Lightweight SDN/NFV-based framework for dynamic data-flow and network slice adaptation

Sumbal Zahoor[1], Ali Mamoon[2]
[1]Department of Electrical Engineering, Wigan and Leigh College and University Centre, Greater Manchester, United Kingdom
[2]Engineering Department, Calrom Ltd., United Kingdom

Article Info

Article history:

Received Jan 14, 2026
Revised Feb 6, 2026
Accepted Mar 29, 2026

Keywords:

Dynamic routing
Lightweight virtualization
Network function virtualization
Network slicing
Quality of service management
Software-defined networking

ABSTRACT

The increasing demand for responsive and reliable network services in next-generation communication systems has intensified the need for dynamic resource management and quality of service (QoS) assurance. Software-defined networking (SDN) and network function virtualization (NFV) provide programmability and flexibility for modern networks. However, practical platforms that demonstrate real-time adaptive behavior remain limited. This study differs from prior simulation-focused work by demonstrating real-time adaptive slice control in a reproducible container-based SDN/NFV emulation environment. A bottleneck-aware slice controller is developed to classify degradations as network-limited, server-limited, or service failure using joint indicators, and to select rerouting or service migration using stability constraints and a lightweight action-cost model. Experimental results show that throughput is restored to above 90% of nominal capacity. Recovery typically occurs within two to three control iterations. Service continuity is maintained with low control-plane overhead. The work provides a reproducible experimental baseline and a decision mechanism that reduces incorrect reroutes/migrations under ambiguous key performance indicator (KPI) drops.

Corresponding Author:

Sumbal Zahoor
Department of Electrical Engineering, Wigan and Leigh College and University Centre
Greater Manchester, United Kingdom
Email: sumbalzahoor@yahoo.com

1. INTRODUCTION

Modern communication networks are increasingly required to support heterogeneous services with stringent and often conflicting performance requirements, including high throughput, low latency, and high reliability [1]–[3]. A key challenge is that throughput degradation in sliced networks is often ambiguous: identical performance drops may originate from network congestion, server-side processing limits, or complete service failure. Existing slice controllers typically react to such degradation using threshold-driven rules without identifying the underlying bottleneck. This can lead to incorrect rerouting decisions, unnecessary service migrations, and oscillatory control behaviour under transient conditions [4], [5].

Software-defined networking (SDN) and network function virtualization (NFV) have emerged as key architectural enablers for addressing these challenges. SDN decouples the control plane from the data plane, enabling centralized monitoring and programmable control of traffic flows [6], while NFV virtualizes conventional hardware-based network functions and deploys them as software instances on commodity platforms [7]–[10]. The integration of these paradigms facilitates network slicing, allowing multiple logically isolated networks to operate over shared physical infrastructure while meeting application-specific performance requirements [11], [12].

Although SDN/NFV architectures enable programmable slice management, most existing controllers treat throughput as a single reactive trigger and do not explicitly distinguish between network-limited and server-limited bottlenecks. As a result, adaptation strategies are often brittle and may trigger unnecessary control actions, increasing recovery latency and control-plane overhead. Experimental evidence on bottleneck-aware and stability-aware slice adaptation remains limited, particularly in reproducible emulation environments where routing and service placement interact in real time.

Prior research on SDN/NFV can be grouped into three main themes: network slicing architectures, intelligent control mechanisms, and emulation-based evaluation platforms. Studies on slicing architectures demonstrate performance gains across vehicular, internet of things (IoT), and multi-service environments [13]–[16], while intelligent approaches incorporate intent-based and learning-assisted control. Despite these advances, many contributions rely primarily on analytical models or simulation-based evaluations, offering limited insight into real-time system behaviour under practical operating conditions.

Lightweight emulation platforms such as Mininet provide a practical alternative to large-scale hardware testbeds by enabling controlled experimentation with programmable network architectures. Extensions including Containernet and ComNetsEmu further enhance this capability by supporting container-based virtualization and microservice deployment within SDN environments [17]–[19]. These platforms allow researchers to study realistic interactions between routing control, virtualized services, and network dynamics. However, relatively few studies have employed such frameworks in unified adaptive scenarios to investigate dynamic slice switching, server migration, failure recovery, and heterogeneous demand handling within a unified SDN/NFV system. This gap highlights the need for practical, reproducible experimental frameworks that demonstrate how programmable networks respond to real-time performance degradation, server-side variability, and asymmetric client requirements. Experimental evidence remains limited on how SDN controllers can autonomously adapt data flows and service placement while preserving quality of service (QoS) across multiple operational scenarios.

To overcome these limitations, this paper introduces a diagnosis-driven slice controller that identifies whether performance degradation originates from the network, the server, or a complete service failure using combined performance indicators. The controller applies temporal stability constraints to avoid unnecessary oscillations and uses a lightweight, cost-aware decision model to determine whether traffic should be rerouted or services should be migrated. The proposed framework is implemented in a reproducible SDN/NFV emulation environment using Ryu, Mininet, and ComNetsEmu, and is evaluated across four representative operational scenarios. The main contributions are:

- Bottleneck-aware slice adaptation: a mechanism that differentiates network-limited and server-limited degradation to avoid incorrect control actions.
- Stability-aware control: the use of temporal filtering, persistence windows, and cooldown intervals to reduce oscillatory behaviour during transient performance drops.
- Cost-aware decision making: a lightweight objective model that guides the choice between rerouting and service migration.
- Reproducible experimental validation: a container-based SDN/NFV test environment evaluated under four practical scenarios.

In contrast to throughput-only reactive controllers, the proposed approach clearly separates fault diagnosis from corrective action, allowing more reliable adaptation when performance degradation is ambiguous.

2. RESEARCH METHOD

This study employs a lightweight SDN/NFV-enabled emulation framework to investigate dynamic data-flow management under varying network conditions. The framework integrates Mininet for virtual switch and host emulation with ComNetsEmu to support container-based deployment of virtualized services. An SDN controller implemented using the Ryu framework manages forwarding behaviour, while a custom slice controller module is responsible for performance monitoring and adaptive decision-making. This design follows established SDN/NFV architectural principles reported in the literature.

All experiments were conducted within a single Ubuntu 20.04 virtual machine to ensure reproducibility and consistency across experimental runs. The environment includes Mininet version 2.2.2, Docker, ComNetsEmu, Python 3.8, and the Ryu SDN controller. Before each experiment, the emulation environment was fully reinitialised by clearing Mininet forwarding states, restarting the controller, and rebuilding service containers. Table 1 summarises the hardware and software configuration used to deploy the emulation environment.

The emulated network topology consists of multiple interconnected OpenFlow switches and seven hosts. Five hosts (H2, H3, H4, H6, and H7) operate as traffic-generating clients, while two hosts (H1 and H5) function as service-providing servers. All inter-switch links are initially configured at 15 Mbps (observed throughput ≈14.3 Mbps in baseline runs), providing a stable baseline for evaluating performance degradation, adaptive routing, and service migration. Traffic flows are generated using standard transport-layer benchmarking tools to enable accurate measurement of throughput and latency.

The slice controller operates as the central control component within the framework. It periodically collects flow statistics and link-state information from the SDN controller, including observed throughput, reachability status, and latency indicators. These metrics are evaluated against predefined service-level thresholds to detect congestion, link degradation, or server-side performance anomalies. Similar monitoring-based control approaches have been shown to be effective for QoS-aware management in programmable networks [20], [21].

Routing and server-selection decisions are driven primarily by observed throughput, followed by link availability and recent performance stability. When multiple feasible alternatives exist, the controller selects the option that provides the highest sustained throughput while preserving route continuity. If a server becomes unreachable or exhibits persistent performance degradation, traffic is migrated to an alternative server through updated OpenFlow rules without manual intervention. This decision logic aligns with adaptive resource orchestration strategies reported in prior SDN/NFV studies [22]. The slice controller operates according to a periodic monitoring loop: (i) collect throughput and reachability metrics; (ii) compare metrics with predefined thresholds; (iii) select the highest-performing feasible path or server; and (iv) update forwarding rules if degradation is detected. This loop enables rapid adaptive response while minimizing unnecessary reconfiguration.

Table 1. Emulation environment specification

Component	Specification
Processor	Intel Core i7 (4 cores, 3.60 GHz)
RAM	8 GB
Storage	120 GB SSD
Operating system	Ubuntu 20.04 LTS
Python version	Python 3.8.10
Tools used	Mininet 2.2.2, ComNetsEmu, Docker
Controller	Ryu SDN Controller

The slice controller operates as an iterative monitoring and decision loop implemented in Python. At each monitoring cycle, the controller measures end-to-end throughput, latency indicators, and server reachability for the active slice. To reduce sensitivity to short-term fluctuations, throughput measurements are smoothed using an exponential moving average:

$$\bar{T}(k) = \alpha T(k) + (1-\alpha)\bar{T}(k-1) \quad 0 < \alpha < 1$$

where $T(k)$ is the measured throughput at cycle k and $\bar{T}(k)$ is the filtered throughput.

A slice is considered degraded when the filtered throughput falls below a predefined service threshold for several consecutive cycles. The service threshold is defined relative to the nominal link capacity to represent QoS violation. When degradation is detected, the controller classifies the bottleneck into one of three states: network-limited, server-limited, or service failure. A service failure is identified when the active server becomes unreachable. If the server remains reachable but an alternative route provides higher throughput, the degradation is classified as network limited. If routing alternatives do not improve performance but another server offers higher sustained throughput, the degradation is classified as server limited.

Based on this diagnosis, the controller selects one of three actions: maintain the current configuration, reroute traffic, or migrate the service to an alternative server. Action selection follows a lightweight rule that prioritizes QoS recovery while minimizing unnecessary reconfiguration. To prevent oscillatory behaviour, adaptation is triggered only when degradation persists over a short persistence window, and a cooldown interval is enforced after each action. This separation between bottleneck diagnosis and action selection enables stable and correct adaptation under ambiguous throughput drops. The evaluated scenarios are consistent with experimental conditions commonly investigated in SDN/NFV emulation and adaptation studies [23]-[25].

3. RESULTS AND DISCUSSION

This section presents the experimental results obtained from the proposed SDN/NFV-based framework under four operational scenarios. Figure 1 illustrates the throughput dynamics observed during bandwidth degradation, server underperformance, emergency recovery, and heterogeneous demand. To complement the visual trends, Table 2 summarises the key quantitative results, including degraded throughput levels, post-adaptation performance, and recovery time.

Across all scenarios, the controller successfully restored throughput to above 90% of the nominal link capacity within a small number of control iterations. In failure scenarios, complete service loss was detected and recovered autonomously, while under heterogeneous demand the assigned bandwidth ratios were preserved, demonstrating effective slice isolation and resource fairness.

Table 2. Summary of quantitative performance across evaluated scenarios

Scenario	Metric	Degraded state	After adaptation	Recovery time
Bandwidth degradation	Throughput (Mbps)	~7.5	>14.0	≤ 3 iterations
Server underperformance	Throughput (Mbps)	~7–8	>14.0	2–3 iterations
Emergency server failure	Throughput (Mbps)	0	>14.0	1–2 iterations
Heterogeneous demand	Throughput (Mbps)	4.8 / 9.6	Maintained	Stable

3.1. Bandwidth degradation

The first scenario examines the impact of reduced link capacity on end-to-end throughput. As shown in Figure 1(a), throughput initially operates close to the configured link capacity of 15 Mbps under normal conditions. When bandwidth degradation is introduced, the throughput decreases by approximately 40–50%, indicating that the active path is no longer able to support the required service rate. Following detection of the degraded condition, the SDN controller selects an alternative path with higher available capacity. This adaptive rerouting results in recovery of throughput to above 90% of the baseline level within two control iterations, demonstrating effective detection of bandwidth degradation and timely path reconfiguration.

3.2. Server underperformance

The second scenario focuses on server-side performance limitations. In this case, network conditions remain unchanged while the active server exhibits reduced processing capability. As shown in Figure 1(b), throughput remains consistently low during the initial iterations despite adequate network resources. After service migration is initiated by the controller, throughput increases by more than a factor of two compared to the degraded state and stabilizes near baseline levels within 2–3 monitoring cycles, highlighting the importance of server-aware adaptation in maintaining end-to-end service quality.

3.3. Emergency recovery

The third scenario evaluates system behaviour under complete service disruption. As illustrated in Figure 1(c), throughput drops to 0 Mbps when the active server becomes unreachable, indicating a full loss of service. The controller detects the server unavailability and initiates recovery by redirecting traffic to the standby server. Throughput is restored to near-baseline levels within one to two monitoring intervals, demonstrating effective failure detection and autonomous service restoration. Repeated failure events exhibit a consistent loss-and-recovery pattern, confirming stable recovery behaviour under emergency conditions.

3.4. Heterogeneous demand

The final scenario considers multiple clients with different bandwidth requirements sharing the same network infrastructure. As shown in Figure 1(d), the clients maintain stable and differentiated throughput levels in accordance with their assigned slices. The observed throughput ratio between low-demand and high-demand clients remains consistent, with deviations below 10%, indicating effective slice isolation, fair resource allocation, and the absence of starvation.

3.5. Discussion

The results demonstrate that distinguishing the origin of performance degradation improves adaptation correctness. In bandwidth degradation scenarios, the controller classifies the bottleneck as network-limited and prioritizes rerouting, whereas server underperformance triggers service migration. This discrimination prevents unnecessary control actions that would occur in throughput-only reactive policies. The incorporation of temporal stability constraints reduces oscillatory behavior during transient fluctuations and maintains consistent recovery performance across repeated events. A throughput-only threshold controller would treat both bandwidth degradation and server underperformance as identical key performance indicator (KPI) drops, often rerouting first even when the bottleneck is server-side; the proposed diagnosis-first mechanism avoids this mis-action and reduces unnecessary reconfiguration.

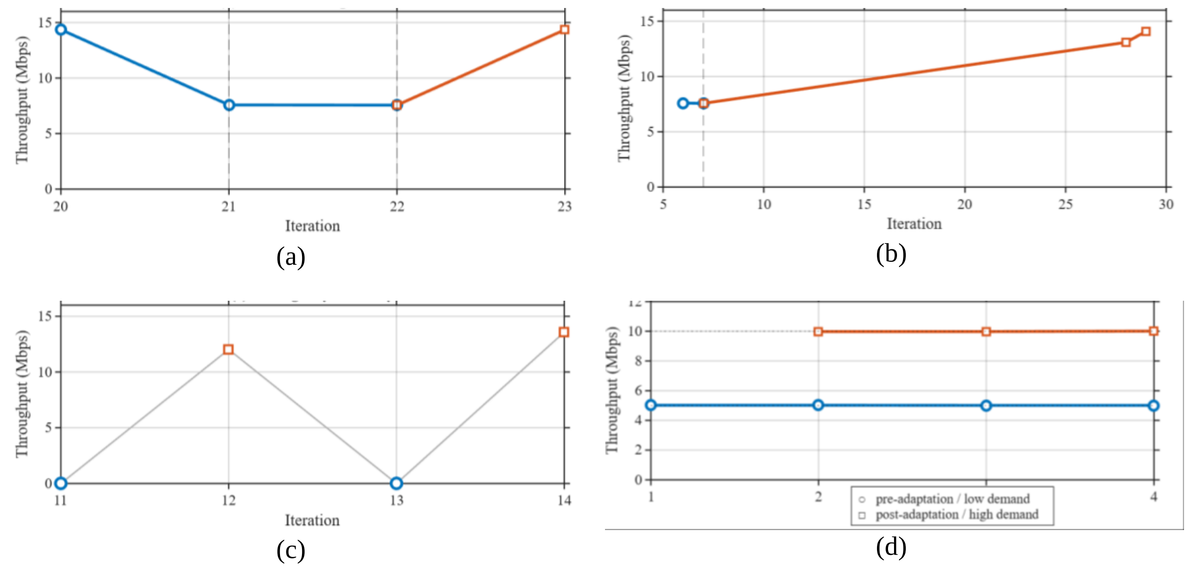

Figure 1. Throughput behavior under four operational scenarios: (a) bandwidth degradation, (b) server underperformance, (c) emergency recovery, and (d) heterogeneous demand

4. CONCLUSION

This paper introduced a bottleneck-aware and stability-aware slice control model for SDN/NFV networks. The key contribution is a lightweight decision mechanism that distinguishes network and server bottlenecks and selects adaptation actions using cost and stability constraints. The aim was to evaluate whether real-time monitoring and centralized SDN control can effectively maintain QoS under changing network and server conditions. The experimental results show that the proposed framework adapts reliably across four representative operational scenarios: bandwidth degradation, server underperformance, emergency server failure, and heterogeneous client demand. In each case, the controller demonstrated robust adaptive behavior and maintained service continuity and adjusted routing or service placement without manual intervention. These findings confirm that bottleneck-aware diagnosis combined with stability constraints enables correct and low-overhead adaptation under ambiguous KPI drops.

The use of a lightweight emulation environment based on Mininet and ComNetsEmu allows the framework to be evaluated in a reproducible and practical manner, without relying on complex or costly testbed infrastructures. This makes the proposed approach suitable for early-stage evaluation of SDN/NFV-based adaptive networking solutions. Future work will focus on scaling the framework to larger topologies, incorporating additional performance metrics such as latency and packet loss, and exploring learning-based control strategies to further enhance adaptability in real-world deployments. Across all evaluated scenarios, the proposed framework consistently restores service to near-baseline throughput levels within a small number of control iterations, confirming its suitability for lightweight, real-time SDN/NFV-based adaptive resource management.

FUNDING INFORMATION

This research received no external funding.

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
Sumbal Zahoor	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓		✓	
Ali Mamoon			✓			✓		✓		✓	✓	✓		

C	: **C**onceptualization	I	: **I**nvestigation	Vi	: **Vi**sualization
M	: **M**ethodology	R	: **R**esources	Su	: **Su**pervision
So	: **So**ftware	D	: **D**ata Curation	P	: **P**roject administration
Va	: **Va**lidation	O	: Writing - **O**riginal Draft	Fu	: **Fu**nding acquisition
Fo	: **Fo**rmal analysis	E	: Writing - Review & **E**diting		

CONFLICT OF INTEREST STATEMENT

The authors declare that there are no financial, commercial, or institutional relationships that could be construed as a potential conflict of interest in relation to this research. The authors further confirm that the conduct, analysis, and reporting of this study were carried out independently and objectively.

DATA AVAILABILITY

The data supporting the findings of this study are available from the corresponding author upon reasonable request. The experimental results were obtained using a lightweight SDN/NFV emulation environment, and the associated topology configurations, controller scripts, and container-based service definitions can be shared to support reproducibility.

REFERENCES

[1] M. Patzold, "Toward realizing the full potential of a 5G-empowered world [mobile radio]," *IEEE Vehicular Technology Magazine*, vol. 15, no. 1, pp. 5–11, 2020, doi: 10.1109/MVT.2019.2956891.

[2] F. Bahramisirat, M. A. Gregory, and S. Li, "Multi-access edge computing resource slice allocation: a review," *IEEE Access*, vol. 12, pp. 188572–188589, 2024, doi: 10.1109/ACCESS.2024.3515077.

[3] W. Rafique, L. Qi, I. Yaqoob, M. Imran, R. U. Rasool, and W. Dou, "Complementing IoT services through software defined networking and edge computing: a comprehensive survey," *IEEE Communications Surveys and Tutorials*, vol. 22, no. 3, pp. 1761–1804, 2020, doi: 10.1109/COMST.2020.2997475.

[4] S. Khan, F. K. Hussain, and O. K. Hussain, "Guaranteeing end-to-end QoS provisioning in SOA based SDN architecture: A survey and open issues," *Future Generation Computer Systems*, vol. 119, pp. 176–187, 2021, doi: 10.1016/j.future.2021.02.011.

[5] R. Krishnan and S. Durairaj, "Reliability and performance of resource efficiency in dynamic optimization scheduling using multi-agent microservice cloud-fog on IoT applications," *Computing*, vol. 106, no. 12, pp. 3837–3878, 2024, doi: 10.1007/s00607-024-01301-1.

[6] S. Zahoor *et al.*, "Comprehensive analysis of network slicing for the developing commercial needs and networking challenges," *Sensors*, vol. 22, no. 17, p. 6623, 2022, doi: 10.3390/s22176623.

[7] S. Ammar, C. Pong Lau, and B. Shihada, "An in-depth survey on virtualization technologies in 6G integrated terrestrial and non-terrestrial networks," *IEEE Open Journal of the Communications Society*, vol. 5, pp. 3690–3734, 2024, doi: 10.1109/OJCOMS.2024.3414622.

[8] K. Kaur, V. Mangat, and K. Kumar, "A review on virtualized infrastructure managers with management and orchestration features in NFV architecture," *Computer Networks*, vol. 217, p. 109281, 2022, doi: 10.1016/j.comnet.2022.109281.

[9] P. Alemany *et al.*, "A KPI-enabled NFV MANO architecture for network slicing with QoS," *IEEE Communications Magazine*, vol. 59, no. 7, pp. 44–50, 2021, doi: 10.1109/MCOM.001.2001077.

[10] H. O. Otieno, B. Malila, and J. Mwangama, "Deployment and management of intelligent end-to-end network slicing in 5G and beyond 5G networks: a systematic review," *IEEE Access*, vol. 12, pp. 190411–190433, 2024, doi: 10.1109/ACCESS.2024.3516051.

[11] S. Zahoor, I. Ahmad, A. U. Rehman, E. T. Eldin, N. A. Ghamry, and M. Shafiq, "Performance evaluation of virtualization methodologies to facilitate NFV deployment," *Computers, Materials and Continua*, vol. 75, no. 1, pp. 311–329, 2023, doi: 10.32604/cmc.2023.035960.

[12] X. S. Shen *et al.*, "Data management for future wireless networks: architecture, privacy preservation, and regulation," *IEEE Network*, vol. 35, no. 1, pp. 8–15, 2021, doi: 10.1109/MNET.011.2000666.

[13] T. Do Vale Saraiva, C. A. V. Campos, R. D. R. Fontes, C. E. Rothenberg, S. Sorour, and S. Valaee, "An application-driven framework for intelligent transportation systems using 5G network slicing," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 8, pp. 5247–5260, 2021, doi: 10.1109/TITS.2021.3086064.

[14] A. Ahmad, A. Floris, and L. Atzori, "Timber: an SDN-based emulation platform for experimental research on video streaming," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 7, pp. 1374–1387, 2020, doi: 10.1109/JSAC.2020.2999683.

[15] M. Zhao, Y. Zhang, Q. Liu, A. Kak, and N. Choi, "AdaSlicing: adaptive online network slicing under continual network dynamics in open radio access networks," *Proceedings - IEEE INFOCOM*. IEEE, pp. 1–10, 2025, doi: 10.1109/INFOCOM55648.2025.11044475.

[16] K. Abbas, T. A. Khan, M. Afaq, and W. C. Song, "Network slice lifecycle management for 5G mobile networks: an intent-based networking approach," *IEEE Access*, vol. 9, pp. 80128–80146, 2021, doi: 10.1109/ACCESS.2021.3084834.

[17] Z. Xiang, S. Pandi, J. Cabrera, F. Granelli, P. Seeling, and F. H. P. Fitzek, "An open source testbed for virtualized communication networks," *IEEE Communications Magazine*, vol. 59, no. 2, pp. 77–83, 2021, doi: 10.1109/MCOM.001.2000578.

[18] F. H. P. Fitzek, F. Granelli, and P. Seeling, *Computing in communication networks: from theory to practice*. London: Elsevier Academic Press, 2020.

[19] W. T. Mnyandu, A. Terzoli, and H. Kobo, "Emulating LTSP clients with mininet and QEMU: enabling scalable testing for educational deployments," *Proceedings of 2024 4th International Multidisciplinary Information Technology and Engineering Conference, IMITEC 2024*. IEEE, pp. 30–37, 2024, doi: 10.1109/IMITEC60221.2024.10851125.

[20] M. Rostami and S. Goli-Bidgoli, "An overview of QoS-aware load balancing techniques in SDN-based IoT networks," *Journal of Cloud Computing*, vol. 13, no. 1, 2024, doi: 10.1186/s13677-024-00651-7.

[21] E. Zeydan and Y. Turk, "Recent advances in intent-based networking: a survey," *IEEE Vehicular Technology Conference*, vol. 2020-May. IEEE, pp. 1–5, 2020, doi: 10.1109/VTC2020-Spring48590.2020.9128422.

[22] M. Afaq, J. Iqbal, T. Ahmed, I. Ul Islam, and M. Khan, "Towards 5G network slicing for vehicular ad-hoc networks: An end-to-end approach," *Computer Communications*, vol. 149, pp. 252–258, 2020, doi: 10.1016/j.comcom.2019.10.018.

[23] A. Ahmad, A. Floris, and L. Atzori, "Timber: An SDN-Based Emulation Platform for Experimental Research on Video Streaming," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 7, pp. 1374–1387, Jul. 2020, doi: 10.1109/jsac.2020.2999683.

[24] A. K. Alnaim, "Securing 5G virtual networks: a critical analysis of SDN, NFV, and network slicing security," *International Journal of Information Security*, vol. 23, no. 6, pp. 3569–3589, 2024, doi: 10.1007/s10207-024-00900-5.

[25] V. P. Singh, M. P. Singh, S. Hegde, and M. Gupta, "Security in 5G network slices: concerns and opportunities," *IEEE Access*, vol. 12, pp. 52727–52743, 2024, doi: 10.1109/ACCESS.2024.3386632.

BIOGRAPHIES OF AUTHORS

Sumbal Zahoor received the Ph.D. degree in Electrical Engineering and is currently a lecturer in Electrical and Electronic Engineering at Wigan and Leigh College and University Centre, United Kingdom. Her teaching and research experience spans electronic systems, control engineering, signal processing, and programmable networking. Her research interests include software-defined networking (SDN), network function virtualization (NFV), network slicing, lightweight emulation platforms, and performance evaluation of virtualized and distributed systems. She has expertise in modelling, simulation, and experimental validation using tools such as MATLAB/Simulink, Mininet, and container-based virtualization technologies. In this study, she led conceptualization and methodology, implemented the experimental framework, conducted the analysis and validation, and prepared the original manuscript. She can be contacted at email: sumbalzahoor@yahoo.com.

Ali Mamoon is an Engineering Lead at Calrom, United Kingdom, with over a decade of professional experience in software engineering, cloud-based application development, and distributed systems. He holds a Master's degree in Computer Science from the National University of Computer and Emerging Sciences, Pakistan. His technical expertise includes cloud application development, DevOps practices, virtualization technologies, and performance monitoring using modern observability stacks. His professional background spans large-scale, mission-critical systems within the aviation and travel industry, where he has led development teams and contributed to platform architecture and system optimization. In this research, he contributed to software experimentation support, investigation, data preparation, visualization of results, and critical review and editing of the manuscript, particularly in areas related to virtualization and cloud-oriented system implementation. He can be contacted at email: ali@alimamoon.com.