

Mobile Fog Computing by Using SDN/NFV on 5G Edge Nodes

G. R. Sreekanth^{1,*}, S. Ahmed Najat Ahmed², Marko Sarac³, Ivana Strumberger³, Nebojsa Bacanin³
and Miodrag Zivkovic³

¹Department of Computer Science and Engineering, Kongu Engineering College, Erode, 638060, India

²Department of Computer Engineering, Lebanese French University, Erbil, 44001, Iraq

³Singidunum University, Belgrade, 11000, Serbia

*Corresponding Author: G. R. Sreekanth. Email: gsrreekanth1@yahoo.com

Received: 28 May 2021; Accepted: 19 July 2021

Abstract: Fog computing provides quality of service for cloud infrastructure. As the data computation intensifies, edge computing becomes difficult. Therefore, mobile fog computing is used for reducing traffic and the time for data computation in the network. In previous studies, software-defined networking (SDN) and network functions virtualization (NFV) were used separately in edge computing. Current industrial and academic research is tackling to integrate SDN and NFV in different environments to address the challenges in performance, reliability, and scalability. SDN/NFV is still in development. The traditional Internet of things (IoT) data analysis system is only based on a linear and time-variant system that needs an IoT data system with a high-precision model. This paper proposes a combined architecture of SDN and NFV on an edge node server for IoT devices to reduce the computational complexity in cloud-based fog computing. SDN provides a generalization structure of the forwarding plane, which is separated from the control plane. Meanwhile, NFV concentrates on virtualization by combining the forwarding model with virtual network functions (VNFs) as a single or chain of VNFs, which leads to interoperability and consistency. The orchestrator layer in the proposed software-defined NFV is responsible for handling real-time tasks by using an edge node server through the SDN controller via four actions: task creation, modification, operation, and completion. Our proposed architecture is simulated on the EstiNet simulator, and total time delay, reliability, and satisfaction are used as evaluation parameters. The simulation results are compared with the results of existing architectures, such as software-defined unified virtual monitoring function and ASTP, to analyze the performance of the proposed architecture. The analysis results indicate that our proposed architecture achieves better performance in terms of total time delay (1800 s for 200 IoT devices), reliability (90%), and satisfaction (90%).

Keywords: Mobile fog computing; edge computing; edge node; IoT; software-defined networking; network functions virtualization; orchestrator



This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1 Introduction

In recent years, cloud computing has been the main storage platform for many applications. A high demand for storage space has also been noted due to high-level information processing and digital motivation. Edge devices are preferred to help in cloud servers' traffic. However, future edge devices alone cannot process the cloud storage. Therefore, mobile fog computing (MFC) has been introduced by researchers. The quality of service (QoS) of smart devices is guaranteed in fog and cloud computing. They are considered promising technologies for enhancing the quality of devices. In advancement techniques, such as the Internet of things (IoT), considerable data are generated and transferred over the network at a particular time. This condition leads to the importance and need for fog computing on edge nodes for storage and computation of data in the network to reduce the traffic constantly. Hence, the edge devices in the network seem close to users, and the latency for transmitting data from the cloud data center to users is reduced, which leads to rapid computation. MFC is a next assured technology to reduce the latency of transmission, and services in a timely manner will save energy and increase scalability. To make MFC reliable and time efficient, software-defined networking (SDN) and network functions virtualization (NFV) are used.

SDN separates data and control planes in networking switches. The entire network is controlled using centralized software logical programs. This methodology manages networks with new configuration. The main difference between SDN and NFV is as follows: SDN separates the control function from the forwarding function in networks, whereas NFV removes forwarding and other functions from the hardware machine it computes. Both systems mainly rely on the virtualization concept that enables network infrastructure design to separate hardware and software.

SDN and NFV compute together by transferring network data and execute a router control function in a virtual machine. This process is used in cloud computing-based MFC to reduce the computational complexity in the network. The basic function in the network is provided by NFV, while SDN focuses on specific uses. SDN separates hardware in networks, while NFV concentrates on the virtualization of physical hardware during traffic. The traditional IoT data analysis system is only based on a linear and time-variant system that needs an IoT data system with a high-precision model (EstiNet). This limitation leads to the concept of big data management-based SDN to determine the network bandwidth and storage units. This concept will extend research interests to the next level. Instead of connecting IoT devices to the cloud server directly, mobile edge computing (MEC) connects them to the edge server.

1.1 Scope of This Work

- SDN provides a unified model of the forwarding plane that is separated from the control plane. Meanwhile, NFV concentrates on visualization.
- The combination of a forwarding model with virtual network functions (VNFs) leads to interoperability and homogeneity.
- One protocol is adequate for control, management, and orchestration.
- The main advantage of using this concept is that it makes dynamic and flexible network infrastructure.

1.2 Objective of This Work

- The use of SDN/NFV on edge nodes leads to reduced latency and increased scalability.
- The application of SDN/NFV in edge computing results in a dynamic, quick, and reliable computing environment.
- The scaling capabilities of this combined architecture use the network resources in an optimal way.

- Data may be transferred to edge nodes for fast transmission to users, which will increase the execution time of each task.

The rest of this paper is organized as follows. Section 2 presents a detailed literature review. Section 3 explains the proposed architecture and indicates its simulation. Section 4 discusses the simulation result and comparative analysis with existing architectures. Section 5 concludes this work.

2 Related Work

Current industrial and academic studies are tackling to integrate SDN and NFV in different environments to address the challenges in performance, reliability, and scalability. SDN/NFV is still in development. Reference [1] provided a detailed review of an integrated SDN/NFV architecture by synthesizing its architectural design and stated the areas for improvement.

NFV solves many issues by transferring network functions and hardware utilization to commercial off-the-shelf (COTS) software systems in virtual functions [2]. In current networks, the Internet protocol layer combines data and control planes vertically into network devices. On the contrary, SDN separates the control plane from the data plane in the network.

The traditional IoT data analysis system is only based on a linear and time-variant system that needs an IoT data system with a high-precision model [3], which leads to the concept of big data management-based SDN to determine the network bandwidth and storage units. MEC is a recent research topic that combines computing resources in the contiguity of IoT devices [4]. This topic will extend research interests to the next level. Instead of connecting IoT devices to the cloud server directly, MEC connects them to the edge server. Current research on this topic [5] estimates that approximately 29 billion IoT smart devices will be connected to the Internet by 2022.

A distributed deep learning-driven task off-loading (DDTO) method has been proposed to identify the edge and cloud differences in choosing off-loading devices. It generates a near-optimal off-loading decision in mobile, edge, and cloud-centralized servers. The proposed DDTTO in [6] presented high accuracy, which generated decisions on off-loading data effectively and efficiently in edge and cloud servers. Reference [7] discussed NFV and its relationship with SDN and cloud computing. The authors surveyed the clear functions of NFV and identified the promising future research guidelines. They also outlined the key NFV projects and used cases.

The next-generation computing migration outlines for networks were discussed in [8]. The paper discussed MEC with SDN and NFV technologies. It also considered decision making on multiattribute and migrant computing for solving MEC based on SDN and NFV.

Reference [9] proposed Computing Mode Selection (CMS) and priority-based execution sequences for tasks (ASTP) in the architecture of a smart factory based on IIoT and SDN. The SDN controller has the design of the CMS module in it. The mode of optimal computing is obtained by each service while operating in the CMS algorithm. Then, on the basis of task performance in real time and calculated amount, the priorities of tasks are calculated. To complete the transmission of tasks, the SDN controller sends the flow table to the switch in accordance with the task priority values.

Reference [10] discussed the off-loading tasks in MEC. It showed the trade-off between latency and reliability in off-loading. In the proposed architecture, the user equipment divides a task into many subtasks, then they are off-loaded in order of multiple edges. To reduce the latency and off-loading probability, the authors formulated optimization problems, which are nonconvex. They proposed three algorithms based on heuristic approach, linear reformulation, and half defined relaxation. The formulated problems were solved through optimizing edge node candidates' selection, off-loading order, and task allocation.

The architectures of adaptive transmission for SDN and fog computing for IIoT were proposed in [11]. In this work, data streams were divided into two groups, and two strategies were proposed for low and high deadlines. The proposed method performed better than existing methods. The architectures of MEC ecosystem and orchestration were developed in [12]. The orchestrator and external elements functioned in a module to indicate the QoS with the concentration on the channel state of information along with the life cycle of MEC services.

To address network scalability, single process, distributed monitoring of virtual functions, network flexibility, and acceptance of virtual life cycle problems were managed in [13]. The proposed software-defined unified virtual monitoring function called (SuVMF) based on SDN for large networks was used.

Reference [14] stated that SDN and NFV provide a complete structure of network devices. Such a structure needs a common model on abstraction for forwarding model and network functions. Interoperability and homogeneity are ensured in this model.

The utilization of SDN in network access, core network, and WAN between edge nodes and clouds has been reviewed in detail. The issues and future challenges have also been addressed. Reference [15] made significant contributions, ranging from SDN and NFV through ecosystem to orchestration with SDN-NFV, by considering the issues of optimization, network control, and security.

Reference [16] introduced a novel method called autonomic network management and optimization architecture. This work combined SDN and software-defined radio to improve the flexibility, performance, and available resource usage in SDN with a cognitive radio network system. An efficient VNF that works effectively at edges was proposed in [17] for enhancing network services in IoT applications.

To simplify the distribution of service-chained VNFs, the authors proposed an approach that extends the network cloud infrastructure with MEC for permitting delay-sensitive network traffic. To make 5G networks heterogeneous, [18] designed a model using the SDN architecture. In terms of the security issues in the centralization of SDN, the authors proposed an error-free security architecture for 5G networks. Reference [19] exploited the flexibility of using SDN technology. It proposed an architecture called SDN-based MEC for solving the problem of choosing MEC and end user off-load data. The proposed architecture utilizes a reinforcement learning method and game theory techniques.

Reference [20] designed an NFV orchestration that is scalable with clouds. To implement the data plane, segment routing was used. The proposed architecture exhibited minimal complexity in accordance with the number of orchestrations. Reference [21] reviewed SDN technologies and services with tools. It also presented an overview of tactical, ad hoc, military, and coalition networks. Reference [22] presented the challenges related to the adaptation of 5G technologies. The authors proposed a workflow for PaaS in multicloud based on modularity, microservices, and shape order.

Reference [23] presented the uses of combining SDN with MEC on IoT with detailed reviews. References [24,25] focused mainly on IoT research and applications.

3 Proposed 5G SDN/NFV Concept for Fog Computing

3.1 Problem Description

With the advancement in IoT, large volumes of data are produced from IoT devices, such as smart home appliances, tablets, smartphones, routers, smart traffic systems, smart energy meters, cellular base stations, connected vehicles, and smart building controllers. Cloud computing efficiently processes considerable data in networks. With the rapid growth of the kind of IoT devices, excessive amounts of data must be transferred continuously over the network within a particular time. These data must be processed and responded as early as possible. As a result, the edge devices in the network are near to users, and the latency for transmitting data from the cloud data center to users is reduced, which leads to fast

computation. We redevelop different communication methods with fog computing, such as SDN, NFV, content distribution network, the concept of 5G technology, and Long-Reach Passive Optical Network.

3.2 Architectural Overview

An overview of the proposed work is presented in Fig. 1. Cloud services are stored at the top of the layer called tier 3, which consists of the cloud server with large capacity. Tier 2 consists of an edge node, which is connected to all other networks and comprises all network devices, such as routers and gateways. The proposed SDN/NFV architecture is applied on the edge node. Tier 3 consists of IoT devices, such as cameras, smartphones, sensors, and computers.

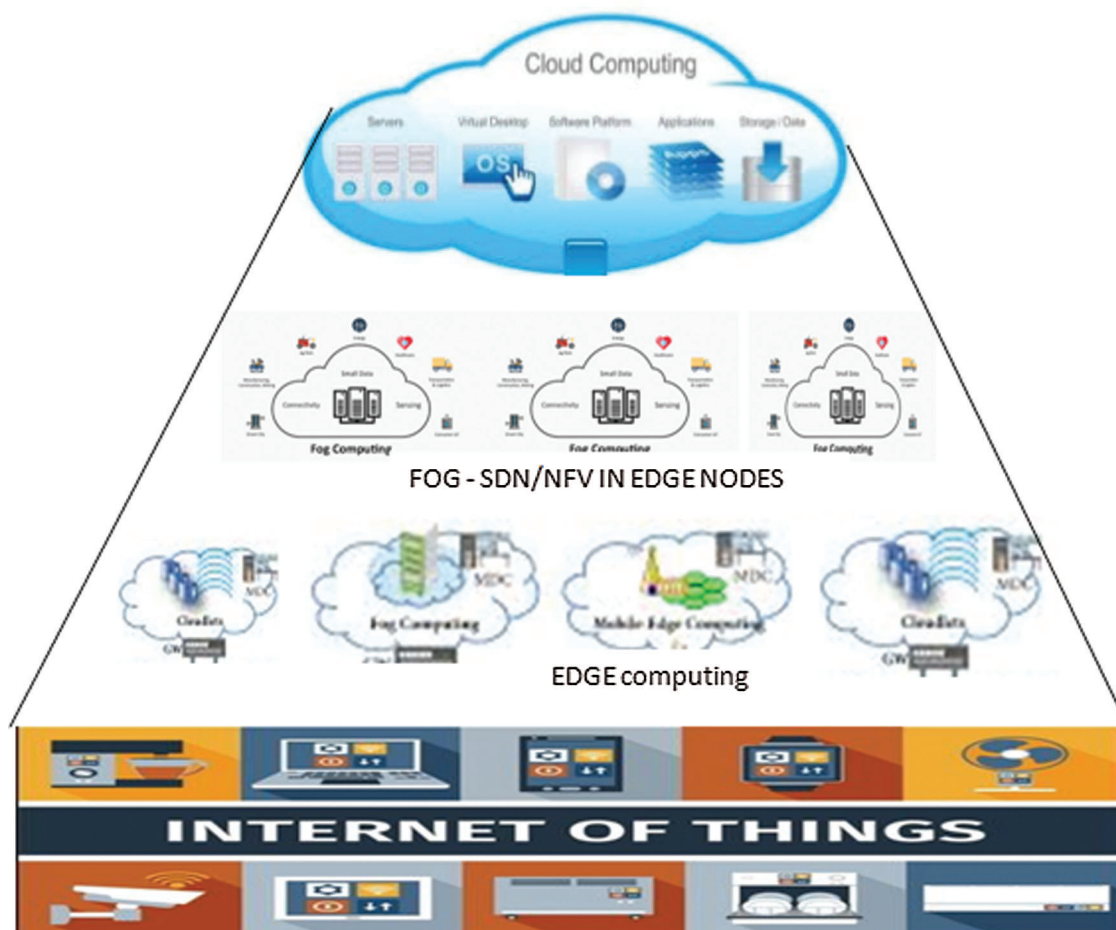


Figure 1: Overview of the proposed SDN/NFV on edge nodes

Fig. 2 shows a detailed overview of our proposed architecture. It consists of three layers: application, control, and infrastructure. The first layer is responsible for network services at the cloud data center. These services communicate to the control plane of SDN's second layer called control layer through the open SDN orchestrator. In this proposed work, an SDN controller called Ryu is used. Ryu is an open-source structure created by NTT and written in Python. It supports OpenFlow, OF-Cong, and NETCONF. SDN's control plane is in charge of the traffic control in the application layer. The edge nodes directly communicate with the end devices. They communicate with the application layer through the SDN

controller. Accordingly, the latency will be reduced, and data processing will be increased. SDN's main feature is the division of the control and forwarding planes.

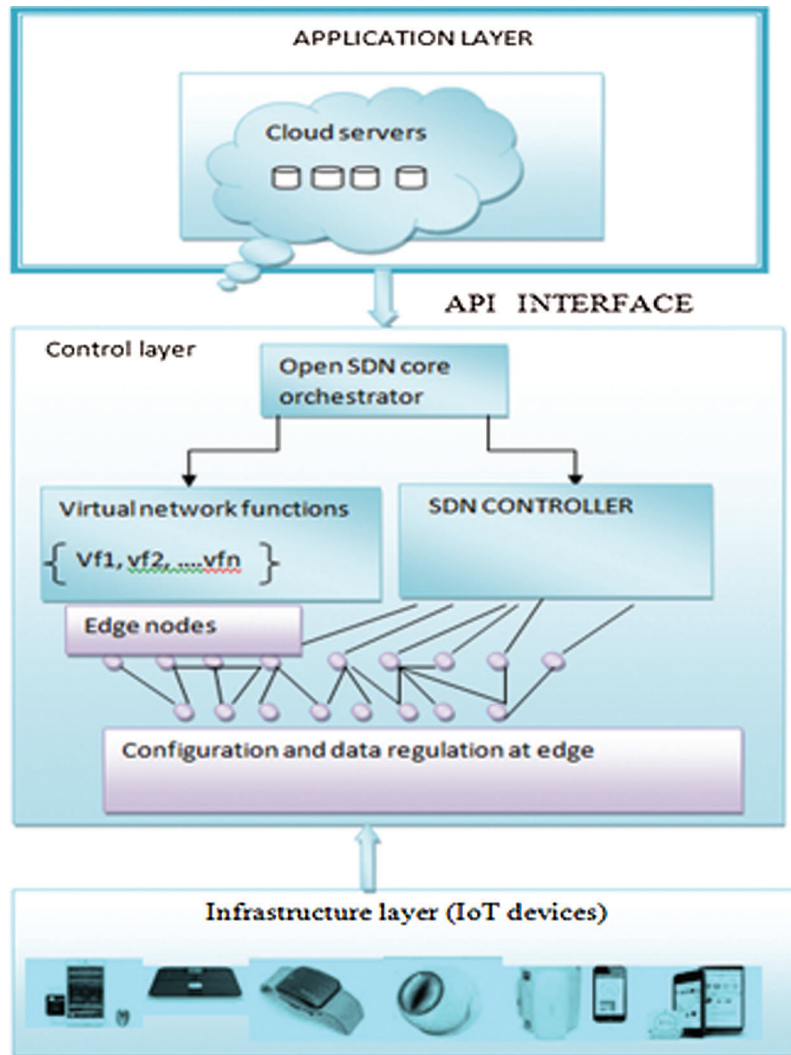


Figure 2: Architecture of the proposed SDNFV on edge nodes

In Fig. 3, in SDN's control plane, software called SDN controller is installed, which is responsible for handling the network traffic. It will run on COTS systems. The data plane is responsible for transferring data based on rules. The SDN controller combined with NFV VNFs at the edge is responsible for managing the rules through API. It does not have direct access to API. Consequently, the scalability and data processing accelerate. This work mainly focuses on traffic control by monitoring the data transferred to IoT devices from clouds and vice versa.

The transferred data must be related to the query created by the end user, which means data processing is improved in terms of time. The task of transferring the correct data to the user is based on three indexes: i) higher, ii) lower, and iii) neutral. If the importance of a real-time task is high and the computational necessity is low, then the task belongs to the higher index. By contrast, when the importance of a real-time task is low

and the computational necessity is high, then the task belongs to the lower index. That is, the smaller the calculation amounts are, the lower the complexity of the task will be.

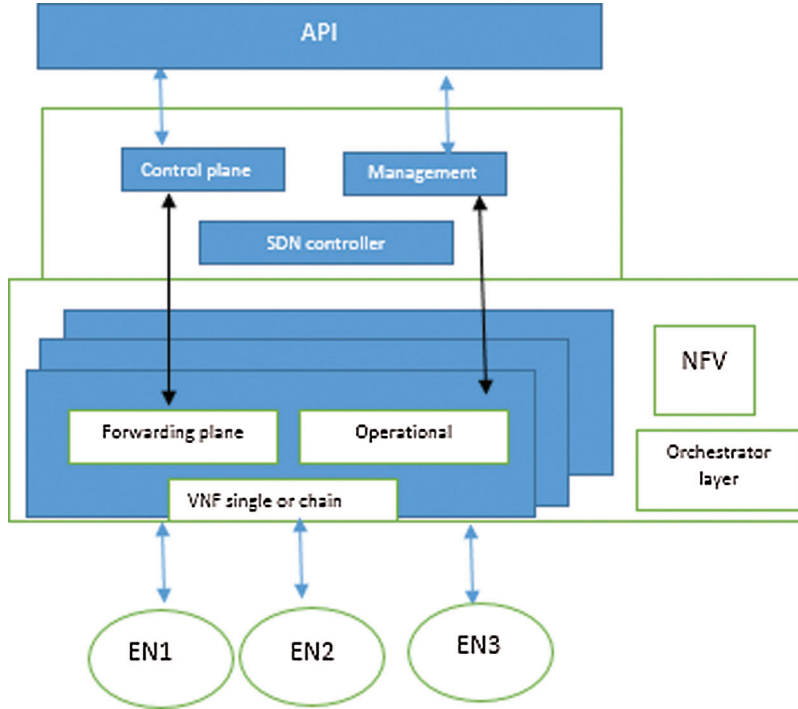


Figure 3: Orchestrator layer in the proposed SDNFV on edge nodes

3.3 Algorithm 1: Proposed Software-Defined NFV (SDNFV) on Edge Nodes

1. Data transmission in the industrial layer consists of IoT devices. A set of $N(d)$ devices called $N(d) = \{1, 2, \dots, N\}$ exists, and each device has a task to be completed in a certain limit.

2. The orchestrator layer in the proposed SDNFV is responsible for handling real-time tasks by using an edge node server through the SDN controller via four actions: task creation, modification, operation, and completion.

If request == valid then

 If task == exists then

 Executed from the catalogue of the network system

 Else

 Call the service-chained module of SDNFV

 Update the task

 end

3. Once the task is created, activated, and completed, the computational complexity is calculated as

$$T_n = \{S_n, C_n, L_{n,max}\} n \in N \quad (1)$$

where

T_n – task for a particular device

C_n – calculation of the CPU cycle amount needed to complete the task

$L_{n,max}$ – maximum latency of device n (in s)

4. After all input D_n is received, the task is performed on the fog node. Thus, on the edge node, the execution time of task J_n consists of two sectors: transmission and computing time. $f_{n,E}$ denotes the computational resource of the edge server.

$$J_{n,E} = \frac{D_n}{r_{n,E}} + \frac{C_n}{f_{n,E}} \quad (2)$$

5. The application layer performs the computation task based on the five indexes indicated in Eqs. (3)–(7) to analyze the cost, time, and computation.

3.4 Workflow of the Proposed Architecture

The orchestrator layer in the proposed SDNFV is responsible for handling real-time tasks by using an edge node server through the SDN controller via four actions: task creation, modification, operation, and task completion. The workflow of the proposed architecture is depicted in Fig. 4.

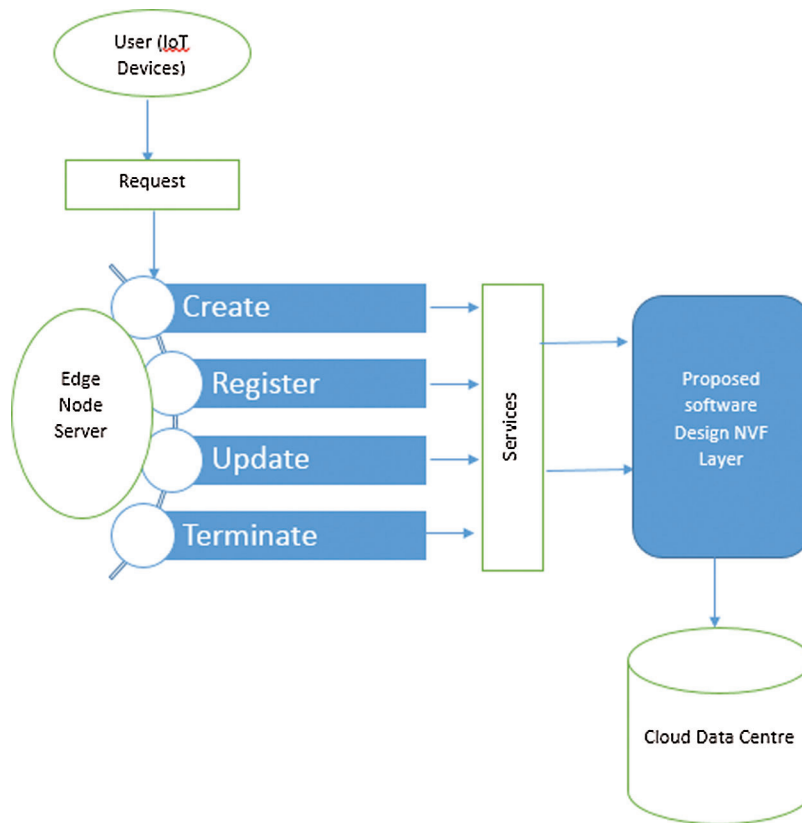


Figure 4: Workflow of the proposed SDNFV on edge nodes

From Fig. 4, the workflow of the proposed architecture is elucidated as follows:

- i) **Task Request:** An IoT device (user) sends the task request to the cloud processor. It is described using predefined templates.

- ii) **Task Checking:** Once the request is received by the orchestrator in the proposed architecture via the edge node, it performs a security check to determine whether the request is authorized. If it is authorized, then it will be processed. Otherwise, the task request will be terminated.
- iii) **Task Registration:** If the service of the task requested is already available, then the required resources will be reserved. If the requested service is not already available in the network system, then the task is created using the service-chained module of the proposed architecture's orchestrator. A single or chain of VNFs with the forwarding plane of SDN is responsible for it. To activate the initiated request, the proposed layer interacts with VNF for granting the task resources.
- iv) **Task Update:** Task update is requested by the user. Again, a security check is performed on the modification request. Once the task that needs modification is verified, the update is performed through the service-scaling module of the proposed layer. The SDN controller of the proposed layer is responsible for monitoring the service in terms of the lifetime performance of the proposed architecture.
- v) **Task Termination:** Once the device user sends a request to terminate the service, then the service manager of the orchestrator layer will terminate the request by completing the task.

3.5 Mathematical Computation

The proposed SDNFV model considers five indexes: cost, time, bandwidth, energy consumption, and energy calculation.

i) Cost Index

The cost can be calculated using Eq. (3). The cost of the distance from source to destination is calculated by multiplying W_c as the unit of communication cost with the distance. The lower the cost is, the better the architecture will be.

$$C(S_i, D_j) = W_c \times d_{si,dj} \quad (3)$$

ii) Time Index

It is the time the user needs from sending data to receiving data from the cloud center. Minimum time leads to a better model. It is calculated by adding transmission and computational delays as

$$T = T_p + T_c \quad (4)$$

iii) Bandwidth Index

m virtual machines exist at the cloud center to serve m users at a time, then the total bandwidth is fixed, as shown as follows [1]:

$$B = \begin{cases} B_{\max}, & m < T_1 \\ W/m, & T_1 \leq m \leq T_2 \\ B_{mm}, & n > T_2 \end{cases} \quad (5)$$

where B is the amount of bandwidth, and T_1 and T_2 are the numbers of users at the cloud center.

iv) Energy Consumption

Energy consumption is the energy spent when the cloud center offers a service to device users. This is also called cost index, which means lower energy leads to a better model.

$$E_{si} = E_b + m * C_{vm} \quad (6)$$

where E_{ai} is the total energy consumed, E_b is the total standby power consumption, and c_{vm} represents the unit of energy consumption.

v) Capacity of the Machines at the Cloud Center

The capacity is calculated as

$$\text{Energy capacity} = \text{total energy capacity of the machine} / \text{number of users} \quad (7)$$

To evaluate the performance of the proposed architecture of SDNFV on edge nodes, the simulation results are analyzed and compared with the results of existing architectures. The effective performance of the proposed architecture is shown in Fig. 5. Our proposed SDNFV architecture on edge nodes is the middle architecture between the IoT devices and cloud data center. While using the proposed architecture on edge nodes for data processing, the response accelerates and the throughput is reduced because the configuration and processing of data are performed on the edge server.

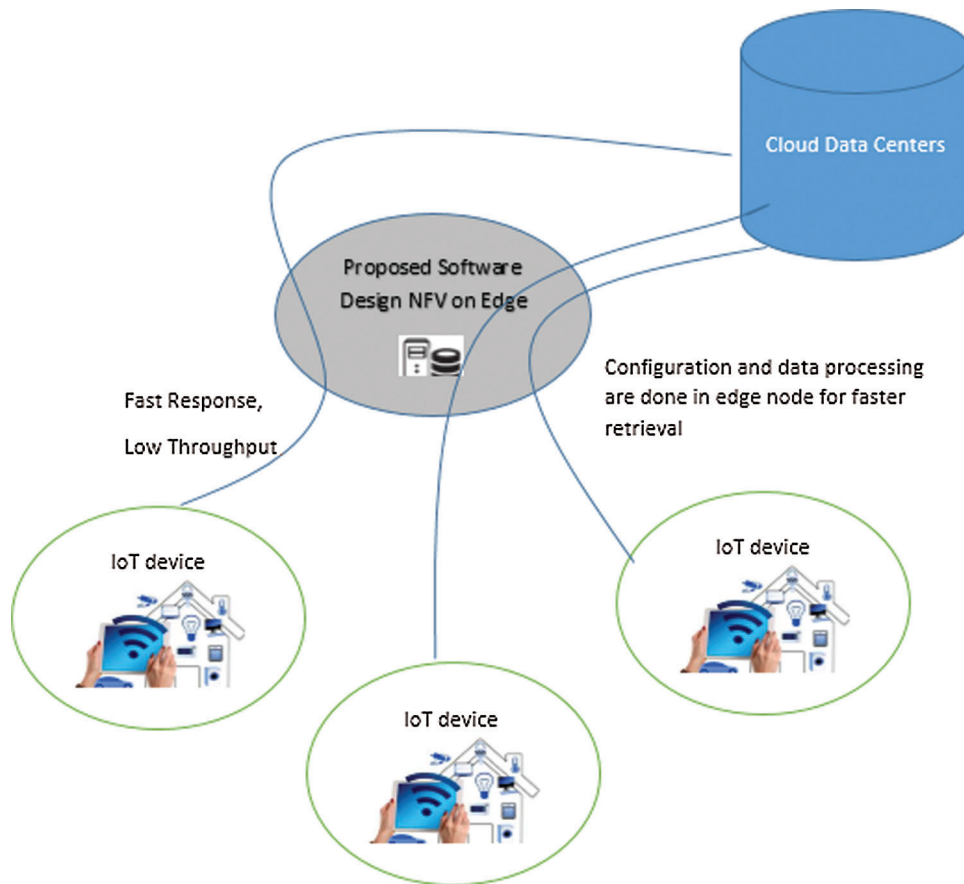


Figure 5: Effective performance of the proposed architecture

4 Experimental Evaluation

4.1 Mathematical Computation

MATLAB is used to develop the simulation environment for algorithm execution. A system of IIoT with SDN is established on edge devices. The EstiNet simulator is used for the simulation. In SDN, OpenFlow is processed between a controller and a switch. The simulated OpenFlow with SDN controller enables an Ethernet switch to support control panels as in-band or out-of-band by using multiple controllers. To control the simulated OpenFlow switches, open-source controller software can be used in the simulated SDN. The SDN Wi-Fi infrastructure can be simulated using OpenFlow-enabled access and switch devices. This simulation platform is best suitable for researchers in SDN.

4.2 Simulation Result of the Proposed Architecture and Simulation Parameters

The proposed architecture lies on edge nodes near IoT devices. The response from the cloud center for a task request is processed with high execution time and low time delay, and the request is validated before processing. Hence, the user will obtain a response with low computational power and high performance. While the device count increases the time, the complexity throughput is also increased gradually. The simulation result is shown in Tab. 1. Figs. 6–8 show the simulation result of the proposed SDNFV architecture in terms of the number of IoT device requests, such as 50, 80, 100, 150, and 200. Our proposed architecture presents computational complexity of 1500 J, cost of 240, execution time of 230 s, and throughput of 2200 kbps for 200 devices. To analyze the performance of the proposed architecture further, it is compared with existing architectures, such as ASTP and SuVMF.

Table 1: Simulation result of the proposed architecture

No. of devices	Time complexity	Cost	Time	Throughput
50	750	140	80	50
80	850	180	100	1000
100	1100	200	130	1400
150	1300	220	180	1800
200	1500	240	230	2200

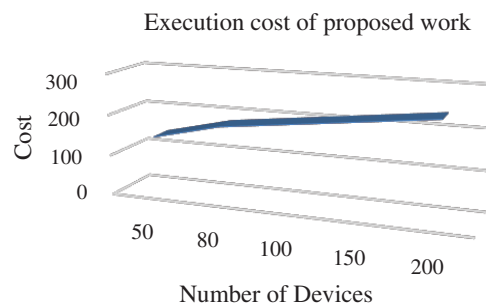


Figure 6: Computational complexity of the proposed architecture

4.3 Results and Discussion

Total Time Delay: The total time delay of the architectures is shown in Fig. 9. As the device amount increases, the delay time is also increased for all the architectures. However, our proposed SDNFV has less

total time delay compared with the other existing architectures because it selects each device task through the edge node of the framework via NFV. For the requests of 200 IoT devices, the time delay of the proposed architecture is 1800 s only, that of ASTP is 2500 s, and that of SuVMF is 3000 s.

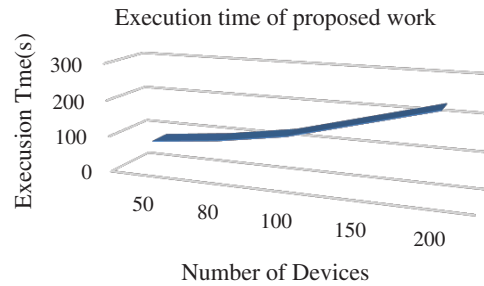


Figure 7: Execution time of the proposed architecture

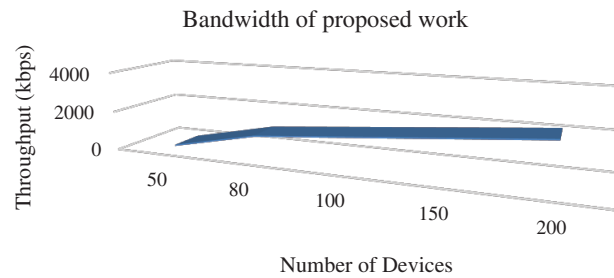


Figure 8: Throughput of the proposed architecture

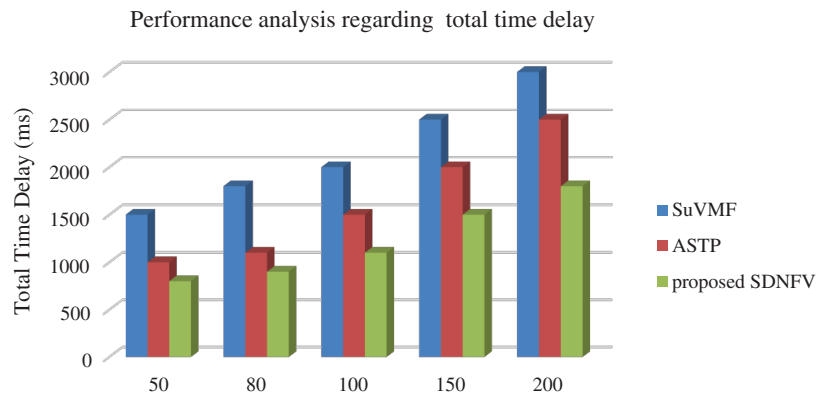


Figure 9: Performance analysis in terms of total time delay

Reliability. This is an important index for evaluating the success of any architecture in task processing. When the failure rate to complete a task is low, it means the reliability of the architecture is high. From Fig. 10. with the increase in the device number, the reliability of all architectures decreases. With the concept of SDN with NFV on edge nodes, the device requests are validated, and response is performed through VNF's forwarding plane in the orchestrator layer. The reliability of the proposed architecture is higher than that of the other existing architectures. The proposed architecture achieves 90% reliability on 200 devices, which is higher than that of the two other architectures (85% for ASTP and 70% for SuVMF).

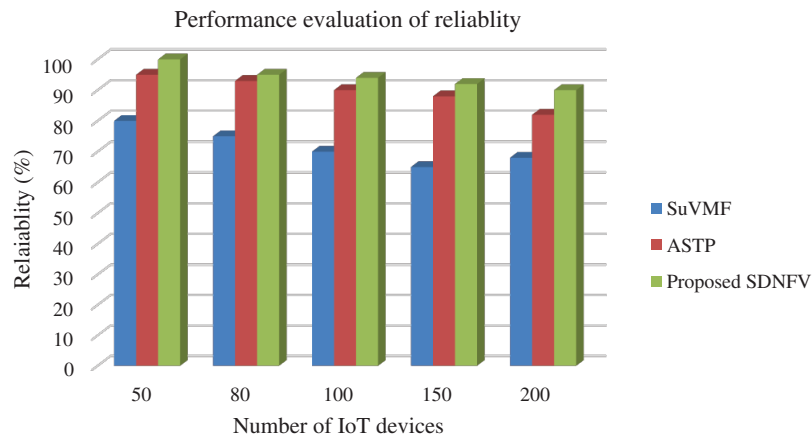


Figure 10: Performance analysis in terms of reliability

Satisfaction. It is a comprehensive index for real-time systems. The best architecture must rapidly respond to requests and achieve user satisfaction. When the number of device requests is increased, the tasks must be processed rapidly; if the system is not able to respond on time, the satisfaction will be low. Fig. 11. shows the analysis of the architectures in terms of user satisfaction.

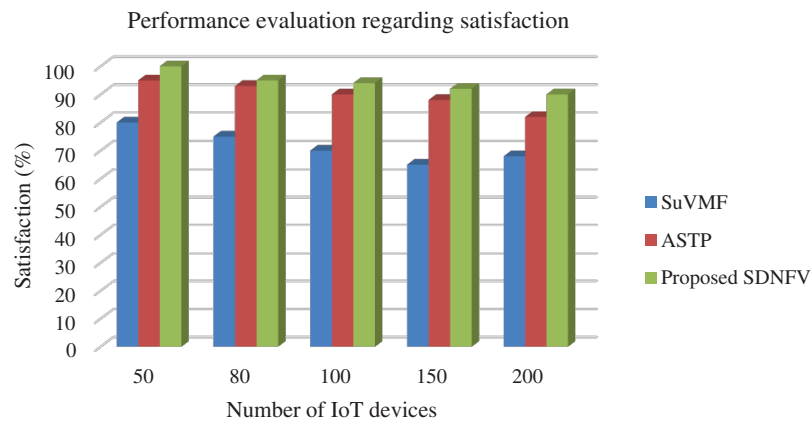


Figure 11: Performance analysis in terms of satisfaction

The proposed SDNFV on edge nodes has the highest satisfaction among the architectures. The proposed architecture lies on the edge nodes near the IoT devices. The response from the cloud center for a task request is processed with high execution time and low time delay, and the request is validated before processing. Hence, the user will receive the response as early as possible with full satisfaction. Our proposed architecture achieves 90% satisfaction compared with ASTP at 82% and SuVMF at 68%. In sum, our proposed SDN with NFV on edge nodes performs better in terms of total time delay, reliability, and satisfaction.

5 Conclusions

The MFC architecture uses SDN/NFV on edge nodes to reduce the complexity of data computation. Every hardware is separated and controlled using software virtualization concepts. It produces high reliability with low time delay. This architecture improves the QoS among edge computing devices in

cloud computing infrastructure. The SDN controller combined with NFV VNFs on edge nodes is responsible for managing rules through API. Considerable data can be computed with minimal complexity when using SDNFV in fog devices. Information growth can be a competitor for every growing technology. This architecture can help reduce the computational complexity. The simulation results prove that our proposed architecture obtains better throughput, execution time, cost, reliability, and satisfaction compared with existing architectures. The proposed SDNFV obtains high satisfaction of 90% compared with the others. In the future, we can focus on security in this architecture and add multiple SDNFV nodes to the orchestration architecture at different locations. The multiple nodes will help further increase the efficiency of this architecture.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

- [1] J. O. Fajardo, F. Liberal, I. Giannoulakis, E. Kafetzakis, V. Pii *et al.*, “Introducing mobile edge computing capabilities through distributed 5G cloud enabled small cells,” *Mobile Networks and Applications*, vol. 21, no. 4, pp. 564–574, 2016.
- [2] N. ISG, “Network functions virtualisation (nfv)-network operator perspectives on industry progress,” *ETSI, Tech*, no. 1, 2013.
- [3] R. Chaudhary, G. S. Aujla, N. Kumar and J. J. Rodrigues, “Optimized big data management across multi-cloud data centers: Software-defined-network-based analysis,” *IEEE Communications Magazine*, vol. 56, no. 2, pp. 118–126, 2018.
- [4] R. Mahmud, R. Kotagiri and R. Buyya, “Fog computing: A taxonomy, survey and future directions,” in *Internet of Everything*, 1st edition, Springer, Cham, pp. 103–130, 2018.
- [5] B. Varghese and R. Buyya, “Next generation cloud computing: New trends and research directions,” *Future Generation Computer Systems*, vol. 79, pp. 849–861, 2018.
- [6] H. Wu, Z. Zhang, C. Guan, K. Wolter and M. Xu, “Collaborate edge and cloud computing with distributed deep learning for smart city internet of things,” *IEEE Internet of Things Journal*, vol. 7, no. 9, pp. 8099–8110, 2020.
- [7] R. Mijumbi, J. Serrat, J. L. Gorricho, N. Bouten, F. De turck *et al.*, “Network function virtualization: State-of-the-art and research challenges,” *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 236–262, 2015.
- [8] Z. Lv and W. Xiu, “Interaction of edge-cloud computing based on SDN and NFV for next generation IoT,” *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 5706–5712, 2019.
- [9] J. Wang and D. Li, “Adaptive computing optimization in software-defined network-based industrial internet of things with fog computing,” *Sensors*, vol. 18, no. 8, pp. 1–14, 2018.
- [10] J. Liu and Q. Zhang, “Offloading schemes in mobile edge computing for ultra-reliable low latency communications,” *IEEE Access*, vol. 6, pp. 12825–12837, 2018.
- [11] A. C. Baktir, A. Ozgovde and C. Ersoy, “How can edge computing benefit from software-defined networking: A survey, use cases, and future directions,” *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2359–2391, 2017.
- [12] S. Peng, J. Fajardo, P. Sayyad Khodashenas, B. Blanco, F. Liberal *et al.*, “Qoe-oriented mobile edge service management leveraging SDN and NFV,” *Mobile Information Systems*, vol. 2017, pp. 1–14, 2017.
- [13] T. Choi, S. Kang, S. Yoon, S. Yang, S. Song *et al.*, “SuVMF: Software-defined unified virtual monitoring function for SDN-based large-scale networks,” in *Proc. CFI, ACM*, Tokyo, Japan, pp. 1–6, 2014.
- [14] E. Haleplidis, J. H. Salim, S. Denazis and O. Koufopavlou, “Towards a network abstraction model for SDN,” *Journal of Network and Systems Management*, vol. 23, no. 2, pp. 309–327, 2015.
- [15] S. Papavassiliou, “*Software Defined Networking (SDN) and Network Function Virtualization (NFV)*,” Multidisciplinary Digital Publishing Institute, vol. 12, no. 1, pp. 1–3, 2020.

- [16] A. Stamou, G. Kakkavas, K. Tsitseklis, V. Karyotis and S. Papavassiliou, "Autonomic network management and cross-layer optimization in software defined radio environments," *Future Internet*, vol. 11, no. 2, pp. 1–18, 2019.
- [17] A. Leivadreas, G. Kesidis, M. Ibnkahla and I. Lambadaris, "VNF placement optimization at the edge and cloud," *Future Internet*, vol. 11, no. 3, pp. 1–13, 2019.
- [18] J. Yao, Z. Han, M. Sohail and L. Wang, "A robust security architecture for SDN-based 5G networks," *Future Internet*, vol. 11, no. 4, pp. 85, 2019.
- [19] G. Mitsis, P. A. Apostolopoulos, E. E. Tsiropoulou and S. Papavassiliou, "Intelligent dynamic data offloading in a competitive mobile edge computing market," *Future Internet*, vol. 11, no. 5, pp. 1–19, 2019.
- [20] V. Eramo, F. G. Lavacca, T. Catena, M. Polverini and A. Cianfrani, "Effectiveness of segment routing technology in reducing the bandwidth and cloud resources provisioning times in network function virtualization architectures," *Future Internet*, vol. 11, no. 3, pp. 1–20, 2019.
- [21] V. Gkioulos, H. Gunleifsen and G. K. Weldehawaryat, "A systematic literature review on military software defined networks," *Future Internet*, vol. 10, no. 9, pp. 1–39, 2018.
- [22] A. Mimidis Kentis, J. Soler, P. Veitch, A. Broadbent, M. Mobilio *et al.*, "The next generation platform as a service: Composition and deployment of platforms and services," *Future Internet*, vol. 11, no. 5, pp. 1–20, 2019.
- [23] P. Porambage, J. O. Kwuibe, M. Liyanage, M. Ylianttila and T. Taleb, "Survey on multi-access edge computing for internet of things realization," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 2961–2991, 2018.
- [24] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang *et al.*, "A survey on internet of things: Architecture, enabling technologies, security and privacy, and applications," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1125–1142, 2017.
- [25] D. Miorandi, S. Sicari, F. De Pellegrini and I. Chlamtac, "Internet of things: Vision, applications and research challenges," *Ad hoc Networks*, vol. 10, no. 7, pp. 1497–1516, 2012.