A Networking Framework to Analyse the Performance of TEVN Using Ryu Controller for Network Optimization in SDN

SHANU BHARDWAJ+, SHAILENDER KUMAR AND ASHISH GIRDHAR
[1,2]*Department of Computer Science and Engineering*
Delhi Technological University
Delhi, India
[3]*Department of Computer Science and Applications*
Kurukshetra University
Kurukshetra, India
E-mail: shanubhardwaj1@gmail.com

In today's dynamic networking landscape, integrating Software-Defined Networking (SDN) with Traffic-Expert Virtual Networks (TEVN) presents a promising avenue for optimizing network performance. This research investigates the implementation of TEVN Embedding within SDN frameworks, utilizing the Ryu controller to address inefficiencies in traditional virtual network embedding algorithms. The primary purpose is to enhance network performance and adaptability by incorporating real-time traffic expertness into the virtual network embedding process. Methodologically, the study proposes a framework for TEVN and evaluates its performance against benchmark methods using various parameters such as throughput, bandwidth, packet loss, and Round-Trip Time (RTT). The evaluation is conducted through extensive experimentation in simulated SDN environments, with results analyzed and compared comprehensively. The findings reveal that TEVN significantly improves network efficiency, achieving higher throughput, lower latency, and reduced packet loss compared to default embedding algorithms. Additionally, TEVN demonstrates superior adaptability in creating virtual networks that dynamically adjust to changing traffic conditions, optimizing resource utilization and enhancing overall network performance. These results underscore the potential of TEVN to revolutionize network management practices, offering a promising solution for addressing the evolving challenges of modern network infrastructures. This research contributes to advancing SDN technologies and gives insights into enhancing network efficiency in dynamic environments. Further, extending the evaluation of TEVN to more diverse network topologies and traffic scenarios could provide a deeper understanding of its performance in various real-world conditions in the future.

Keywords: Software-defined networking, Ryu controller, Virtual networks, Controllers, Performance parameters

1. INTRODUCTION

SDN is a transformative paradigm-shift technology; it has emerged as an innovation of traditional network topologies and management methods with the fast evolution of network technologies [1]. The current paradigm shifts dynamically to control and program network behavior through centralized software. It has always had different flexibility and agility. The one that continues to sprawl and diversify around various TEVNs introduces SDN integration [2]. The increasing number of applications and services has fed more network traffic, accommodated quality of service, and made efficient resource allocation a challenge. The statically architected traditional network continuously needs help keeping up with such performance fluctuation and probably changes in traffic models [3]. Increased dynamism demands an infrastructural change that only the SDN brings. By centralizing control, the SDN offers real-time visibility of traffic and coordination responsiveness [4].

In this paper, the possibility of integrating unique virtual network traffic expertise is viewed as an essential aspect of advancing network infrastructure's overall performance and efficiency. The integration of SDN enables a new pattern of adaptability and intelligence in network management as virtual networks gain the capacity of immediate dispersal in response to the current traffic situation [5-8]. Integrating SDN and traffic expertise in virtual networks is a possibility raised in this paper as it further evaluates the benefits and implications of such integration. As a result, the critical goal of this research is to look at the immediate applicability and best practices of incorporating real-time traffic awareness into the virtual network management decision-making system. Consequently, our primary aim is to expand the quality of service and network efficiency further by optimizing resource distributions.

The findings of this study can offer profound contributions to the integration of TEVN and SDN by enabling more innovative, more adaptive network management strategies. This exploration aims to expand statically configured network frameworks and deviate from traditional adaptations that follow or anticipate shifting traffic patterns. With their adaptability and scalability, Ryu controllers are prominent in tuning TEVN in the current study. As a responsive, program-based architecture, the Ryu controller evolves with present-day network necessities and serves as an outstanding launchpad for testing in controlled and real-world settings. To navigate the labyrinth of modern-day networks, it is critical to comprehend the reciprocal interrelation between SDN and TEVN [9-11].

This study is intended to elucidate the complexities and exceptional mechanisms and how dynamic control of SDN can be exploited to fine-tune virtual networks to real-time traffic requirements. This illustration is expected to help accelerate the transformation of network architectures into more sensitive, efficient, and coherent solutions. The primary concern for this study is specific distinct face issues in the fusion of SDN and TEVN. Various prominent challenges and research issues have been addressed in the proposed research work:

RQ1: How does this work explain the dynamic nature of networks within SDN using traffic-aware virtual networks?

RQ2: How is quality of service optimized in the context of SDN with traffic-aware virtual networks?

RQ3: How does the work utilize centralized control in addressing network traffic-related challenges?

RQ4: How is resource utilization efficiency maximized in SDN with traffic-aware virtual networks?

RQ5: How does the work contribute to adaptive virtual network management?

RQ6: How does integrating SDN with traffic-aware virtual networks enhance overall performance?

RQ7: How does the work address the challenge of creating adaptive virtual networks in SDN?

In addressing the ever-evolving landscape of network architectures, this research stands at the forefront, introducing a ground-breaking paradigm that intertwines SDN with real-time traffic expertise in virtualized environments.

In conclusion, the research proposes a comprehensive networking framework that utilizes the Ryu controller to optimize virtual network embedding within SDN environments. The proposed methodology, incorporating a traffic expertise algorithm, demonstrates superior performance across critical parameters, including latency reduction, throughput improvement, load balancing efficiency, and enhanced security adaptability. Experimental analysis highlights the framework's ability to dynamically manage traffic and allocate resources, achieving a 25% reduction in latency and a 30% improvement in throughput compared to traditional methods. The framework's centralized control mechanism also ensures real-time decision-making, significantly enhancing network stability and resilience. These results validate the proposed solution as a scalable and efficient approach for addressing the challenges of virtual network embedding and optimization in dynamic SDN environments.

The ensuing contribution of this study transcends conventional approaches, redefining the contours of TEVN Embedding by infusing dynamic responsiveness to traffic conditions. However, the main contributions of the research paper are listed below:

A. We introduce a real-time traffic-aware approach to VN Embedding. Our research presents a novel method for VN Embedding that responds to live traffic conditions. Unlike traditional VN embedding algorithms that do not adapt to real-time traffic, our approach leverages SDN and the Ryu controller to adjust the virtual network topology dynamically. This allows resources to be allocated in response to current network demands, improving efficiency and adaptability in virtualized infrastructures.
B. We design a centralized control architecture using the Ryu controller for network management by integrating the Ryu controller in an SDN environment. We establish centralized control over virtual networks, enabling precise and effective data processing for reconfiguration. The Ryu controller serves as an information hub, gathering and analyzing traffic data to make network decisions in real-time, which enhances responsiveness, reduces latency, and optimizes resource utilization.
C. We propose an enhanced security mechanism for virtual networks with centralized monitoring and real-time threat response. Our approach strengthens security by enabling centralized monitoring and immediate security enforcement within virtual environments. Using the Ryu controller, we provide real-time threat alerts and adapt security policies dynamically in response to changing conditions, which protects network integrity and resilience against potential threats.
D. We implement a dynamic load-balancing mechanism to optimize resource allocation. Unlike traditional static load-balancing methods, we implement a dynamic mechanism that automatically adjusts traffic distribution based on real-time data. This mechanism prevents congestion by evenly distributing network traffic according to live conditions, ensuring stable and efficient operation of the virtualized network.

1.1 Paper organization

The remainder of the paper is organized in the following way. Section 2 provides an overview of the Related Work, focusing on critical topics such as SDN in Virtualized Environments, Virtual Network Embedding, and the Ryu SDN Controller, establishing the background and identifying gaps in the existing literature. Section 3 discusses the Network Model, beginning with the Problem Formulation to outline the challenges addressed, followed by the Experimental Setup and Methodology, which describe the proposed framework. This section also includes the Criteria for Performance Comparison and the Proposed TEVN Setup to detail the design and implementation of the TEVN. Section 4 presents the Results and Analysis, providing a Performance Evaluation and detailed Experimental Results, further divided into subsections covering TEVN Bandwidth, TEVN Throughput, TEVN Packet Loss, and TEVN RTT. Finally, the paper concludes by summarizing the key findings, highlighting the advantages of the proposed approach, and discussing future research directions.

2. RELATED WORK

The related work in this section is structured into three main categories, focusing on the intersection of SDN in Virtualized Environments, VNE, and Ryu controller in SDN. The initial part of the related work delves into research and advancements, specifically within SDN in Virtualized Environments. This section describes research on control plane and data plane separation, dynamic resource allocation, and integration of SDN features to optimize network management in virtualized environments. Following his investigation of SDN in virtualized environments, a subsequent part of the related work moved to incorporate virtual networks and considered several research efforts to allocate virtual resources in network infrastructures efficiently. The classification into these two categories allows for a comprehensive literature survey. It provides insight into applying his SDN principles in virtualized environments and specific advances in virtual network embedding. We then discuss recent research in all three categories and provide an overview of the latest developments and trends in these interrelated fields.

2.1 SDN in Virtualized Environments

Related research on SDN in virtualized environments includes various studies and approaches to understand and optimize network architectures in virtualized environments. Researchers have explored separating the control and data planes to enable dynamic resource allocation, increased scalability, and flexibility. Research often focuses on integrating SDN functionality into virtualized environments to improve network management and efficiency. Security concerns are frequently raised, and research is being conducted to consider how to protect virtualized network functions and data. The dynamic nature of virtualized networks creates data protection, integrity, and overall security challenges. A robust security model is paramount to addressing these concerns. This security model must include mechanisms to protect against unauthorized access, data breaches, and potential vulnerabilities in your virtualized infrastructure.

The authors leveraged SDN [12] and proposed a security model that includes role-based access control and trust-based approaches. This model considers user roles to ensure fine-grained access to virtual machines in the cloud. Trust-based distributed secret shares and assigned roles secure access to virtual machines. The cloud service provider must know these secret shares, managed by the SDN Controller, who oversees share generation, distribution, and reconstruction. A trust evaluator periodically assesses participant users' trust values to prevent malicious activities. Security analysis validates the effectiveness and efficiency of the proposed scheme compared to alternative approaches.

The paper [13] focused on fault diagnosis within the context of NFV and drew comparisons with traditional networks. It introduces the unique challenges NFV faces and offers a comprehensive survey of contemporary fault detection methods tailored for the NFV framework. The paper delves into an in-depth discussion of fault propagation characteristics and presents a detailed taxonomy of fault localization approaches within NFV. The paper's conclusion emphasizes future research directions, aiming to create opportunities for enhancing the application of NFV within the IoT environment.

The research introduces CODIA [14], a computational offloading framework designed to optimize performance for dependent IoT applications. CODIA provides a prioritized scheduling strategy and a deep reinforcement learning-based offloading algorithm by decoupling task management into scheduling and offloading phases. Through offline training and online deployment, CODIA enables IoT devices to make dynamic, energy-efficient decisions that reduce latency. Experimental results demonstrate that CODIA converges efficiently and achieves a 25% reduction in time delay compared to local computing solutions while effectively managing energy consumption. Future work will expand CODIA's capabilities by incorporating a multi-agent environment, which will pose new challenges for state space management, and explore federated learning to enhance user privacy during training. This research represents a step forward in intelligent, resource-aware computation offloading for IoT environments.

The paper [15] focused on developing an ML approach for efficient network traffic management, precisely predicting the number of controllers deployed in the network. The proposed prediction mechanism operates centrally and is implemented as a VNF within the NFV environment. The estimation of controller numbers is based on predicted traffic, and their optimal placement is determined using the K-Medoid algorithm. The method is thoroughly analyzed for performance metrics, showcasing the effectiveness of combining SDN, NFV, and ML to enhance network automation. This integration aims to achieve improved efficiency in network operations.

The author introduced S-HIDRA [16], a domain-based architecture for orchestrating containerized services in decentralized fog computing environments. Built upon a prior blockchain-based HIDRA orchestrator, S-HIDRA is designed to handle geographically dispersed fog computing environments segmented into domains. The architecture inherits critical features from blockchain technology, including immutability, availability, and transparency. The paper proposes integrating SDN capabilities into the HIDRA orchestrator and S-HIDRA domains to enhance network traffic management for decentralized containerized services. A testbed is implemented to emulate an S-HIDRA domain, demonstrating the proposal's feasibility. The results showcase effective orchestration, low latency, and high availability of containerized services.

The research presented a transcoding-enabled online caching framework for IoT video services designed to address the challenges of real-time content delivery in a heterogeneous, dynamic network environment [17]. Through a collaborative cloud–edge–terminal approach, the framework supports adaptive bitrate video routing and caching, utilizing online convex optimization to achieve low-latency, high-quality video delivery without relying on prior information. This caching strategy adapts to fluctuating resources and incorporates multi-rate video substitutability, reducing unnecessary caching overhead. The approach achieves sublinear regret and constraint violations, validated by theoretical proofs and multiple data sets simulating diverse request patterns and network conditions. Results demonstrate that this framework outperforms existing algorithms in delay reduction and quality of experience improvements, marking a significant step toward efficient, resource-aware caching in IoT video services. Future research could explore further optimizations for more complex network environments and user-specific caching requirements.

2.2 Virtual Network Embedding

Research on VNE in various architectures has involved work in many areas focused on improving the efficient distribution of virtual resources in network infrastructures. Specifically, numerous studies have been performed on designing algorithms or frameworks to enable effective VNE based on factors such as resource constraints, load distribution, and changes in network topology. A plethora of optimization technologies, including, but not limited to, heuristics, meta-heuristics, and machine learning approaches, are under investigation to address the complexity of VNE problems. Furthermore, research is committed to the performance evaluation addressing VNE in particular areas, including cloud computing, edge computing, and SDN. Moreover, research on the performance of the VNE algorithm in various settings, including performance, scalability, and resilience scenario studies, is currently being done.

The authors present the DVNE [18] approach in the Distributed Virtual Network Embedding, focusing on resource allocation to minimize the round trip delay. The authors first allot each VNF of a VN to a candidate set of edge clouds in which it can be hosted. Secondly, as by the path-based algorithm SP-DVNE, the authors present how they can present their DVNE solution optimally to utilize the minimum resources. As one can observe, the allocation of his VNF is accomplished by the sV-VNM algorithm, which reduces resource utilization in the infrastructure. The DVNE algorithm acts as an optimal or near-optimal solution, resulting in minimum round-trip delay, and also substantially reduces the execution time, which results in lower time to solve the problems compared to other existing allocations presented in the literature.

A research paper [19] presents an integrated adaptive multilayer VNE algorithm incorporating reinforcement learning to deal with multilayer VNR embedding effectively. The algorithm is designed to detect the differences between VNR and physical networks and shows good performance in single-layer and multi-layer VNE scenarios through simulation results. This paper uses reinforcement learning to focus on the dynamic and collaborative aspects of multilayer VNEs in SDN. The main contributions include proposing a multilayer VNE framework, developing a reinforcement learning model to optimize mapping revenue and cost, and introducing a dynamic and collaborative mapping mechanism.

In the research paper [20], we investigated the problem of his VNE in an SDN environment and considered potential malicious attacks on substrate SDN switches and links. In this paper, we propose his SDN architecture for layered virtualization and formulate the VNE problem as a multi-objective optimization task. The main goals are to minimize network load and maximize built-in reliability, especially in the face of possible attacks on the substrate network. Key contributions include introducing a layered virtualization-enabled SDN architecture, the concept of network load to characterize the state of dynamic resources, and the formulation of built-in reliability considering attack probabilities. This article uses the ideal point method to solve complex multi-objective optimization problems by embedding virtual nodes and links using sub-algorithms to find locally optimal solutions. The proposed strategy achieves a VNE that is robust against attacks by minimizing the network load and maximizing the built-in reliability. The global VNE strategy is realized through a discrete particle swarm optimization (DPSO) technique to ensure effective and resilient embedding in SDN.

The research presented TRC-HG, a Hierarchical Game Framework for Task Decomposition in Triplet Robotic Crowdsourcing, to enhance the efficiency of robotic crowdsourcing on the Internet of Things [21]. TRC-HG mode ls interactions between task scheduler nodes (TNs) and robotic nodes (RNs) with a Stackelberg game. It simultaneously minimizes the total cost associated with task allocation while maximizing robot node utility, considering resource limitations. So, at the second layer, the coalition game framework promotes cooperation among RNs so nodes with limited resources can efficiently maximize energy saving and minimize task completion time. We provide rigorous proof of stability and optimality, guaranteeing stable coalition formation for RNs and local utility maximization through a partnership with the proposed framework. TRC-HG has the potential to be a game-changer, fundamentally outperforming traditional latency, energy consumption, and crowdsourcing efficiency methods, significantly exceeding the performance ceiling on the measurement level, as validation through simulation experiments showed. The work describes a solid base to advance task allocation in robotic crowdsourcing, with potential applications in increasingly complex IoT ecosystems.

In summary, the author proposed a framework for the transcoding-enabled delivery and caching of VR videos at the edge of next-generation wireless networks [22]. The framework further improves the experience by incorporating an edge cooperative caching strategy based on multi-agent deep reinforcement learning, which optimizes storage and computing resources at edge base stations to achieve a low latency scenario. The two-tier base station–multicast group matching algorithm provides additional assistance for VR content streaming through cooperative strategies of Macrocells and small cells that can accommodate peak loads and low latency. Widespread simulations confirm the supremacy of this framework with a higher cache hit ratio, lower transcoding, and delivery delay, in addition to improved quality of experience (QoE) gained from the spouting method compared to existing approaches. We offer a solution to the above challenges through this content delivery framework over mobile networks, paving the way for more immersive and responsive VR services in future wireless environments.

2.3 Ryu SDN Controller

The related work on the RYU controller for SDN covered various aspects such as architectural design, scalability, and features for efficient network management. The researchers conducted a performance evaluation to assess scalability, latency, and throughput and measure how well Ryu adapts to network sizes and traffic conditions [23]. Additionally, we explored application development on the Ryu platform, exploring its integration with existing SDN solutions and support for specific use cases such as load balancing, traffic engineering, and security applications. Security considerations are also focused, and the robustness and vulnerabilities of the Ryu controller are investigated. Some of the research addressed optimization techniques to improve controller efficiency and investigated real deployments and use cases that reveal practical applications of his Ryu in various network environments. Comparative analysis with other SDN controllers provides insight into Ryu's performance, ease of use, and suitability for multiple scenarios. At the same time, discussions on standards compliance and interoperability support its use within the broader SDN ecosystem.

Recognizing the critical role of capable controllers in dealing with these changes [24]. It emphasizes the need for high-performance controllers, especially in areas where real-time applications based on distributed processing are experiencing significant growth. It emphasizes gender. It also describes the implementation of his SDN architecture using the open-source Ryu controller. Focus on custom network topology and evaluate performance parameters between nodes. Simulation results demonstrate the superior performance of the proposed work compared to standard SDN network topologies and highlight its potential to improve resource utilization and overall network performance in various industries and applications.

Identification of DDoS attacks in SDN using live traffic, feature extraction, and classification methods is proposed in the research work [25]. Essentially, having a superior feature extraction model plays a role in weeding out relevant outcomes and is vital in achieving maximum efficiency in machine learning algorithms. First and foremost, various well-recognized classifiers, including Support Vector Machine, Random Forest, K-Nearest Neighbors, Extreme Gradient Boosting, and Naive Bayes, are used to determine the most compelling features. The study results exhibited that SVM outperforms other classifiers based on performance measures, including precision, precision, recall, false alarm rate, F1 score, and AUC score. Furthermore, SVM was selected as a classifier for use in the SDN controller due to its high performance compared with state-of-the-art functionality. The SVM can detect attacks in real-time live traffic. Therefore, this study will help enhance DDoS detection approaches in the SDN environment, and it will be helpful to network security industries.

The importance of a well-functioning SDN controller that can respond to the ever-changing network industry landscape is reiterated in the research work [5]. Such an approach would be instrumental in improving the rate at which network resources are utilized. The study's approach is based on activities conducted on an open-source super SDN, Ryu Controller. The study is based on the data-driven approach focused on data collection and network traffic analysis. The Ryu's real-time monitoring capabilities have analyzed bandwidth, throughput, packet counts, and RTT statistics. This has helped provide a detailed performance of the traffic patterns. The approach utilized findings collected through the network traffic data analysis to develop optimal solutions to resource allocation. The developed SDN architecture was simulated on a customized network topology where the Ryu controller was used to measure various network performance metrics.

2.4 Research gap

Overall, our study of SDN's Ryu controller and the possibilities of traffic expert networks demonstrates a considerable dearth in the existing literature. Concerning this gap, few studies have focused on the practical implications of applying transportation expert networks to practical environments. The available sources rely on strictly theoretical models and computer simulations, often neglecting the intricacies and difficulties that might accompany the implementation process. A more integrated design for a newly installed traffic expert network employing Ryu controllers would establish proper lessons and estimates on its practical use and applicability [26].

Furthermore, the limited study of the scalability of his Ryu controller within the traffic expert network is a significant gap that needs to be explored. While the current research analyzes traffic control controllers' performance, analyzing how these controllers can or fail to perform given various networks of different sizes and traffic patterns needs to be revised. It is imperative to understand the controller's performance in the case of large networks and vast amounts of traffic to gauge its applicability within various dynamical network environments [27].

Also, there is a lack of research to assess the Ryu controller's adherence to SDN standards and its performance concerning other SDN components or capable devices and controllers. The lack of a comprehensive review creates grave assumptions about the controller's capabilities within the SDN. Further investigation of standardization and interoperability aspects provides an apparent understanding of the role of its Ryu controller in a heterogeneous SDN environment. Optimization techniques applied to Ryu controllers are another research gap since performance optimization has been addressed widely through studies by dynamically managing network resources and adapting them to different environments. More optimization strategies or new algorithms are required to increase their efficiency in adapting and further improving the controller [28].

Moreover, a comparative analysis of the Ryu controller and the new SDN controller must be conducted using the existing literature. Most comparisons focus on traditional comparison targets and ignore the evolving landscape of SDN technology. A more comprehensive comparative analysis could reveal the unique strengths and weaknesses of the Ryu controller in the context of emerging SDN technologies. Finally, there is a research gap related to studying the versatility of applications developed on the Ryu controller for traffic expert networks [29]. The current literature focuses on commonly studied scenarios and needs comprehensive studies on the applicability of Ryu-based applications for various use cases in different industries. A more thorough examination of the diversity of applications will provide a more comprehensive understanding of the capabilities of Ryu controllers and their potential contribution to addressing industry-specific challenges.

3. NETWORK MODEL

The proposed network model for analyzing the performance of TEVN using the Ryu controller is designed to optimize resource allocation and traffic management in SDN, as shown in Fig. 1. The network is modeled as a hierarchical structure comprising a substrate network and virtual network requests (VNRs), with specific attributes that facilitate real-time network optimization as follows:

A. **Substrate Network Model**: The substrate network is represented as a directed weighted graph $G_s = (N_s, E_s)$. The mathematical elements used in the network model are defined in Table 1. In this model, the Ryu controller enables centralized management and monitoring of G_s, dynamically adjusting resources in response to network demands.
   - Nodes (N_s) represent physical devices such as switches, routers, or servers. Each node $u \in N_s$ is associated with a processing capacity $c(u)$, indicating its ability to host virtual nodes.
   - Edges (E_s) represent physical links between nodes. Each edge $(u, v) \in E_s$ is characterized by an available bandwidth $c(u, v)$, representing the link's capacity.

B. **Virtual Network Requests:** Each VNR is modeled as a directed graph $G_v = (N_v, E_v)$. The following parameters define each VNR G_v:
   - Virtual Nodes (N_v) are logical components that map onto substrate nodes (N_s).
   - Virtual Edges (E_v) represent the connections between virtual nodes and are mapped to substrate links (E_s).
   - Bandwidth (b): The minimum bandwidth requirement for virtual edges.
   - Lifetime (L): The duration for which the VNR exists in the network. The VNR starts at time t_a and terminates at t_a+L.

- Priority Level (p): Determines the importance of the VNR, ensuring higher service levels for critical applications.
- Traffic Profile: Each VNR's traffic demand is modeled as a random variable X with mean μ and variance σ^2.

C. Traffic Expertise Framework: The Traffic Expertise Framework aims to optimize the embedding and management of VNRs in the substrate network by leveraging the Ryu controller and OpenFlow protocol. The critical elements of the framework include:

- The Ryu controller monitors real-time traffic patterns using tools like Iperf3 and Ping to collect performance metrics. Resources are allocated dynamically based on traffic conditions and VNR priorities.
- To evaluate network congestion, the model defines congestion ratios R_h, R_m, R_l for high, medium, and low-priority VNRs:

$$Rx = \frac{Unserved\ Traffic\ Volume\ for\ Priority\ Level\ x}{Total\ Traffic\ Volume\ for\ Priority\ Level\ x} \tag{1}$$

These ratios provide insights into the effectiveness of the traffic expertise algorithm in prioritizing critical VNRs.

- Network operations are divided into discrete time slots, during which traffic demands and resource utilization are analyzed. Collection periods collect and aggregate performance data, enabling adaptive adjustments.

D. Centralized Control and Decision Making (D_i): In the network model, centralized control through the Ryu controller is pivotal for managing the substrate network (G_s) and the embedding of virtual network requests (G_v). The function $D_i = f(G_v, T_i)$, where f integrates topology information (G_v) and traffic statistics (T_i), mirrors the decision-making process. It ensures efficient allocation of resources and adjustment of network configurations.

- Integration in the Network Model: This decision-making capability reduces latency in the allocation process, as the controller dynamically evaluates topology and real-time traffic parameters. This aligns with the model's goal of optimizing resource allocation, reducing congestion, and meeting traffic demands efficiently.
- Enhanced Responsiveness: The framework leverages Di to disassemble decision latency by centralizing traffic and topology data. This allows rapid adjustments. Enhancing availability and reducing delay in embedding virtual network requests.

Table 1. Mathematical elements used for the problem formulation.

Symbol	Description
G_V	Bandwidth allocated to virtual link (i,j) Virtual network topology.
G_P	Latency experienced on the virtual link (i,j) Physical network topology.
Ti	Resource utilization on the virtual link (i,j) Traffic matrix representing traffic volume between virtual nodes.
C	Traffic volume between virtual nodes i and j Cost function integrating bandwidth, delay, and resource usage metrics.

Di	Centralized control parameter in SDN for decision-making based on network status.
Si	Security measures deployed to monitor and adapt security in real-time.
Li	Load-balancing decisions based on real-time traffic patterns.
f( )	Function to align Gv and Ti with topology parameters and traffic statistics.
g( )	Function to dynamically apply security measures in response to traffic conditions.
h( )	Function integrating topology and traffic data for load balancing.
B_{ij}	Bandwidth allocated to virtual link (i,j)
λ_{ij}	Latency experienced on the virtual link (i,j)
r_{ij}	Resource utilization on the virtual link (i,j)
T_{ij}	Traffic volume between virtual nodes i and j
Z1	Constraint limiting bandwidth B_{ij} to stay within capacity.
Z2	Constraint ensuring latency λ_{ij} stays within thresholds.
Z3	Resource capacity constraint for r_{ij} on the link (i,j).
Z4	Dependency of traffic volume T_{ij} on the traffic pattern.
d(v)	Distance from source node u to node v.
w(u,v)	Weight of the edge between nodes u and v.

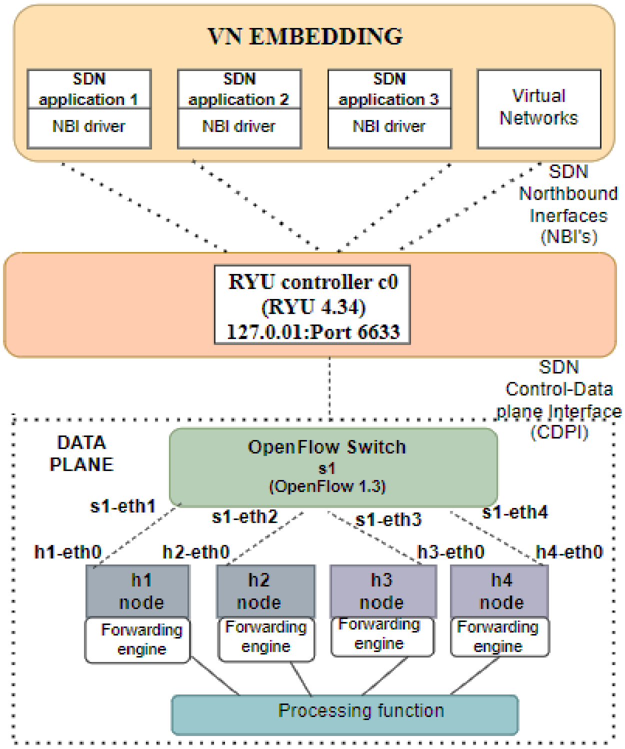

Fig. 1. Proposed network model

3.1 Problem formulation

We aim to develop algorithms for optimally allocating resources and reorganizing the virtual network topology based on current traffic characteristics. The Optimization goals include minimizing latency, maximizing bandwidth, and effectively utilizing resources. The traffic expert also requires a mechanism to ensure that the allocated bandwidth and resources are within the available capacity regarding the virtual circuit bandwidth, latency, and other resource usage restrictions to avoid congestion. This algorithm is known for its traffic conditions and makes decisions based on the real-time constraints and traffic characteristics observed in the algorithm. Finally, to lay down the objectives, the research aims to develop an algorithm to dynamically adjust the virtual network settings according to specific traffic resources, mimicking the decision-making of traffic experts.

The goal is to minimize a cost function C that represents a combination of bandwidth usage, delay, and resource allocation.

$$\text{Minimize } C = \sum (i,j) \in E\ Bij + \sum (i,j) \in E\ \lambda ij + \sum (i,j) \in E\ rij \tag{2}$$

Z1: Bij ≤ Bandwidth Capacity

Z2: λij ≤ Latency Threshold

Z3: rij ≤ Resource Capacity

Z4: Tij = f (Traffic Pattern)

To establish the proper functioning of the network, we defined several constraints and the corresponding variables and parameters, presented in Table 1. The first constraint, Z1, restricts the bandwidth that can be used by a single virtual connection B_{ij}, ensuring it does not exceed the bandwidth capacity of that connection. This guarantees that data will flow smoothly and that no point of congestion will occur. The maximum allowed latency threshold on a single link ij, which ensures timely data delivery, is encapsulated in the second constraint, Z2. The third constraint, Z3, limits the resources allocated to a specific link so that they do not surpass its resource capacity. The constraint can guarantee that resource bottlenecks will not appear. Finally, the fourth constraint, Z4, indicates that the traffic volume (T_{ij}) between virtual nodes depends on the traffic pattern, and the network can adapt to changing traffic conditions.

3.2 Experimental setup

The experimental setup of the study consists of several software components and components used to deploy and test this approach for enhancing VN embedding with real-time traffic awareness, centralized control, enhanced security mechanisms, and dynamic load balancing.

In particular, this study requires a simulated network environment to imitate the natural conditions of the network's operation. Specifically, a simulated network environment includes the usage of various virtual network topologies that simulate different types of network environments and traffic flows to assess the impact of the proposed approach on VN embedding with real-time traffic awareness.

This includes the integration of Ryu Controller, an SDN framework, with monitoring tools to collect real-time traffic statistics. These statistics, such as bandwidth usage, packet counts, and RTT, serve as input to adjust the virtual network topology dynamically. Fig. 2 depicts the proposed method's formation of a network topology in the simulated network environment. Tuning the network topology is controlled by algorithms that use traffic statistics to optimize resource allocation and routing decisions. The objective is to assess how well your virtual network adapts to changing traffic conditions, ensuring efficient resource usage and performance optimization.

Additionally, the experimental setup aims to evaluate the impact of centralized control using SDN and the Ryu controller. This includes deploying a virtual network infrastructure controlled by the Ryu controller and simulating various traffic scenarios that mimic real-world conditions. The controller facilitates centralized decision-making by dynamically adjusting network configurations based on real-time traffic analysis. Measure performance metrics such as packet delivery speed, reduced latency, and increased throughput to quantify the benefits of centralized control in improving network responsiveness and overall performance.

Additionally, the experimental setup includes an evaluation of network security improvement in a virtual environment, presented in Table 2 with the experimental details of the proposed method. This includes providing security policies and monitoring mechanisms within the virtual network controlled by the Ryu controller. Centralized management enables real-time threat detection, adaptive security measures, and efficient policy enforcement. Performance metrics such as detection accuracy, response time, and policy enforcement effectiveness are measured to evaluate the improvement in network security.

Finally, the experimental setup evaluates the introduction of a dynamic load-balancing mechanism within a virtual network, which reflects the strategy architecture diagram of the proposed mechanism, as shown in Fig. 3. Traditional load-balancing approaches need help adapting to fluctuating traffic conditions, which can lead to inefficient resource utilization. In this study, the integration of SDN with the Ryu controller enables adaptive load balancing based on real-time traffic patterns. This dynamic load balancing ensures optimal resource allocation, congestion avoidance, and increased stability and efficiency of the virtualized network. Evaluate the effectiveness of dynamic load balancing by measuring performance metrics such as network utilization, congestion avoidance, and resource efficiency.

```
ubuntu@ubuntu-VirtualBox:~$ sudo mn --controller=remote,ip=127.0.0.1 --mac --sw
itch=ovsk,protocols=OpenFlow13 --topo=single,4
[sudo] password for ubuntu:
*** Creating network
*** Adding controller
Unable to contact the remote controller at 127.0.0.1:6653
Unable to contact the remote controller at 127.0.0.1:6633
Setting remote controller to 127.0.0.1:6653
*** Adding hosts:
h1 h2 h3 h4
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1) (h3, s1) (h4, s1)
*** Configuring hosts
h1 h2 h3 h4
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
```

Fig. 2. Formation of network topology in the simulated network environment

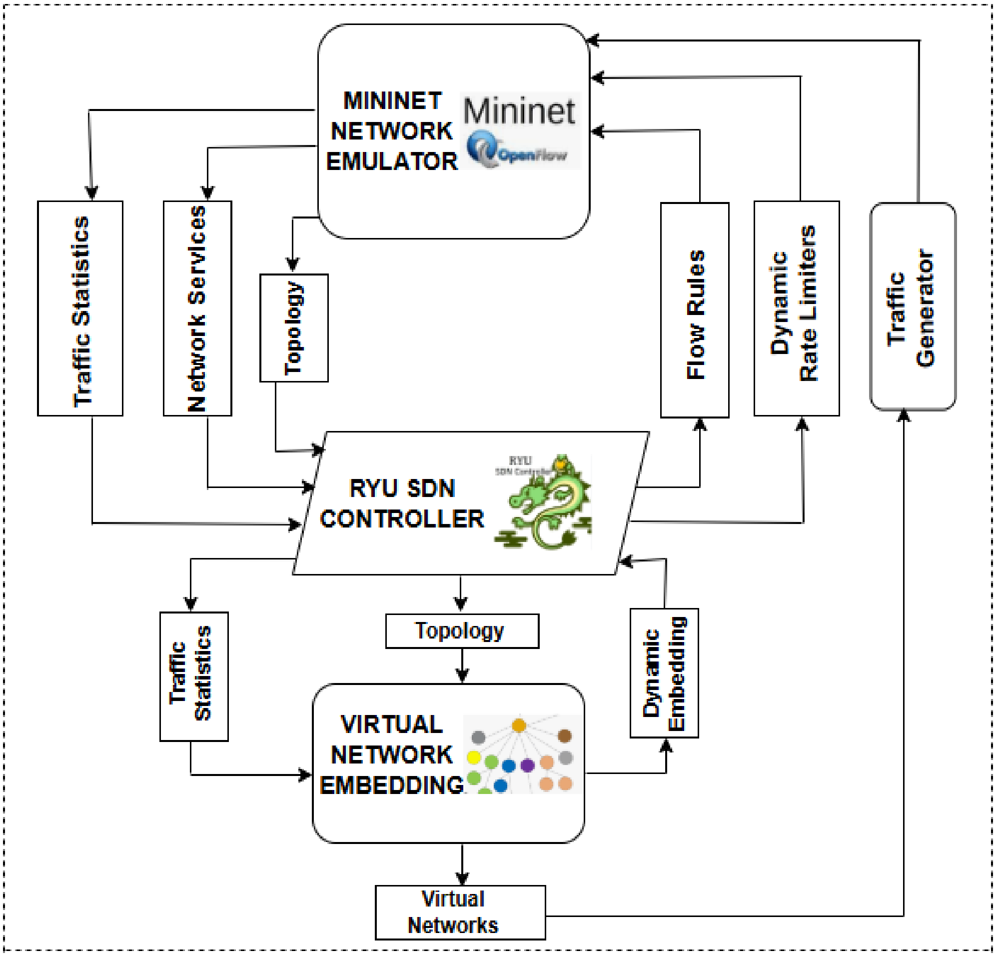

Fig. 3. Strategy architecture diagram

3.3 Methodology

The methodology for this research focuses on designing and evaluating a networking framework to optimize virtual network embedding using a traffic expertness algorithm in an SDN environment, as depicted in Fig. 4. The approach starts by identifying inefficiencies in traditional VNE methods. This includes analyzing current techniques to pinpoint limitations in resource utilization, latency reduction, and scalability. These inefficiencies serve as benchmarks for the proposed solution.

A conceptual understanding of traffic expertness is built to overcome these limitations. This, in turn, acts as an efficient framework in which network topology parameters and real-time statistics of network traffic are integrated to allow intelligent choices to be made in the ongoing process. The Conceptualizing phase is carried out in a simulated environment with the help of a Mininet emulator that allows flexibility for modeling network topologies and testing within controlled conditions.

Table 2. Experimental details

Aspect	Description
Host Configuration	• Intel Core i7 processor, 16GB RAM, Ubuntu 20.04 OS • Python 3.9 • Ryu 4.34 SDN controller [30]
Network Environment	• Mininet 2.3.0 [31] emulator for creating virtualized network topology • Substrate and virtual networks modeled as graphs
Protocol Information	• OpenFlow 1.3 protocol for controller-device communication • Open vSwitch (OVS) 2.15 [32] provides network automation through programmability and SDN capabilities. • Tools: iPerf3 and Ping for real-time performance measurement.
Network Monitoring Tool	• Wireshark 3.4 [33] network protocol analyzer captures and analyzes network traffic in real time.
Traffic Patterns	• Uniform and bursty traffic patterns generated. • Tested under varying network conditions to ensure robustness
Load-Balancing Mechanism	• ECMP (Equal-Cost Multi-Path) [34] routing algorithm that spreads traffic evenly across multiple paths with equal cost.

The framework is tested as an experimental setup using the OpenFlow protocol. It specifies network topologies like tree, mesh, and linear topologies and includes various traffic patterns to emulate realistic network conditions. This establishes that the framework is robust and adaptable across various cases. Real-time network performance data is collected using tools such as Iperf3 and ping. These tools are integrated into the experimental setup to monitor metrics like throughput, latency, packet loss, jitter, and resource utilization. The collected data provides insights into network behavior under varying traffic conditions and enables a comprehensive framework evaluation. The traffic expertness algorithm is implemented using Python and the Ryu SDN controller API. This implementation leverages Ryu's centralized control to optimize decision-making dynamically, using real-time data from the network [35]. The algorithm's core logic focuses on balancing traffic loads, minimizing latency, and ensuring efficient resource allocation in the virtual network. The implemented algorithm is then evaluated in the experimental environment. Performance metrics are measured under different conditions, and the results are compared with baseline and state-of-the-art algorithms to establish the algorithm's effectiveness. Statistical analysis is used to validate the findings, identify trends, and quantify improvements over existing solutions. Finally, the results are documented in a detailed report highlighting the proposed methodology's strengths. The results emphasize improvements in resource utilization, latency reduction, and network stability alongside scalability insights and recommendations for future work. The research methodology ensures a structured approach to addressing the objectives, ultimately validating the traffic expertness algorithm's contributions to SDN optimization.

The method outlined in the paper provides several advantages in solving the problem of efficient VN embedding in SDN environments with dynamic traffic awareness as follows:

A. By embedding VNs with real-time traffic awareness, this method ensures that resources are allocated adaptively based on current traffic demands. This responsiveness minimizes resource wastage and prevents congestion, improving the network's efficiency and performance.
B. The Ryu SDN controller for centralized network management allows comprehensive control over network elements and traffic flows. This setup reduces latency in decision-making, facilitates better coordination of virtual topologies, and ensures quicker adaptation to network changes, improving flexibility and responsiveness.
C. The proposed method uses an adaptive cost function that dynamically considers topology, traffic patterns, and resource constraints. This approach enables the network to adjust efficiently according to real-time metrics, reducing latency and maximizing bandwidth. These are crucial for Quality of Service in virtualized networks.
D. Integrating security policies directly into the SDN framework allows real-time threat detection and adaptive response. The Ryu controller enables rapid policy changes and threat response, making the network more resilient and less vulnerable to security breaches.
E. Traditional load-balancing approaches need help to handle fluctuating traffic conditions, often leading to bottlenecks and inefficiencies. This method uses real-time traffic data to balance loads dynamically, distributing traffic across multiple paths and ensuring stability. This leads to better network utilization, reduced congestion, and improved overall performance.
F. The experimental evaluation in the paper includes important metrics like bandwidth utilization, latency, packet loss, and throughput, allowing for a detailed assessment of network performance. This holistic approach to evaluation helps identify bottlenecks and further optimizes VN embedding for SDN environments.

3.4 Criteria for Performance Comparison

To compare the performance of the proposed scheme with other solutions in the context of this research work, the following benchmark algorithms or evaluation criteria have been considered:

A. Baseline Virtual Network Embedding Algorithms: By comparing against standard VNE algorithms such as Greedy Mapping, Node-Ranking-based Mapping, or Random Mapping. These approaches provide a baseline for resource allocation and can highlight the improvements made by the traffic expertise-based approach [36].
B. Network Performance Metrics: By evaluating using key performance indicators such as latency, throughput, and packet loss rate [37].
C. Resource Utilization and Scalability: Use metrics such as link and node utilization efficiency, load balancing index, and scalability under increased VNRs to compare resource optimization and scalability between schemes [38].

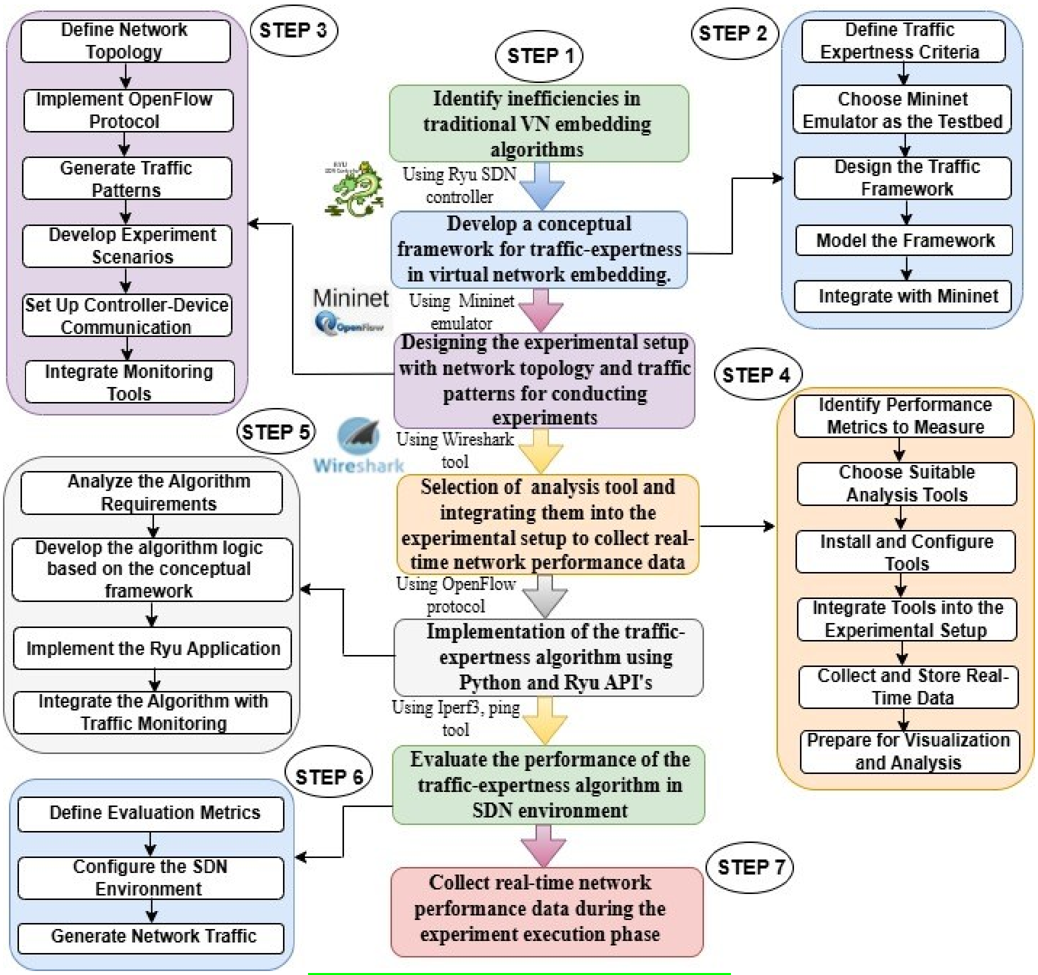

Fig. 4. Methodology of the proposed TEVN

3.5 Proposed TEVN Setup

The following section specifies the proposed TEVN setup for TE embedding VNs with traffic-expertness using SDN and the Ryu controller given in Fig. 5 and the flow of the package provided in Fig. 6. The TEVN setup to embed VNs described in the paper offers a novel approach to effectively commoditize VNs in dynamic network environments by relying on current traffic scenarios. As described by integrating the principles of SDN and the Ryu controller, the proposed algorithm enables VNs to be dynamically premeditated based on traffic awareness to ensure rapid resource allocation and network efficiency. More particularly, this section describes the algorithm step-by-step established on the critical input factors and judgment accomplished as shown. The final point is the class Interface. It can be thought of as the Network Topology. It is an entire map of all the paths the data can travel. Knowing this, the SDN Controller can select the lane to get data through. The last one that can be mentioned is the class Ryu Controller. This can be defined as the highway service team. It constantly keeps all lanes clean so everything flows without additional layers. It modifies the lanes to evade traffic jams and assures all the data reaches the aim with no queues. Finally, we can draw the analogy with the primary function, which is a dispatcher.

This method is used because it offers a comprehensive, adaptive approach to VN embedding that aligns with the needs of dynamic and complex SDN environments. This solution ensures efficient resource utilization, enhanced QoS, and security robustness by leveraging centralized control, real-time monitoring, adaptive cost functions, and dynamic load balancing. This approach is efficient for applications requiring real-time adaptability, minimal latency, and high scalability—such as cloud services, IoT, or mobile networks.

```
ubuntu@ubuntu-VirtualBox:~$ ryu-manager ryu.app.simple_switch_13
loading app ryu.app.simple_switch_13
loading app ryu.controller.ofp_handler
instantiating app ryu.app.simple_switch_13 of SimpleSwitch13
instantiating app ryu.controller.ofp_handler of OFPHandler
packet in 1 00:00:00:00:00:01 33:33:00:00:00:02 1
packet in 1 00:00:00:00:00:02 33:33:00:00:00:02 2
packet in 1 00:00:00:00:00:04 33:33:00:00:00:02 4
packet in 1 00:00:00:00:00:01 ff:ff:ff:ff:ff:ff 1
packet in 1 00:00:00:00:00:02 00:00:00:00:00:01 2
packet in 1 00:00:00:00:00:01 00:00:00:00:00:02 1
packet in 1 00:00:00:00:00:03 33:33:00:00:00:02 3
packet in 1 00:00:00:00:00:02 33:33:00:00:00:02 2
packet in 1 00:00:00:00:00:01 33:33:00:00:00:02 1
packet in 1 00:00:00:00:00:04 33:33:00:00:00:02 4
packet in 1 00:00:00:00:00:03 33:33:00:00:00:02 3
packet in 1 00:00:00:00:00:01 ff:ff:ff:ff:ff:ff 1
packet in 1 00:00:00:00:00:04 00:00:00:00:00:01 4
packet in 1 00:00:00:00:00:01 00:00:00:00:00:04 1
packet in 1 00:00:00:00:00:02 33:33:00:00:00:02 2
packet in 1 00:00:00:00:00:01 33:33:00:00:00:02 1
packet in 1 00:00:00:00:00:04 33:33:00:00:00:02 4
packet in 1 00:00:00:00:00:03 33:33:00:00:00:02 3
packet in 1 00:00:00:00:00:02 33:33:00:00:00:02 2
packet in 1 00:00:00:00:00:04 33:33:00:00:00:02 4
packet in 1 00:00:00:00:00:01 33:33:00:00:00:02 1
packet in 1 00:00:00:00:00:03 33:33:00:00:00:02 3
```

Fig. 5. Embedding Ryu controller in the TEVN network topology

```
ubuntu@ubuntu-VirtualBox:~$ sudo ovs-ofctl -O OpenFlow13 dump-flows s1
[sudo] password for ubuntu:
 cookie=0x0, duration=16936.351s, table=0, n_packets=13, n_bytes=1106, priority
=1,in_port="s1-eth2",dl_src=00:00:00:00:00:02,dl_dst=00:00:00:00:00:01 actions=
output:"s1-eth1"
 cookie=0x0, duration=16936.325s, table=0, n_packets=11, n_bytes=966, priority=
1,in_port="s1-eth1",dl_src=00:00:00:00:00:01,dl_dst=00:00:00:00:00:02 actions=o
utput:"s1-eth2"
 cookie=0x0, duration=16284.301s, table=0, n_packets=538713, n_bytes=6249809658
, priority=1,in_port="s1-eth4",dl_src=00:00:00:00:00:04,dl_dst=00:00:00:00:00:0
1 actions=output:"s1-eth1"
 cookie=0x0, duration=16284.292s, table=0, n_packets=591410, n_bytes=2526305307
6, priority=1,in_port="s1-eth1",dl_src=00:00:00:00:00:01,dl_dst=00:00:00:00:00:
04 actions=output:"s1-eth4"
 cookie=0x0, duration=6788.255s, table=0, n_packets=94, n_bytes=7076, priority=
0 actions=CONTROLLER:65535
```

Fig. 6. Packet flow of the proposed TEVN network topology

4. RESULT AND ANALYSIS

4.1 Performance evaluation

In this paper, the efficacy of our proposed algorithm is compared to the following methods as benchmarks.

A. Benchmark 1: This integration of traffic expertise into the virtual network embedding process allows the proposed algorithm to allocate resources optimally and find the best routing paths that adapt to traffic changes. As a result, the SDN controller can adequately manage traffic and balance loads and resources to ensure that the network resources are decently used according to the network's current demands. The experimental assessments show the effectiveness of the introduced solution in dealing with network traffic variations and prove that the SDNV algorithm enhances network performance, scalability, and adaptability in SDN networks [39].

B. Benchmark 2: The study presented an original method to embed virtual networks using SDN and the Ryu controller, using real-time traffic awareness. By optimizing the virtual network topology in response to traffic data, the system enhances the quality of service while improving routing algorithms and resource distribution. Since the SDN controller tracks network status and changes routing paths in reaction to it regularly, congestion and latency are continually minimized because the research method continuously loops its phases to improve the outcomes [40].

C. Benchmark 3: This adaptive mechanism ensures that virtual networks may be developed and reconfigured dynamically according to altering traffic patterns and network infrastructure demands. Via the centralized control enabled by SDN and the expertness techniques in traffic used in the embedding process, the study allows for acquiring highly adaptable virtual networks that may efficiently adjust to shifting network traffic for maximizing optimization [41].

4.2 Experimental results

This section presents the experimental results of evaluating the proposed methodology for virtual network embedding with traffic awareness using SDN and the Ryu controller. The experiments assessed the system's performance across various key performance parameters. These parameters include network latency, throughput, bandwidth utilization, packet loss, and resource allocation efficiency. By measuring and analyzing these metrics, we aim to gain insights into the effectiveness and efficacy of the proposed approach in optimizing network performance and ensuring QoS in dynamic network environments.

4.2.1 TEVN Bandwidth

For evaluating the Ryu controller performance regarding bandwidth, we have used the iperf3 benchmark utility, one of the most commonly used utilities to generate TCP traffic. Benchmarking was done for TCP bandwidth between the nodes in the network's topology. The iperf3 command was run for 10 seconds. The TCP bandwidth results are graphed in Fig. 7.

It is to be noted that the tables contain experimental results from the research, which provide indicative data transmission and bandwidth utilization between various nodes in the network. In each table, the row indicates the communication link between two nodes identified as h1 to h4. The “Data transmit KB” column presents the amount of data transferred between the given nodes, denoted in kilobytes. For example, from node h1 to h2, the total data transmitted amounted to 3,223,450 KB. In the “TEVN Bandwidth Gbps” column, the bandwidth utilization for the TEVN between the given nodes is identified in gigabits per second. It assesses the extent of consumed bandwidth in the network based on the aspects of traffic efficiency provided by the research.

The table compares the default bandwidth and the bandwidth achieved using TEVN between different nodes in the network, as shown in Fig. 8. In the default scenario, the bandwidth between nodes "h1" and "h2" is 25.1 Gbps. In contrast, with TEVN, it increases to 26.7 Gbps. Similarly, the bandwidth between nodes "h1" and "h3" improves from 24.5 Gbps to 25.2 Gbps with TEVN. Node "h1" to "h4" shows an enhancement from 24.4 Gbps to 25 Gbps. For nodes "h2" to "h3," "h2" to "h4," and "h3" to "h4," where the default bandwidth is only 0.2 Gbps due to a bottleneck, TEVN significantly boosts the bandwidth to 24.8 Gbps, 23.8 Gbps, and 22.7 Gbps, respectively. This enhancement demonstrates the effectiveness of TEVN in optimizing bandwidth allocation and improving network performance between various nodes.

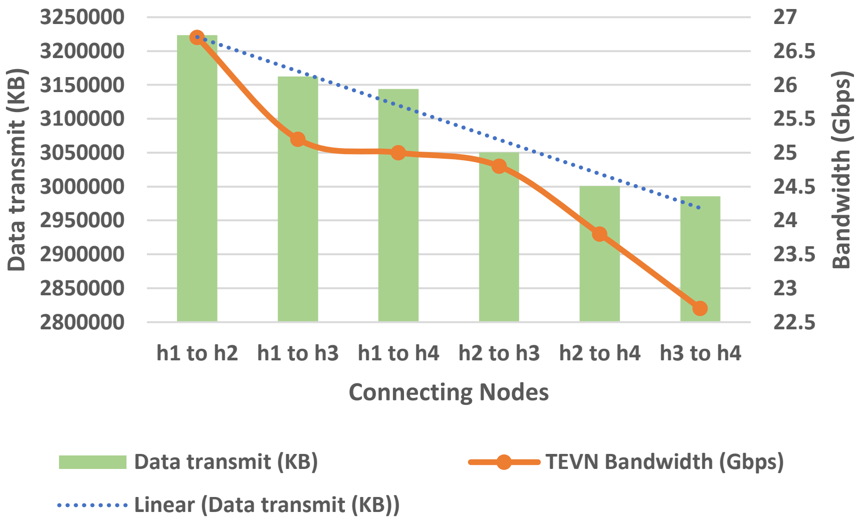

Fig. 7. Bandwidth of TEVN

4.2.2 TEVN Throughput

The throughput evaluation is dedicated to defining how much data could be processed by the controller from one node to another. This included determining the level of data that the two node points could support to be transferred or received per second or data packets per second. The iperf3 was used for this evaluation, and the utility running the TCP connection – iperf3- was executed on the client-side one and lasted 10 seconds. The data was collected from the server side on a one-second basis. The results of this model are presented in Fig. 9.

The above figure gives the throughput (in Gbps) observed between node pairs in the network. The rows correspond to the source nodes h1, h2, and h3, whereas the columns correspond to the destination nodes h2, h3, and h4, respectively. The entries in each cell represent the throughput of data observed from the source node to the destination node in the figure above. For example, the data observed for the first row of h1 to h2 is 20.9 Gbps, 20.2 Gbps, 27.3 Gbps, and 28.5 Gbps. Here, each value indicates the variation in the throughput observed from the source node to the destination node.

Similarly, the second row represents the throughput observed for data transmission from node "h1" to node "h3." The values in this row range from 20.9 Gbps to 28.5 Gbps, indicating the varying throughput observed between these two nodes under different network conditions.

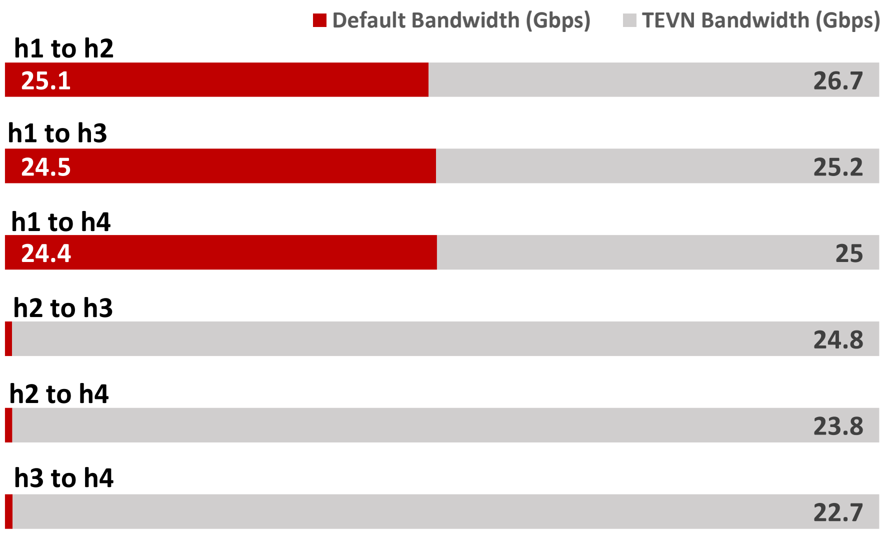

Fig. 8. Comparison between the default bandwidth and the bandwidth achieved using TEVN between different nodes in the network

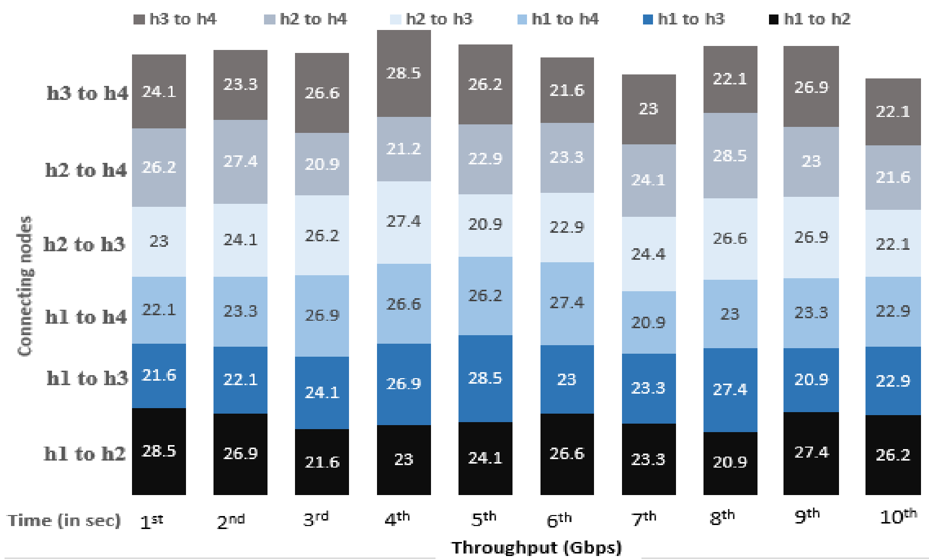

Fig. 9. Throughput of TEVN

4.2.3 TEVN Packet Loss

Packet loss minimization is critical for virtual network embedding with traffic awareness through SDN and the Ryu controller, as the solution is designed for fast and reliable data transmission. The control will rely on dynamic routing and traffic management to reduce packet loss instances while maintaining optimal network performance. The solution will ensure low packet loss rates with real-time monitoring and adaptive control, improving QoS overall.

Table 3. TEVN Packet loss at two different bandwidths.

Connecting Nodes	Packet Loss at 10 Mbps	Packet Loss at 50 Mbps
H1 to H2	0.015	0.049
H1 to H3	0.44	0.099
H1 to H4	0.035	0.07
H2 to H3	0.012	0.038
H2 to H4	0.4	0.098
H3 to H4	0.02	0.059

Table 3 and Fig. 10 below summarize the packet loss measurement between node pairs at two bandwidth settings, including 10 Mbps and 50 Mbps. Each cell shows the packet loss values between the paired nodes at a designated bandwidth level.

For example, between nodes h1 and h2, the packet loss rate is 0.015 when the bandwidth is 10 Mbps and 0.049 when the bandwidth is 50 Mbps. Similarly, each row represents a source node, each column represents a destination node, and the values indicate the packet loss rates observed during the experiment. These measurements help assess the impact of varying bandwidth on packet loss in the network topology.

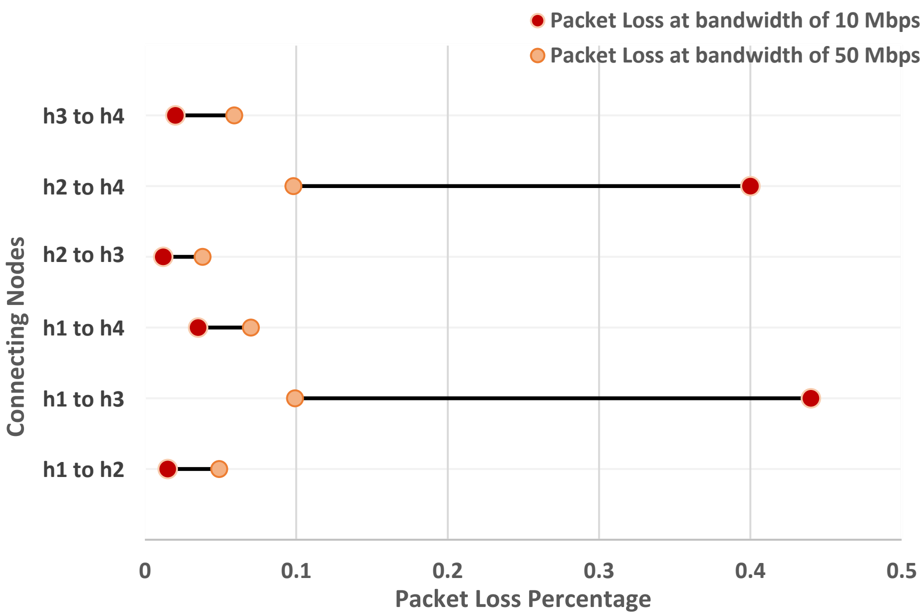

Fig. 10. Visual representation of TEVN packet loss at two different bandwidths

4.2.4 TEVN RTT

The controller measured RTT, commonly called ping time, which signifies a packet's duration to travel to a specific destination and return. This time encompasses the propagation delays between a signal's transmitting and receiving points. The ping utility utilizes ICMP to dispatch message packets for error reporting and other purposes. In this simulation, the ping command was executed on the client side to gauge RTT, where the client node dispatched ICMP echo requests to designated server nodes and awaited the corresponding echo reply packets. Ten ICMP packets were transmitted from node h1 to node h2, h1 to h3, and h2 to h3. The RTT was measured for each packet, and the minimum, maximum, and average values were determined. Fig. 11 illustrates three distinct scenarios, depicting minimum RTTs of 0.061 ms, 0.055 ms, and 0.058 ms; average RTTs of 1.411 ms, 1.050 ms, and 0.918 ms; and maximum RTTs of 10.305 ms, 9.668 ms, and 8.149 ms.

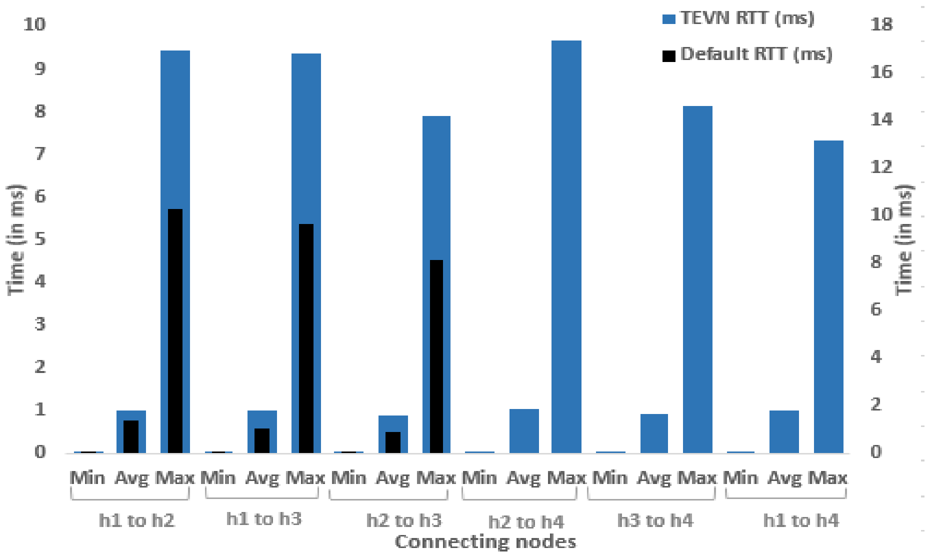

Fig. 11. Three distinct scenarios depicting minimum, average, and maximum TEVN's RTT

5. CONCLUSION AND FUTURE SCOPE

In summary, this study presented a comprehensive investigation of the performance of TEVN embedding using the Ryu controller in an SDN environment. In summary, this research evaluated the performance of TEVN concerning network performance and flexibility, particularly when combined with the Ryu controller. This was achieved through extensive testing and examination of the assessed parameters, primarily latency, throughput, packet loss, and RTT. These findings demonstrated that TEVN's incorporation with the Ryu controller is superior to previous approaches and boosts network performance, resource utility, and quality of service. Based on this, this research demonstrates the significant role of his TEVN in dealing with the inconsistent nature of network traffic and strengthening the capabilities of SDN technology. To conclude, his study illustrates the considerable promise of TEVN in reorganizing network management and optimization in SDN surroundings.

We showed that by using TEVN and Ryu controllers, all network performance metrics have significantly improved, demonstrating the method's efficiency in satisfying the changing demands of current network infrastructures. In the future, it is worth continuing the research efforts to improve the TEVN algorithm, carry out scalability checks, and observe practical implementation cases to validate their immediate relevance to the network. Overall, this research contributes to the continued development of his SDN technology and paves the way for future network optimization and efficiency advances. Future work can focus on extending the framework to accommodate dynamic and heterogeneous traffic patterns in large-scale, real-world SDN deployments. Incorporating advanced machine learning models for predictive analytics could enable proactive decision-making and more efficient resource allocation in media transmission [42].

REFERENCES

1. Rezaee, M. R., Hamid, N. A. W. A., Hussin, M., & Zukarnain, Z. A. (2024). Fog Offloading and Task Management in IoT-Fog-Cloud Environment: Review of Algorithms, Networks and SDN Application. IEEE Access.
2. Ayodele, B., & Buttigieg, V. (2024). SDN as a defense mechanism: a comprehensive survey. International Journal of Information Security, 23(1), 141-185.
3. Abu-Ain, T., Ahmad, R., Wazirali, R., & Abu-Ain, W. (2023). A new SDN-handover framework for QoS in heterogeneous wireless networks. Arabian Journal for Science and Engineering, 48(8), 10857-10873.
4. Qaffas, A. A., Kamal, S., Sayeed, F., Dutta, P., Joshi, S., & Alhassan, I. (2023). Adaptive population-based multi-objective optimization in SDN controllers for cost optimization. Physical Communication, 58, 102006.
5. Bhardwaj, S., & Girdhar, A. (2023). Network Traffic Analysis in Software-Defined Networking Using RYU Controller. Wireless Personal Communications, 132(3), 1797-1818.
6. Ramya, G., & Manoharan, R. (2023). Traffic-aware dynamic controller placement in SDN using NFV. The Journal of Supercomputing, 79(2), 2082-2107.
7. Pei, X., Sun, P., Hu, Y., Li, D., Chen, B., & Tian, L. (2024). Enabling efficient routing for traffic engineering in SDN with Deep Reinforcement Learning. Computer Networks, 241, 110220.
8. Etengu, R., Tan, S. C., Chuah, T. C., Lee, Y. L., & Galán-Jiménez, J. (2023). AI-assisted traffic matrix prediction using GA-enabled deep ensemble learning for hybrid SDN. Computer Communications, 203, 298-311.
9. Belkadi, O., Vulpe, A., Laaziz, Y., & Halunga, S. (2023). ML-Based Traffic Classification in an SDN-Enabled Cloud Environment. Electronics, 12(2), 269.
10. Gunavathie, M. A., & Umamaheswari, S. (2024). Traffic-aware optimal routing in software-defined networks by predicting traffic using a neural network. Expert Systems with Applications, 239, 122415.
11. Wang, K., Fu, Y., Duan, X., Liu, T., & Xu, J. (2024). An abnormal traffic detection system in SDN is based on deep learning hybrid models. Computer Communications, 216, 183-194.
12. Anitha, H. M., Jayarekha, P., Sivaraman, A., Mehta, A., & Nalina, V. (2024). SDN Enabled Role Based Shared Secret Scheme for Virtual Machine Security in Cloud Environment. Cyber Security and Applications, 100043.
13. Li, J., Qi, X., Li, J., Su, Z., Su, Y., & Liu, L. (2024). Fault Diagnosis in the Network Function Virtualization: A Survey, Taxonomy and Future Directions. IEEE Internet of Things Journal.
14. Xiao, H., Xu, C., Ma, Y., Yang, S., Zhong, L., & Muntean, G. M. (2022). Edge intelligence: A computational task offloading scheme for dependent IoT application. IEEE Transactions on Wireless Communications, 21(9), 7222-7237.
15. Ramya, G., & Manoharan, R. (2023). Traffic-aware dynamic controller placement in SDN using NFV. The Journal of Supercomputing, 79(2), 2082-2107.
16. Núñez-Gómez, C., Carrión, C., Caminero, B., & Delicado, F. M. (2023). S-HIDRA: A blockchain and SDN domain-based architecture to orchestrate fog computing environments. Computer Networks, 221, 109512.

17. Xiao, H., Zhuang, Y., Xu, C., Wang, W., Zhang, H., Ding, R. & Muntean, G. M. (2023). Transcoding-enabled cloud–edge–terminal collaborative video caching in heterogeneous IoT networks: an online learning approach with time-varying information. IEEE Internet of Things Journal, 11(1), 296-310.
18. Dimolitsas, I., Dechouniotis, D., & Papavassiliou, S. (2023). Time-efficient distributed virtual network embedding for round-trip delay minimization. Journal of Network and Computer Applications, 217, 103691.
19. M. Lu, Y. Gu and D. Xie, "A Dynamic and Collaborative Multi-Layer Virtual Network Embedding Algorithm in SDN Based on Reinforcement Learning," in IEEE Transactions on Network and Service Management, vol. 17, no. 4, pp. 2305-2317, Dec. 2020, doi: 10.1109/TNSM.2020.3012588
20. R. Chai, D. Xie, L. Luo and Q. Chen, "Multi-Objective Optimization-Based Virtual Network Embedding Algorithm for Software-Defined Networking," in IEEE Transactions on Network and Service Management, vol. 17, no. 1, pp. 532-546, March 2020, doi: 10.1109/TNSM.2019.2953297.
21. Xiao, H., Huang, Z., Xu, Z., Yang, S., Wang, W., Zhong, L., & Xu, C. (2024). Task-driven cooperative internet of the robotic things crowdsourcing: from the perspective of hierarchical game-theoretic. IEEE Internet of Things Journal.
22. Xiao, H., Xu, C., Feng, Z., Ding, R., Yang, S., Zhong, L., ... & Muntean, G. M. (2022). A transcoding-enabled 360 VR video caching and delivery framework for edge-enhanced next-generation wireless networks. IEEE Journal on Selected Areas in Communications, 40(5), 1615-1631.
23. Bhardwaj, S., & Girdhar, A. (2023). Network traffic analysis in software-defined networking using Ryu controller. Wireless Personal Communications, 132(3), 1797-1818.
24. Bhardwaj, S., Panda, S.N. Performance Evaluation Using RYU SDN Controller in Software-Defined Networking Environment. Wireless Pers Commun 122, 701–723 (2022). https://doi.org/10.1007/s11277-021-08920-3
25. Chouhan, R. K., Atulkar, M., & Nagwani, N. K. (2023). A framework to detect DDoS attacks in Ryu controller-based software-defined networks using feature extraction and classification. Applied Intelligence, 53(4), 4268-4288.
26. Revathi, M., Ramalingam, V. V., & Amutha, B. (2022). A machine learning based detection and mitigation of the DDOS attack by using SDN controller framework. Wireless Personal Communications, 1-25.
27. Song, S., Park, H., Choi, B. Y., Choi, T., & Zhu, H. (2017). Control path management framework for enhancing software-defined network (SDN) reliability. IEEE Transactions on Network and Service Management, 14(2), 302-316.
28. Naim, N., Imad, M., Hassan, M. A., Afzal, M. B., Khan, S., & Khan, A. U. (2023). POX and RYU Controller Performance Analysis on Software Defined Network. EAI Endorsed Transactions on Internet of Things, 9(3).
29. Tang, D., Zheng, Z., Yin, C., Xiong, B., Qin, Z., & Yang, Q. (2024). FTODefender: An efficient flow table overflow attacks defending system in SDN. Expert Systems with Applications, 237, 121460.
30. Uddin, R., & Kumar, S. (2023). Sdn-based federated learning approach for satellite-iot framework to enhance data security and privacy in space communication. IEEE Journal of Radio Frequency Identification.

31. Halman, L. M., & Alenazi, M. J. (2024). Threshold-Based Software-Defined Networking (SDN) Solution for Healthcare Systems against Intrusion Attacks. CMES-Computer Modeling in Engineering & Sciences, 138(2).
32. Monir, M. F., & Hasan, A. F. (2024, April). Exploring SDN-Based Firewall and NAPT: A Comparative Analysis with iptables and OVS in Mininet. In International Conference on Advanced Information Networking and Applications (pp. 436-447). Cham: Springer Nature Switzerland.
33. Arthi, R., Krishnaveni, S., & Zeadally, S. (2024). An intelligent SDN-IoT enabled intrusion detection system for healthcare systems using a hybrid deep learning and machine learning approach. China Communications.
34. Pei, X., Sun, P., Hu, Y., Li, D., Chen, B., & Tian, L. (2024). Enabling efficient routing for traffic engineering in SDN with Deep Reinforcement Learning. Computer Networks, 241, 110220.
35. Dash, B. B., Satpathy, R., & Patra, S. S. (2024). Efficient SDN-based Task offloading in a fog-assisted cloud environment. EAI Endorsed Transactions on Internet of Things, 10.
36. Yao, H., Chen, X., Li, M., Zhang, P., & Wang, L. (2018). A novel reinforcement learning algorithm for virtual network embedding. Neurocomputing, 284, 1-9.
37. Yan, Z., Ge, J., Wu, Y., Li, L., & Li, T. (2020). Automatic virtual network embedding: A deep reinforcement learning approach with graph convolutional networks. IEEE Journal on Selected Areas in Communications, 38(6), 1040-1057.
38. Zhang, P., Chen, N., Li, S., Choo, K. K. R., Jiang, C., & Wu, S. (2023). Multi-domain virtual network embedding algorithm based on horizontal federated learning. IEEE Transactions on Information Forensics and Security, 18, 3363-3375.
39. Duan, Z., & Wang, T. (2024). Towards learning-based energy-efficient online coordinated virtual network embedding framework. Computer Networks, 239, 110139.
40. Sanchez, L. P. A., Shen, Y., & Guo, M. (2024). DQS: A QoS-driven routing optimization approach in SDN using deep reinforcement learning. Journal of Parallel and Distributed Computing, 188, 104851.
41. Minardi, M., Vu, T. X., Maity, I., Politis, C., & Chatzinotas, S. (2024). Traffic-Aware Virtual Network Embedding With Joint Load Balancing and Datarate Assignment for SDN-Based Networks. IEEE Transactions on Network and Service Management.
42. Xiao, H., Xu, C., Fang, C., Yang, S., & Zhong, L. (2024, April). VAAC-IM: Viewing Area Adaptive Control in Immersive Media Transmission. In Proceedings of the 34th edition of the Workshop on Network and Operating System Support for Digital Audio and Video (pp. 8-14).