функционировать с минимальным участием человека, динамически адаптируясь к изменениям трафика, доступности ресурсов и потребностям пользователей для поддержания оптимальной производительности [22].

Заключение

В рамках проведенного анализа выявлено, что архитектура KDN предоставляет широкие возмож-

ности для автоматизации и повышения адаптивности сетей за счет интеграции плоскости знаний, способной в реальном времени генерировать и применять интеллектуальные стратегии управления. Рассмотрены преимущества использования KDN, такие как снижение операционных затрат, улучшение качества обслуживания, а также упрощение управления сетевой инфраструктурой в условиях растущей сложности сетевых сценариев.

Рассмотренная архитектура KDN, включающая плоскости знаний, мониторинга и управления, демонстрирует высокую степень автоматизации, гибкости и адаптивности. Эти свойства позволяют сетям, определяемым знаниями, не только повышать эффективность передачи данных, но и обеспечивать устойчивость и безопасность сетей в динамичных условиях. Интеграция методов ML и DL в рамках плоскости знаний открывает возможности для интеллектуального управления, автономного принятия решений и повышения QoS.

Дальнейшие исследования в этой области должны быть направлены на оптимизацию взаимодействия между различными плоскостями KDN, разработку методов интеграции с существующими сетевыми инфраструктурами и преодоление ограничений, связанных с необходимостью значительных вычислительных ресурсов. Реализация этих задач обеспечит более широкое внедрение KDN в практику построения сетей нового поколения.

Список источников

1. Zoraida B.S.E., Ganesan I. A Comparative Study on Software-Defined Network with Traditional Networks // TEM Journal. 2024. Vol. 13. Iss. 1. PP. 167–176. DOI:10.18421/TEM131-17
2. Hakiri A., Gokhale A., Berthou P., Schmidt D.C., Gayraud T. Software-defined Networking: Challenges and Research Opportunities for Future Internet // Computer Networks. 2014. Vol. 75. Part A. PP. 453–471. DOI:10.1016/j.comnet.2014.10.015
3. Ashtari S., Zhou I., Abolhasan M., Shariati N., Lipman J., Ni W. Knowledge-defined networking: Applications, challenges and future work // Array. 2022. Vol. 14. P. 100136. DOI:10.1016/j.array.2022.100136
4. Jarrahi M.H., Askay D., Eshraghi A., Smith P. Artificial intelligence and knowledge management: A partnership between human and AI // Business Horizons. 2023. Vol. 66. Iss. 1. PP. 87–99. DOI:10.1016/j.bushor.2022.03.002
5. Srinivas T.A., Donald A., Thippanna G., Kousar M., Priya A. From Control to Chaos: The Dynamic SDN Control Plane // International Journal of Advanced Research in Science, Communication and Technology. 2023. Vol. 3. Iss. 2. PP. 494–502. DOI:10.48175/IJARSC-8527
6. Mestres A., Rodriguez-Natal A., Carner J., Barlet-Ros P., Alarcón E., Solé-Simó M., et al. Knowledge-Defined Networking // ACM SIGCOMM Computer Communication Review. 2017. Vol. 47. Iss. 3. PP. 2–10. DOI:10.1145/3138808.3138810
7. Li W. Explore the Evolution of Computer Network Architecture // International Journal of Religion. 2024. Vol. 5. № 11. PP. 9034–9054. DOI:10.61707/ahr2da31
8. Das R., Pohrmen F., Maji A., Saha G. FT-SDN: A Fault-Tolerant Distributed Architecture for Software Defined Network // Wireless Personal Communications. 2020. Vol. 114. PP. 1045–1066. DOI:10.1007/s11277-020-07407-x
9. Wang T., Su Z., Hamdi M. Rethinking the Data Center Networking: Architecture, Network Protocols, and Resource Sharing // IEEE Access. 2014. Vol. 2. PP. 1481–1496. DOI:10.1109/ACCESS.2014.2383439
10. Agoulmine N. Chapter 1. Introduction to Autonomic Concepts Applied to Future Self-Managed Networks // In: Walsh S.M., Strano M.S. (ed.) Autonomic Network Management Principles. Academic Press, 2011. PP. 1–26. DOI:10.1016/B978-0-12-382190-4.00001-2
11. Silver E. An overview of heuristic solution methods // Journal of The Operational Research Society. 2004. Vol. 55. Iss. 9. PP. 936–956. DOI:10.1057/palgrave.jors.2601758
12. Oladipupo T. Machine Learning Overview // In: Zhang Y. (ed.) New Advances in Machine Learning. InTech, 2010. PP. 9–18. DOI:10.5772/9374
13. Nasteski V. An overview of the supervised machine learning methods // Horizons. 2017. Vol. 4. PP. 51–62. DOI:10.20544/HORIZONS.B.04.1.17.P05

14. Yazici İ., Shayea I., Din J. A survey of applications of artificial intelligence and machine learning in future mobile networks-enabled systems // *Engineering Science and Technology, an International Journal*. 2023. Vol. 44. P. 101455. DOI:10.1016/j.jestch.2023.101455
15. Zhang C., Patras P., Haddadi H. Deep Learning in Mobile and Wireless Networking: A Survey // *IEEE Communications Surveys & Tutorials*. 2019. Vol. 21. Iss. 3. PP. 2224–2287. DOI:10.1109/COMST.2019.2904897
16. Magadum A.A., Ranjan A., Narayan D.G. DeepQoS: A Deep Reinforcement Learning based QoS-Aware Routing for Software Defined Data Center Networks // *Proceedings of the 12th International Conference on Computing Communication and Networking Technologies (ICCCNT, Kharagpur, India, 06–08 July 2021)*. IEEE, 2021. DOI:10.1109/ICCCNT51525.2021.9579514
17. Volokyta A., Kogan A., Cherevatenko O., Korenko D., Oboznyi D., Kulakov Y. Traffic Engineering with Specified Quality of Service Parameters in Software-defined Networks // *International Journal of Computer Network and Information Security (IJCNIS)*. 2024. Vol. 16. Iss. 5. PP. 1–13. DOI:10.5815/ijcnis.2024.05.01
18. Sood M. 5G Network Slicing to Deliver Improved User Experience // *International Journal of Computer Trends and Technology*. 2023. Vol. 71. Iss. 9. PP. 59–68. DOI:10.14445/22312803/IJCTT-V71I9P107
19. Yu S., Liu M., Dou W., Liu X., Zhou S. Networking for Big Data: A Survey // *IEEE Communications Surveys & Tutorials*. 2017. Vol. 19. Iss. 1. PP. 531–549. DOI:10.1109/COMST.2016.2610963
20. Wen J., Zhang Z., Lan Y., Cui Z., Cai J., Zhang W. A survey on federated learning: challenges and applications // *International Journal of Machine Learning & Cybernetics*. 2023. Vol. 14. PP. 513–535. DOI:10.1007/s13042-022-01647-y
21. Strannegård C., Häggström O., Wessberg J., Balkenius C. Transparent Neural Networks: Integrating Concept Formation and Reasoning // *Proceedings of the 5th International Conference on Artificial General Intelligence (AGI, Oxford, UK, 8–11 December 2012)*. *Lecture Notes in Computer Science*. Berlin, Heidelberg: Springer, 2012. Vol. 7716. PP. 302–311. DOI:10.1007/978-3-642-35506-6_31
22. El Boudani B., Dagiuklas T., Iqbal M. SO-KDN: A Self-Organised Knowledge Defined Networks Architecture for Reliable Routing. 2021. PP. 160–166. DOI:10.1145/3459955.3460617

References

1. Zoraida B.S.E., Ganesan I. A Comparative Study on Software-Defined Network with Traditional Networks. *TEM Journal*. 2024;13(1):167–76. DOI:10.18421/TEM131-17
2. Hakiri A., Gokhale A., Berthou P., Schmidt D.C., Gayraud T. Software-defined Networking: Challenges and Research Opportunities for Future Internet. *Computer Networks*. 2014;75(A):453–71. DOI:10.1016/j.comnet.2014.10.015
3. Ashtari S., Zhou I., Abolhasan M., Shariati N., Lipman J., Ni W. Knowledge-defined networking: Applications, challenges and future work. *Array*. 2022;14:100136. DOI:10.1016/j.array.2022.100136
4. Jarrahi M.H., Askay D., Eshraghi A., Smith P. Artificial intelligence and knowledge management: A partnership between human and AI. *Business Horizons*. 2023;66(1):87–99. DOI:10.1016/j.bushor.2022.03.002
5. Srinivas T.A., Donald A., Thippanna G., Kousar M., Priya A. From Control to Chaos: The Dynamic SDN Control Plane. *International Journal of Advanced Research in Science, Communication and Technology*. 2023;3(2):494–502. DOI:10.48175/IJARSC-8527
6. Mestres A., Rodriguez-Natal A., Carner J., Barlet-Ros P., Alarcón E., Solé-Simó M., et al. Knowledge-Defined Networking. *ACM SIGCOMM Computer Communication Review*. 2017;47(3):2–10. DOI:10.1145/3138808.3138810
7. Li W. Explore the Evolution of Computer Network Architecture. *International Journal of Religion*. 2024;5(11):9034–9054. DOI:10.61707/ahr2da31
8. Das R., Pohrmen F., Maji A., Saha G. FT-SDN: A Fault-Tolerant Distributed Architecture for Software Defined Network. *Wireless Personal Communications*. 2020;114:1045–66. DOI:10.1007/s11277-020-07407-x
9. Wang T., Su Z., Hamdi M. Rethinking the Data Center Networking: Architecture, Network Protocols, and Resource Sharing. *IEEE Access*. 2014;2:1481–96. DOI:10.1109/ACCESS.2014.2383439
10. Agoulmine N. Chapter 1. Introduction to Autonomic Concepts Applied to Future Self-Managed Networks. In: *Walsh S.M., Strano M.S. (ed.) Autonomic Network Management Principles*. Academic Press; 2011. p.1–26. DOI:10.1016/B978-0-12-382190-4.00001-2
11. Silver E. An overview of heuristic solution methods. *Journal of the Operational Research Society*. 2004;55(9):936–56. DOI:10.1057/palgrave.jors.2601758
12. Oladipupo T. Machine Learning Overview. In: *Zhang Y. (ed.) New Advances in Machine Learning*. InTech; 2010. p.9–18. DOI:10.5772/9374
13. Nasteski V. An overview of the supervised machine learning methods. *Horizons*. 2017;4:51–62. DOI:10.20544/HORIZONS.B.04.1.17.P05
14. Yazici İ., Shayea I., Din J. A survey of applications of artificial intelligence and machine learning in future mobile networks-enabled systems. *Engineering Science and Technology, an International Journal*. 2023;44:101455. DOI:10.1016/j.jestch.2023.101455
15. Zhang C., Patras P., Haddadi H. Deep Learning in Mobile and Wireless Networking: A Survey. *IEEE Communications Surveys & Tutorials*. 2019;21(3):2224–87. DOI:10.1109/COMST.2019.2904897
16. Magadum A.A., Ranjan A., Narayan D.G. DeepQoS: A Deep Reinforcement Learning based QoS-Aware Routing for Software Defined Data Center Networks. *Proceedings of the 12th International Conference on Computing Communication and Networking Technologies, ICCNT, 06–08 July 2021, Kharagpur, India*. IEEE; 2021. DOI:10.1109/ICCCNT51525.2021.9579514

17. Volokyta A., Kogan A., Cherevatenko O., Korenko D., Oboznyi D., Kulakov Y. Traffic Engineering with Specified Quality of Service Parameters in Software-defined Networks. *International Journal of Computer Network and Information Security (IJCNIS)*. 2024;16(5):1–13. DOI:10.5815/ijcnis.2024.05.01
18. Sood M. 5G Network Slicing to Deliver Improved User Experience. *International Journal of Computer Trends and Technology*. 2023;71(9):59–68. DOI:10.14445/22312803/IJCTT-V71I9P107
19. Yu S., Liu M., Dou W., Liu X., Zhou S. Networking for Big Data: A Survey. *IEEE Communications Surveys & Tutorials*. 2017;19(1):531–49. DOI:10.1109/COMST.2016.2610963
20. Wen J., Zhang Z., Lan Y., Cui Z., Cai J., Zhang W. A survey on federated learning: challenges and applications. *International Journal of Machine Learning & Cybernetics*. 2023;14:513–35. DOI:10.1007/s13042-022-01647-y
21. Strannegård C., Häggström O., Wessberg J., Balkenius C. Transparent Neural Networks: Integrating Concept Formation and Reasoning. *Proceedings of the 5th International Conference on Artificial General Intelligence, AGI, 8–11 December 2012, Oxford, UK. Lecture Notes in Computer Science, vol.7716*. Berlin, Heidelberg: Springer; 2012. p.302–311. DOI:10.1007/978-3-642-35506-6_31
22. El Boudani B., Dagiuklas T., Iqbal M. SO-KDN: A Self-Organised Knowledge Defined Networks Architecture for Reliable Routing. 2021;160–6. DOI:10.1145/3459955.3460617

Статья поступила в редакцию 28.11.2024; одобрена после рецензирования 18.04.2025; принята к публикации 21.04.2025.

The article was submitted 28.11.2024; approved after reviewing 18.04.2025; accepted for publication 21.04.2025.

Информация об авторах:

БЛАН
Филип Сергеевич

аспирант кафедры инфокоммуникационных систем Санкт-Петербургского государственного университета телекоммуникаций им. проф. М.А. Бонч-Бруевича
 <https://orcid.org/0009-0002-8760-1089>

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>

Авторы сообщают об отсутствии конфликтов интересов.

The authors declare no conflicts of interests.