

SDN multi-access edge computing for mobility management

Sri Ramachandra Lakkaiah¹, Hareesh Kumbhinarasaiah²

¹Department of Computer Science and Engineering, Government SKSJTI, Bengaluru, India

²Department of Computer Science and Engineering, K. R. Pet Krishna Government. Engineering College, Mandya, India

Article Info

Article history:

Received Dec 2, 2024

Revised Apr 8, 2025

Accepted Jul 2, 2025

Keywords:

Internet of things

Multi access edge computing

Mobile edge hosts mobility

Software defined networking

ABSTRACT

In recent trends, multi-access edge computing (MEC) is becoming a realistic framework for extensive social networking. The rapid proliferation of internet of things (IoT) devices has led to an unprecedented increase in data generation, placing significant strain on conventional cloud computing infrastructure. MEC also supports ultra-reliable and low latency communications (URLLC) by delivering information and computational resources more quickly to mobile users. As a result, the need for low-latency and reliable communication has become paramount. This paper proposes an MEC architecture that integrates software defined networking (SDN) and virtualization techniques, where MEC enables the orchestration and organization of mobile edge hosts (MEH). Furthermore, the proposed MEC-SDN design minimizes latency while ensuring consistent ultra-low latency communications. The result analysis clearly demonstrates that the proposed MEC-SDN model achieves latency of 6-14 ms, bandwidth of 5.2 Mbits/sec, and SDN-BWMS of 5.4 Mbits/sec, outperforming the existing SDN-Mobile Core Network model. Mobile edge systems are enabled in this research to provide mobility support for users.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Sri Ramachandra Lakkaiah

Department of Computer Science and Engineering, Government SKSJTI

Bengaluru, India

Email: dsram8388@gmail.com

1. INTRODUCTION

SDNs are novel network architectures that feature a separate data plane and a control plane for management, along with a logically centralized controller [1]. SDN can access global topology through SDN controllers which make forwarding decisions based on flow tables [2]. The centralized logical controller maintains a global view of the entire network, enabling the controller to make better decisions than traditional research [3]. Abstraction has enabled the network to manage applications deployed over the controller for open flow switches, preventing vendor lock-in [4]. The SDN controllers include platforms that support dynamic access control, server load balancing, network virtualization, and green network architecture [5], [6]. The routing mechanism is a crucial aspect of network resource optimization, and it is challenging to achieve optimal performance with distributed routing approaches [7]. Traditional networks, with their uncontrollable distributed routing schemes, often result in wasted resources [8]. SDNs offer insights into network routing schemes, with existing methods utilizing deep learning models that have outperformed traditional machine learning approaches [9]. The centralized controller in SDNs manages network resources efficiently and supports traffic control, enhancing network routing performance [10]. The demand for traffic has increased dramatically, especially with network applications. SDN networks handle large volumes of traffic demands arriving within short periods, necessitating a centralized control plane to complete path assignments [11]. Therefore, traffic and network sizes have increased significantly in the last decade, with

centers now including hundreds of switches [12]. SDN techniques enable the network to operate at near-full capacity, implying that network controllers are required for re-optimization processes [13]. Significant changes in traffic, such as those caused by network failures, require efficient and fast routing optimization techniques to manage large volumes of traffic demands [14]. Addressing the challenges posed by large-scale SDN models is critical [15].

Shah *et al.* [16] developed an end-to-end mobility support architecture that integrated SDN with cloud-native virtualization models, which consisted of containers with multi-access edge computing (MEC) architecture. This facilitated orchestration and management of mobile edge hosts (MEH). The developed model focused primarily on providing end-to-end support in terms of mobility. It showed improvements in performance, specifically in bandwidth and latency. The SDN-MEC model required to maintain continuous service as mobile users were transitioned from one process to another. Additionally, the resource-efficient design of the model mitigated resource constraints imposed by the mobile network bandwidth limitations. This resulted in decreased performance, increased latency, and reduced overall system efficiency. Silva *et al.* [17] developed a hybrid software-defined networking for 4G and 5G. SDN brought mobile networks from network operators with multiple end-to-end users. The hybrid SDN-mobile core network (MCN) operated with 4G and 5G, integrated with support for traffic offloading and capabilities from both 3GPP and non-3GPP technologies. The developed model also handled mobility with IP flows, Wi-Fi latency, LTE throughput, and Wi-Fi throughput. The integration of 4G and 5G networks with Wi-Fi and other non-3GPP technologies introduced complexity in managing user mobility. Miriyala and Sairam [18] developed a fully homomorphic encryption algorithm, which was a simple dual-key model designed for encryption within a complex hybrid structure. The simple dual-key method was used to perform fully homomorphic encryption, which supported the complex hybrid structure. This scheme incorporated a double decryption method using fully homomorphic encryption. The model was implemented with these schemes, and SDN controllers were placed in the MANET environment. SDN enabled the administrator to handle network security through the controller, providing continuous protection for the entire network and device data. As the network size increased, the computational overhead and key management complexity also increased, potentially leading to reduced performance and increased security vulnerabilities.

Devi and Jaison [19] developed an SDN-based model for performing a mechanism that was implemented on a network route for analysis. The SDN implementation was based on the WSN mechanism, which helped improve and maintain the QoS under constraints. The hybrid clone node detection (HCND) model detected cloned nodes in the wireless network. An efficient clone detection model was used to proactively eliminate clone attacks. Therefore, clones were detected locally through geographical regions, making the verification process cost-effective. However, energy efficiency was limited by the distributed protocol, which impacted the performance analysis. Additionally, the limited energy resources of WSN nodes worsen this issue, leading to reduced detection accuracy and increased false positives. Khan and Akhunzada [20] developed a highly scalable and efficient hybrid DL-based SDN model for detecting malware using the framework.

The developed model leveraged IoT resources that were constrained in the devices without exhausting them. The dataset used was from a publicly available source. The proposed technique showed better performance in terms of standard metrics compared to existing models. The model exhibited low computational complexity but posed challenges for devices with limited processing capabilities. This resulted in delayed detection and response times while compromising the security and integrity of the data. Min *et al.* [21] demonstrated an SDN-Orchestrated artificial intelligence-empowered framework for intrusion detection in cyber-physical systems. This research paper presented a Cu-BLSTMGRU detection framework for IoT-based industrial networks with limited resources, which was driven by DL and influenced by SDN. With the combined power of these two advanced technologies, the former was linked to the anomaly detection segment, and the latter was responsible for resource allocation to optimize the framework's compatibility for resource-constrained networks, maximizing the system's capability. Furthermore, SDN provided DL with unified operational integration, enhancing the threat detection capabilities of the DL-based anomaly detection framework. However, it remained challenging to propose a security solution for IoT setups with limited resources. Additionally, the computational complexity of Cu-BLSTMGRU and the communication overhead of SDN did not compromise energy efficiency.

A hybrid framework combining SDN controllers with DL approaches namely, convolutional neural networks (CNN) and bidirectional long short-term memory (BiLSTM), was proposed by Rbah *et al.* [22]. This method presented a special combination that made IoMT security management dynamic and effective. The system provided a comprehensive threat detection solution by handling various IoMT data formats through the integration of CNN and Bi-LSTM. The security landscape for healthcare IoT systems was dynamic and this hybrid solution adjusted with ease. Because of the complexity of IoMT networks, this research necessitated investigation into a hybrid SDN-based DL framework. However, the suggested DL system demanded more time for model training due to feature self-learning and weight adaptation.

Furthermore, as IoMT networks expanded, the framework's scalability and resource intensity grew concerns which caused an increment in latency and energy consumption. An effective enhanced crowd search algorithm with DL-driven cyberattack detection (ECSADL-CAD) model for SDN-enabled IoT environment was presented by Motwakel *et al.* [23]. The primary aim of ECSADL-CAD was to recognize and categorize cyberattacks in the SDN-enabled IoT. In order to achieve this, the data was pre-processed using the ECSADL-CAD model, after which the reinforced deep belief network (RDBN) model was used to detect attacks. Finally, the ECSA-based hyperparameter procedure was carried out for improved security. The computational intensity of ECSADL-CAD resulted in increased energy consumption, particularly in resource-constrained IoT devices. The contributions of this research are noted as follows:

- This research proposes MEC structure that combines SDN and cloud-native virtualization methods. The MEC method is employed to enable the orchestration of MEH.
- The developed structure primarily focuses on end-to-end mobility support, which is essential to maintain service continuity when mobile users transfer data to one another.
- To handle mobility support, mobile edge systems are provided to users, specifically measured in terms of latency, bandwidth, and SDN-BWMS.

The further organization of the research is as follows: section 2 outlines the proposed methodologies and the block diagram used in SDN networks. Section 3 details the integration of MEC-SDN for mobility management, while section 4 presents the results and discussion of the proposed method and finally, section 5 concludes the research.

2. PROPOSED METHOD

The block diagram of the MEC-SDN model is shown in Figure 1. It consists of the following steps: SDN integration, which includes QoS management and UE mobility management, dynamic routing, and ME app mobility. The integrated part is connected to a northbound interface that includes a controller and open flow simulator. The steps involved in network initialization are the orchestrator, RYU controller, and nodes, which are described as follows:

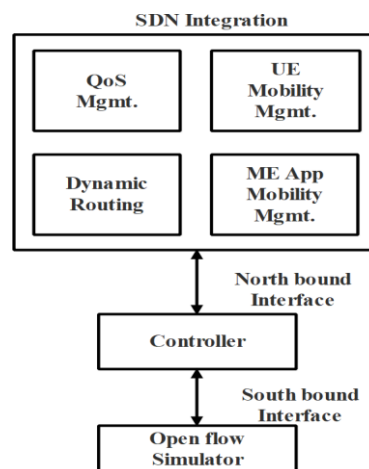


Figure 1. Block diagram of the MEC-SDN model

2.1. Orchestrator

Mobile edge computing (MEC) represents a significant advancement in cloud services [24], offering dynamically constructed dedicated services based on specific request types and contexts. However, transportation services and contexts are not always installed at the nearest MEC, which presents a challenge by primarily reducing the massive bandwidth usage to the cloud while improving the connection speed to the user. MEC is deployed at a low cost, utilizing resource-constrained, affordable devices. However, these devices suffer from security issues, and the hardware platform lacks security mechanisms. The synchronization among the MEH produces the network which is related to the conclusion that directs towards an update [25].

The MEH plays a crucial role in the MEC architecture by maintaining the overall view of the MEH. It is organized within the MEC framework, which consists of all the topology data arranged in the MEH. The

ME services and available resources are structured in such a way that the MEO must select the most suitable MEH for executing application relocation. This decision is based on various constraints, including available services, latency, and resources, among others.

2.2. RYU controller

The RYU controller provides support for the latest open flow structures. The RYU controller [26] is known as an open SDN designed to enhance network operations. Meter tables are implemented, which limit the rate but allow distinct services, as they receive support from the controller. The potential benefits are enhanced with the migration service, applied to a realistic scenario where the DRS runs as a container over workstations. The RYU controller is an SDN controller designed to increase network agility, enabling it to manage and adapt traffic accordingly. The SDN controller operates in an environment primarily focused on communicating information, which is relayed to the routers and switches. The RYU controller supports NTT, which includes cloud data centers and relevant software components defined in application programming interfaces (APIs). This makes it easier for developers to create and manage new networks and control applications. This approach helps organize and customize components, deploying them to meet specific needs. Developers can quickly modify existing components by implementing their own solutions to ensure the underlying network meets the application's demands.

2.3. Nodes

There are two nodes divided substantially from each other which introduced the utility of linux traffic control (LTC). Linux offers a rich set of tools for managing, transmitting, and manipulating packets. The larger Linux community is familiar with these tools, which are used to manage and firewall packets. The model provides hundreds of network services that run the system. While many within the Linux community are aware of the power offered by the control subsystem for traffic. The LTC manages the scheduler used to replicate an accurate MEH deployment model. Several works have focused on facilitating the relocation among MEC nodes, transferring data from conventional nodes. Various fundamental networks related to MEH enable low-latency services. MEC traces the radio access network (RAN) and traffic control (TC), which requires the Linux Utility [27]. The model gains the ability to configure the kernel scheduler, offering distinct options that are familiar features of Linux. Linux utilizes simulation tools to show packet delay for both UDP and TCP applications. The developed model limits the bandwidth usage specific to services, simulating internet connections such as Cable, T1, and DSL. Consequently, Debian Linux has emerged, bundling the IP route for running the installed applications.

Every network administrator has access to the traffic control command, a tool that does not allow the admin to configure the kernel packet scheduler or simulate packet delay. It shows packet loss for UDP/TCP applications or limits bandwidth usage for a specific service. Through this process, better testing must be performed for applications with poor configurations and inconsistent networks.

3. MOBILITY MANAGEMENT BASED ON MEC-SDN

Mobility management is broadly defined as the process of creating and managing mobility at both the system and customer levels, with the goal of improving efficiency, affordability, and public transportation. The mobility features in MEC prevent the fast path that reserves the distribution process, incorporating sections for service migration. Service migration, particularly in real-time services, requires mobility features within MEC to address the complexities of handover management. To this end, SDN is developed as a mobility management application that transfers MEH along with ME applications to ensure low-latency procedures. The SDN application utilizes ME services offered by MEP, such as RNIS and LS, to collect updated data from the radio base station based on channel conditions, which in turn load the MEH for service migration [28].

3.1. Migration management

Service migration management is used in SDN applications to enable low latency and ensure seamless application procedures. The SDN controller displays radio network information, with RNIS and LS services remaining continuous. The model organizes and anticipates the need for application relocation, triggering the migration procedure. The MEP notifies the controller of the target MEH to instantiate the applications. Various communication methods exist between the SDN controller and the target MEP. Migration strategies are commonly applied across campus, enterprise, WAN, carrier, and service provider networks, with best practices based on initial experiences and documented solutions. These practices are divided into separate migration and pre-migration planning phases, where migration is impacted by services available during the pre-migration phase. Anticipated disruptions require alternative solutions, and post-

migration and pre-migration checklists ensure continuous service. In case of unexpected issues, procedures should be in place to revert to the original network to resolve migration problems or service degradation.

The migration phase is recommended for operators who manage the entire network and troubleshoot the migrated network. It involves timely upgrades and version control for protocols, such as OpenFlow. A dummy service is continuously created to check service availability as the migration process occurs. The model minimizes or prevents service disruptions by utilizing migration tools and operational tools. Ideally, the migration process exploits the benefits of SDN based on deployment. There are three categories of tools: monitoring, management, and configuration tools for verification and testing. These tools are used to detect, quantify, and report on network performance, ensuring quality of service. Configuration and management tools assist with migration-related configurations and provide support for rollback. Verification and testing tools are employed to validate OpenFlow controller and switch implementations. Several commercial and open-source tools are available to perform migrations from SDN. The evaluation of these tools is based on their ability to meet migration requirements, including support for protocol versions by multiple vendors.

4. RESULTS AND DISCUSSION

The performance evaluation is conducted on my system equipped with 8GB of RAM and an Intel Core i5 processor. The performance of the proposed research is evaluated under distinct system settings, focusing on bandwidth. The bandwidth of the model ranges from 50 Mbps to 100 Mbps, simulating network congestion conditions. Two nodes are physically separated to introduce delay, using utility information from linux traffic control (TC). Linux TC configures the kernel packet scheduler, realistically replicating the deployment scenario of MEH. The following parameters are used to evaluate the performance of the MEC-SDN model.

a) Latency

Time delay, referred to as the cause and effect of physical changes in the system, is known as latency (L). In a network, it typically represents the time taken for data to reach its destination. Thus, latency is expressed as (1).

$$L = \text{Time to send request} + \text{time to receive response} \quad (1)$$

b) Bandwidth

The maximum amount of data transmitted in each time interval is referred to as bandwidth (B). Additionally, bandwidth is often incorrectly associated with internet speed, as it is essentially focused on the volume of information. Thus, bandwidth is expressed as (2).

$$B = \frac{\text{total data transferred}}{\text{time taken}} \quad (2)$$

c) SDN-BWMS

The SDN-bandwidth management system (SDN-BWMS) is a network management framework that applies SDN principles to efficiently allocate and manage bandwidth resources in a network. SDN-BWMS optimizes bandwidth utilization, ensures QoS, and provides scalable and flexible bandwidth management. The efficiency of SDN-BWMS is calculated using bandwidth utilization efficiency (BUE), which is expressed as (3).

$$BUE = \left(\frac{\text{total used bandwidth}}{\text{total available bandwidth}} \right) \times 100 \quad (3)$$

4.1. Quantitative analysis

This section presents the results of the MEC-SDN model in terms of latency, bandwidth, and SDN-BWMS. The current research moved away from the coverage area, with the network operating over a period of 0 to 60 seconds, demonstrating lower latency compared to conventional cases. Figure 2 provides a graphical representation of the latency evaluation when compared to the existing and MEC-SDN models.

The existing model demonstrated an improvement in migration performance, as the SDN controller was initiated with the migration service and triggered to MNO C, which showed a higher computational load compared to MNO B. Conventional case 1 provided information about the services that were not migrated, with the vehicle serving the remote cloud, as seen in the existing models. Conventional case 2 did not account for the available or required resources for initiating service migration and performed better compared to the existing models. The SDN controller responds to requests by allocating the installed bandwidth with the new rule for idle computational resources. Figure 3 compares the bandwidth performance of the existing and MEC-SDN models.

The SDN controller responds to changes and installs the flow for the relocated application session, limiting the allocated bandwidth to a range of 5 Mbps to maintain service continuity. Latency is induced during migration as a service by the proposed DRS, which showcases lower results compared to cloud-based service migration. Figure 4 provides a comparison of the bandwidth with respect to the existing methods.

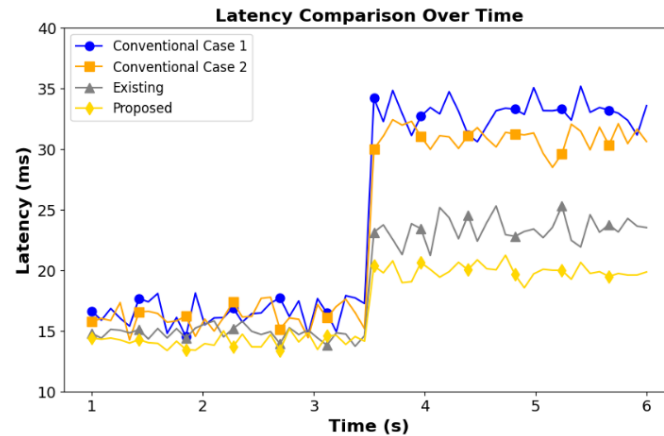


Figure 2. Evaluation of Latency compared with the existing and the MEC-SDN model

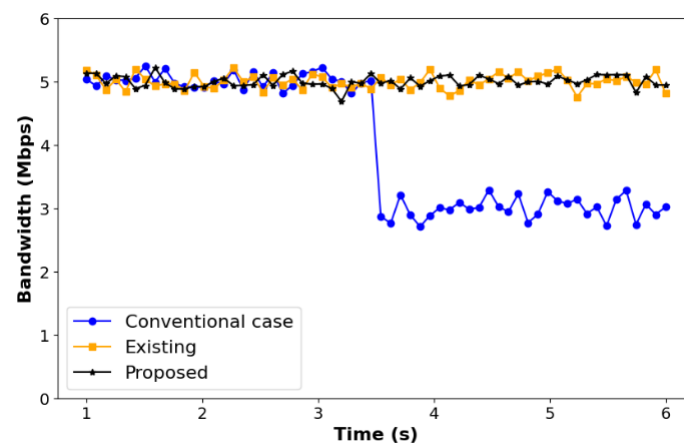


Figure 3. Comparison of Bandwidth with the existing and the MEC-SDN model

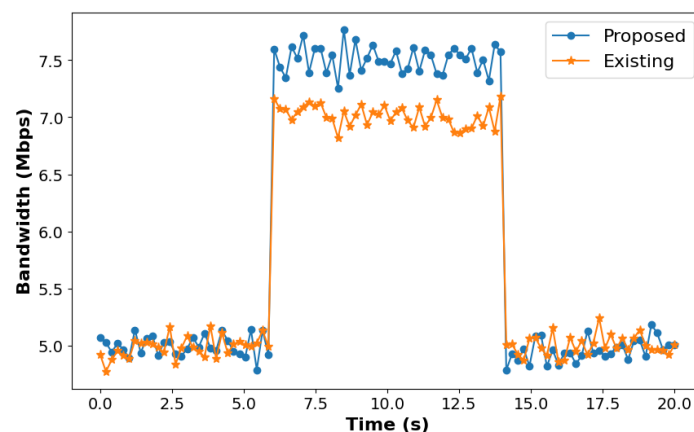


Figure 4. Comparison of the Bandwidth with respect to the existing methods

4.2. Comparative analysis

The SDN controller reacts to changes by installing the flow, which relocates the ME application as a session and limits the bandwidth to 5 Mbps to maintain the service. The migration of the sample container consists of images with sizes of 123 MB and 296 MB, which are induced with the service. The proposed DRS method demonstrates migration performance that shows a lower rate compared to cloud-based migration. The existing copy and transfer techniques, supported by Linux and Docker, are also considered. Table 1 presents the comparative analysis in terms of bandwidth, latency, and SDN-BWMS.

SDN and cloud-native virtualization have enabled the maintenance of continuous service during relocations. The existing approach achieves a latency of 6-16 ms, a bandwidth of 4.8-5 Mbits/sec, and SDN-BWMS of 8 Mbits/sec. The existing hybrid SDN provides mobile network operators with SDN when end-to-end users fail to operate, resulting in a latency range of 13.6 ms to 30.1 ms. In contrast, the MEC-SDN model achieves latency between 6-14 ms, a bandwidth of 5.2 Mbits/sec, and SDN-BWMS of 5.4 Mbits/sec.

Table 1. Comparative analysis

Method	Latency	Bandwidth	SDN-BWMS
Shah <i>et al.</i> [16]	6-16 ms	4.8-5Mbits/sec	8Mbits/sec
Silva <i>et al.</i> [17]	13.6 to 30.1ms	-	-
Proposed	6-14 ms	5.2-5.1 Mbits/sec	5.4 Mbits/sec – 4.8 Mbits/sec

4.3. Discussion

From the overall analysis, this research establishes the efficiency of the MEC-SDN model in minimizing latency and enhancing bandwidth. The result output shows that the proposed MEC-SDN achieves a minimum latency of 6-14 ms and a bandwidth of 5.2 Mbits/sec, which is better than the existing methods. Moreover, this research builds upon previous methods on MEC and SDN, which have independently demonstrated their capacity to enhance overall performance. By integrating MEC-SDN, this research produces a comprehensive output for effective and scalable management. The results of this research align with existing methods, such as those by Shah *et al.* [16] and Silva *et al.* [17], who also discovered the advantages of MEC and SDN in network designs. Future research will explore the reliability and scalability of the MEC-SDN model in large-scale networks, as well as its applicability in various scenarios, including smart cities and IoT. Furthermore, combining machine learning and artificial intelligence techniques with MEC-SDN will further improve overall efficiency and performance. In conclusion, this research validates the capability of MEC-SDN to develop network architectures that provide low-latency and high-bandwidth. As the need for efficient and scalable network management continues to grow, the findings of this study offer a valuable contribution to the development of future network architectures.

5. CONCLUSION

In recent years, the integration of MEC and SDN has become a vital concept in the rapidly developing technological environment. As the demand for low-latency and high-bandwidth applications continues to increase, it is crucial to develop advanced solutions that effectively handle network resources and ensure smooth connectivity. This research establishes that the proposed MEC-SDN provides a feasible solution to address the challenges related to network congestion. As stated earlier, the integration of MEC-SDN significantly enhances network performance, improves the overall quality of experience, and minimizes latency. Some researchers argue that conventional hybrid SDN solutions are sufficient for addressing current network requirements. However, our research reveals that these solutions experience a substantial increase in latency, making them unsuitable for real-time processing applications. In contrast, the proposed MEC-SDN achieves considerably lower latency and enhanced bandwidth, positioning it as an ideal solution for future network architectures. The proposed MEC-SDN has shown its viability and has been tested in various mobility scenarios. The existing hybrid SDN introduced SDN to mobile network operators when end-to-end users failed to operate, as the latency increased from 13.6 ms to 30.1 ms. In comparison, the MEC-SDN model achieved 6-14 ms of latency, a bandwidth of 5.2 Mbits/sec, and SDN-BWMS of 5.4 Mbits/sec. The developed approach highlights the challenges of mobility management, particularly when the slicing of the framework is not supported by edge clouds. However, to further investigate and develop the model, a group of mobile users and mobility would be required, along with resource slicing in the future. Moreover, exploring advanced mobility management approaches will help improve service continuity and reduce handover latency.

FUNDING INFORMATION

Authors state no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Sri Ramachandra Lakkaiah	✓	✓	✓	✓	✓	✓		✓	✓	✓			✓	✓
Hareesh Kumbhinarasaiah		✓				✓		✓	✓	✓	✓	✓		

C : **C**onceptualization

M : **M**ethodology

So : **S**oftware

Va : **V**alidation

Fo : **F**ormal analysis

I : **I**nterpretation

R : **R**esources

D : **D**ata Curation

O : **O**riginal Draft

E : **E**diting

Vi : **V**isualization

Su : **S**upervision

P : **P**roject administration

Fu : **F**unding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

Data availability is not applicable to this paper as no new data were created or analyzed in this study.





REFERENCES

- [1] G. Wu, H. Wang, H. Zhang, Y. Zhao, S. Yu, and S. Shen, "Computation offloading method using stochastic games for software-defined-network-based multiagent mobile edge computing," *IEEE Internet of Things Journal*, vol. 10, no. 20, pp. 17620–17634, Oct. 2023, doi: 10.1109/JIOT.2023.3277541.
- [2] Q. Zhang, C. Li, Y. Huang, and Y. Luo, "Effective multi-controller management and adaptive service deployment strategy in multi-access edge computing environment," *Ad Hoc Networks*, vol. 138, p. 103020, Jan. 2023, doi: 10.1016/j.adhoc.2022.103020.
- [3] R. Singh, R. Sukapuram, and S. Chakraborty, "A survey of mobility-aware Multi-access edge computing: Challenges, use cases and future directions," *Ad Hoc Networks*, vol. 140, p. 103044, Mar. 2023, doi: 10.1016/j.adhoc.2022.103044.
- [4] C. Fan, J. Cui, H. Zhong, I. Bolodurina, and D. He, "MM-SDVN: efficient mobility management scheme for optimal network handover in software-defined vehicular network," *IEEE Internet of Things Journal*, vol. 11, no. 19, pp. 32089–32104, Oct. 2024, doi: 10.1109/JIOT.2024.3422659.
- [5] S. R. Alkaabi, M. A. Gregory, and S. Li, "Multi-access edge computing handover strategies, management, and challenges: a review," *IEEE Access*, vol. 12, pp. 4660–4673, 2024, doi: 10.1109/ACCESS.2024.3349587.
- [6] R. Chaganti, W. Suliman, V. Ravi, and A. Dua, "Deep learning approach for SDN-enabled intrusion detection system in IoT networks," *Information (Switzerland)*, vol. 14, no. 1, p. 41, Jan. 2023, doi: 10.3390/info14010041.
- [7] J. Bhayo, S. A. Shah, S. Hameed, A. Ahmed, J. Nasir, and D. Draheim, "Towards a machine learning-based framework for DDOS attack detection in software-defined IoT (SD-IoT) networks," *Engineering Applications of Artificial Intelligence*, vol. 123, p. 106432, Aug. 2023, doi: 10.1016/j.engappai.2023.106432.
- [8] R. A. Elsayed, R. A. Hamada, M. I. Abdalla, and S. A. Elsaid, "Securing IoT and SDN systems using deep-learning based automatic intrusion detection," *Ain Shams Engineering Journal*, vol. 14, no. 10, p. 102211, Oct. 2023, doi: 10.1016/j.asej.2023.102211.
- [9] N. Sreekanth *et al.*, "Evaluation of estimation in software development using deep learning-modified neural network," *Applied Nanoscience (Switzerland)*, vol. 13, no. 3, pp. 2405–2417, Feb. 2023, doi: 10.1007/s13204-021-02204-9.
- [10] A. K. Bandani, S. Riyazuddin, P. Bidare Divakarachari, S. N. Patil, and G. Arvind Kumar, "Multiplicative long short-term memory-based software-defined networking for handover management in 5G network," *Signal, Image and Video Processing*, vol. 17, no. 6, pp. 2933–2941, Apr. 2023, doi: 10.1007/s11760-023-02514-1.
- [11] M. A. Al-Shareeda, A. A. Alsadhan, H. H. Qasim, and S. Manickam, "Software defined networking for internet of things: review, techniques, challenges, and future directions," *Bulletin of Electrical Engineering and Informatics*, vol. 13, no. 1, pp. 638–647, Feb. 2024, doi: 10.11591/eei.v13i1.6386.
- [12] A. Santos, J. Bernardino, and N. Correia, "Automated application deployment on multi-access edge computing: a survey," *IEEE Access*, vol. 11, pp. 89393–89408, 2023, doi: 10.1109/ACCESS.2023.3307023.
- [13] X. Pei, P. Sun, Y. Hu, D. Li, B. Chen, and L. Tian, "Enabling efficient routing for traffic engineering in SDN with Deep Reinforcement Learning," *Computer Networks*, vol. 241, p. 110220, Mar. 2024, doi: 10.1016/j.comnet.2024.110220.
- [14] C. Yang, F. Liao, S. Lan, L. Wang, W. Shen, and G. Q. Huang, "Flexible resource scheduling for software-defined cloud manufacturing with edge computing," *Engineering*, vol. 22, pp. 60–70, Mar. 2023, doi: 10.1016/j.eng.2021.08.022.
- [15] P. K. Udayaprasad *et al.*, "Energy efficient optimized routing technique with distributed SDN-AI to large scale I-IoT networks," *IEEE Access*, vol. 12, pp. 2742–2759, 2024, doi: 10.1109/ACCESS.2023.3346679.





- [16] S. D. A. Shah, M. A. Gregory, S. Li, and R. D. R. Fontes, "SDN enhanced multi-access edge computing (MEC) for E2E mobility and QoS management," *IEEE Access*, vol. 8, pp. 77459–77469, 2020, doi: 10.1109/ACCESS.2020.2990292.
- [17] R. Silva, D. Santos, F. Meneses, D. Corujo, and R. L. Aguiar, "A hybrid SDN solution for mobile networks," *Computer Networks*, vol. 190, p. 107958, May 2021, doi: 10.1016/j.comnet.2021.107958.
- [18] S. Miriyala and M. S. Sairam, "Improving privacy in SDN based MANET using hybrid encryption and decryption algorithm," *Microprocessors and Microsystems*, p. 103501, Nov. 2020, doi: 10.1016/j.micpro.2020.103501.
- [19] P. P. Devi and B. Jaison, "Protection on wireless sensor network from clone attack using the SDN-enabled hybrid clone node detection mechanisms," *Computer Communications*, vol. 152, pp. 316–322, Feb. 2020, doi: 10.1016/j.comcom.2020.01.064.
- [20] S. Khan and A. Akhunzada, "A hybrid DL-driven intelligent SDN-enabled malware detection framework for internet of medical things (IoMT)," *Computer Communications*, vol. 170, pp. 209–216, Mar. 2021, doi: 10.1016/j.comcom.2021.01.013.
- [21] W. Min *et al.*, "An SDN-orchestrated artificial intelligence-empowered framework to combat intrusions in the next generation cyber-physical systems," *Human-centric Computing and Information Sciences*, vol. 14, no. 11, 2024.
- [22] Y. Rbah *et al.*, "Hybrid software defined network-based deep learning framework for enhancing internet of medical things cybersecurity," *IAES International Journal of Artificial Intelligence*, vol. 13, no. 3, pp. 3599–3610, Sep. 2024, doi: 10.11591/ijai.v13.i3.pp3599-3610.
- [23] A. Motwakel *et al.*, "Enhanced crow search with deep learning-based cyberattack detection in SDN-IoT environment," *Intelligent Automation and Soft Computing*, vol. 36, no. 3, pp. 3157–3173, 2023, doi: 10.32604/iasc.2023.034908.
- [24] M. Y. Akhlaqi and Z. B. Mohd Hanapi, "Task offloading paradigm in mobile edge computing-current issues, adopted approaches, and future directions," *Journal of Network and Computer Applications*, vol. 212, p. 103568, Mar. 2023, doi: 10.1016/j.jnca.2022.103568.
- [25] P. V. Wadatar, R. G. Garroppo, G. Nencioni, and M. Volpi, "Joint multi-objective MEH selection and traffic path computation in 5G-MEC systems," *Computer Networks*, vol. 240, p. 110168, Feb. 2024, doi: 10.1016/j.comnet.2023.110168.
- [26] I. Al Salti and N. Zhang, "An effective, efficient and scalable link discovery (EESLD) framework for hybrid multi-controller SDN networks," *IEEE Access*, vol. 11, pp. 140660–140686, 2023, doi: 10.1109/ACCESS.2023.3339381.
- [27] D. Soldani *et al.*, "E-BPF: a new approach to cloud-native observability, networking and security for current (5G) and future mobile networks (6G and Beyond)," *IEEE Access*, vol. 11, pp. 57174–57202, 2023, doi: 10.1109/ACCESS.2023.3281480.
- [28] Z. E. Ahmed, A. A. Hashim, R. A. Saeed, and M. M. Saeed, "Mobility management enhancement in smart cities using software defined networks," *Scientific African*, vol. 22, p. e01932, Nov. 2023, doi: 10.1016/j.sciaf.2023.e01932.

BIOGRAPHIES OF AUTHORS



Sri Ramachandra Lakkaiah     is an assistant professor in the Department of Computer Science and Engineering at Government SKSJTI, Bangalore. His research interests are in the areas of advanced networking, web technology, big data analytics. He published the papers in reputed journals and presented the papers in international conference. He can be contacted at email: dsram8388@gmail.com.



Hareesh Kumbhinarasaiah     is an associate professor in the Department of Computer Science and Engineering at K. R. Pet Krishna Govt. Engineering College, K R Pet - 571426, Mandya District, Karnataka. His research interests are in the areas of advanced networking, P2P Networks, big data analytics and wireless communications. He published more than 25 papers in reputed journals and presented the papers in international conference. He can be contacted at email: hareeshk.gec@gmail.com.