SDN-Based Fog Computing

Subjects: Computer Science, Software Engineering

Contributor: Amirah Alomari

Software-defined networks (SDN) is an evolution in networking field where the data plane is separated from the control plane and all the controlling and management tasks are deployed in a centralized controller.

Keywords: software-defined networks ; fog computing ; cloud computing ; energy efficiency ; vehicular networks ; edge computing

1. Introduction

Traditional networks rely on network devices to make forwarding and routing decisions by implementing hardware tables that are embedded to the device itself, such as a bridge and router. As well, traffic rules such as filtering and prioritizing are implemented locally in the device. However, software-defined networks (SDN) brought advancement in networking by addressing simplicity, as it aims to reduce the design complexity involved in implementing both the hardware and software of network devices.

The basic concept of SDN is to remove the controllability from network devices and allow one central device, i.e., control unit to control and manage. The control unit is capable of observing the entire network and making decisions regarding forwarding and routing, whereas the task of forwarding is handled by network hardware devices in addition to filtering and traffic prioritization [1]. OpenFlow is a well-known SDN design that follows its basic architecture of decoupling control from the data plane where the controller and switches are communicating through OpenFlow protocols. Switches, on the other hand, contain flow tables and flow entries such as matching fields, counters and sets of actions, while the controller is capable of performing sets of actions on flow entries, such as update, delete and add [2].

Even though cloud computing is a powerful technology, there are some issues and challenges that are still open, such as network mobility, scalability and security issues; thereafter, the possibility to extend software-defined networks in cloud computing has been investigated [3]. Although fog computing brought solutions to handle IoT applications that requires less latency and a higher bandwidth to process data from cloud server to end devices. However, end devices have limited resources that conflict with high demands, besides heterogeneity, which is another challenge introduced in fog computing [4]. Therefore, the current studies are developed for the sake of bringing advantages of SDN to both cloud and fog computing in order to overcome the above-mentioned issues by offering flexible and intelligent resources management solutions.

Furthermore, resource management is a challenging issue, as the difficulty lies with its relatedness to several problems, such as resource heterogeneity, asymmetric communication, inconsistent workload and resources dependency [5][6]. Therefore, this study reviews related research and its contribution to SDN-based cloud in terms of network performance, and energy efficiency. Additionally, it presents state-of-the-art contributions in the field of emerging software defined networks and fog computing. This review contributes to two of the most promising fields in networking; its importance lies in providing a comprehensive review of important and state-of-the-art studies related to SDN cloud and SDN fog. Additionally, it helps researchers to observe related contributions by assessing relative important key information of a specific study. Thus, it can be used as reference for researchers based on their area of concern; since this review classifies novel findings and summarizes it based on several criteria that help researchers to identify, select and contribute to SDN-based cloud/fog accordingly. Moreover, each reviewed research is mapped with corresponding evaluated metrics for ease of configuration of correlated studies based on common interest.

2. SDN-Based Fog Computing

Since cloud computing plays a major rule in data storage, however, fog computing aims to reduce the latency between nodes, especially for delay-sensitive applications. Thus, one of the main reasons to develop SDN-based fog is to ease communication between nodes in order to ensure a reasonable performance. This section presents state-of-the-art researches that are conducted for SDN-fog computing to solve existing challenges specifically towards network performance issues.

An SDN-based solution is proposed in order to solve the complexity of data forwarding in smart grids in fog computing [7]. Three-tier architecture is proposed that consists of IoT as the bottom tier that is responsible on generating data, the middle tier, which is fog, enables smart grids that consist of smart meters (SM) and data collection servers (DCS). While the cloud tier, which is the top layer, consists of SDN controller and responsible in performing processing and permanent storage. Data forwarding uses shortest paths—specifically, Dijkstra Algorithm—with three types of path recovery that are forced when a link failure is encountered during data transmission. The proposed work is simulated in Mininet considering three types of topologies, which are large, medium and small to measure the transmission time, throughput and transmission overhead. The results show that the Dijkstra algorithm has the lowest forwarding time compared to other shortest path algorithms, i.e., Floyed-Warshell and Bellman-Ford. On the other hand, the forwarding time is increased when the path recovery conducted end-to end link failure. However, no link failure case consumes more time than a path recovery conducted through switch-to-switch or switch-to-end cases when the topologies are large and medium, while the number of broken lines has a positive correlation with the forwarding time while a throughput has a negative correlation with the size of topology. While, in [8], multi-layer advanced networking environment middleware is proposed for the SDN-based fog environment to solve heterogeneity and different time requirements by different applications. Based on the application type, a forwarding mechanism is selected by the SDN controller such as by IP, payload content and overlay networks where selection and initiating are performed simultaneously. Real testbed is used for an intensive evaluation that considers mainly latencies and performances. The results show the capability of the proposed middleware to determine the best forwarding mechanism based on the application type and network status.

On the other hand, merging fog computing and SDN in the internet of vehicle is proposed to solve latency problem resulted from generated data by sensors [9]. It consists of five layers, including clusters of Internet of Vehicles (IoV), local and global load balancers, fog servers, SDN controller and cloud servers. Local and global load balancers are designed for local clusters by local load balancers and different clusters which will be performed by SDN controller. Considering CPU utilization as main criteria when determining size of load to be either overloaded or under loaded. A simulation was conducted via OMNET++ and SUMO with a size of vehicles equal to 1000 and evaluated against VANET-Cloud architecture. The results show that the proposed architecture and load balancers capable of increasing resource utilization, decreasing the average response time, outperforming in meeting tasks' deadlines, and uses lowest bandwidth compared to baseline.

Other research employed machine learning techniques to provide intelligence to SDN controllers, such in [10], a deep reinforcement learning was developed to perform task scheduling in the SDN controller by learning from intelligent agents that led to optimal decisions. The SDN controller is capable to generate and learn from historical datasets, filter and re-represent datasets, produce learning policies, selects nodes and activates rules. The evaluation is conducted via testbed using two datasets for training and testing. The results show that the maximum latency achieved equal to 12.5 milliseconds and saved energy up to 87%.

To solve the problem of packet delays imposed by the SDN controller in fog networking, a prediction model is proposed based on regression model which follow standard process of data mining methodology [11]. The process generally consists of data understanding, which aims to build an initial dataset, and data preparation, which aims to clean dataset, remove redundancy and construct features, as well, modeling, which includes selection and assessments machine learning algorithms that results in selecting random forest algorithm. The results shows that proposed model manages to predict delay values similar to actual values.

Moreover, a solution for SDN-fog services based on machine learning—specifically, reinforcement learning—is proposed in [12] in order to allocate service dynamically based on objectives such as power consumption, security and QoS constraints. Proposed reinforcement learning imposes recurrent neural networks in order to reduce the average response time. The service manager is built-in SDN controller specifies the locations of services to be processed dynamically by taking into consideration overall objective.

In [13], the proposed work goal is to increase storage, by reducing the cache redundancy and improving service placement with limited interactions with controller, therefore merging software-defined networks and named data network in fog computing is introduced. By extracting powerful features of both paradigms i.e., devoting controllability to SDN controller to perform routing, function placement, caching decision and service allocation. While the data plane is managed by forwarding information base, content store and pending interest table, which are offered by NDN stateful data plane. The evaluation is conducted using MATLAB against SDN-IP protocol systems by comparing the number of packets exchanged between the nodes and controller and number of new rules inserted by the controller. The results show SDN-NDN systems are capable of reducing communication between nodes and the controller.

In [14], data collection schema and estimation model for SDN-based vehicular fog computing is proposed. The architecture of SDN-based VFC consists of three layers, which are an infrastructure layer that includes vehicles, a fog layer that includes both regional controllers and local SDN controllers and a cloud layer, which includes a global SDN controller. The main tasks consist of the data collection configuration process, fog station deployments and data routing. A collection of data configuration consists of four main phases that are initialization of data collection parameters that are performed by a global SDN controller as data-oriented or fog-oriented. Then, the data collection parameter the sends collected data to fog stations. After that, users' requests are sent to fog stations. Finally, update data collection parameters to fit current environment, which include the deployment of fog station and data migration. A model is built to estimate number of fog stations required to be deployed and evaluation is conducted based on real traffic data. The results show that accurate estimation of fog stations is achieved, which depends on data availability and volume of data collection. While the data collection schema is evaluated in Veins Simulation Framework against classical schema to assess the latency, a number of hops to reach the fog station, recovery time and data loss. The results show that a proposed schema decreases the latency and number of hops. As well, less time is required for recovery, and less data loss is achieved.

The machine learning approach is utilized in [15], specifically the K-Nearest Neighbor (KNN) algorithm, together with multi-objective optimization—specifically, Non-dominated Sorting Genetic Algorithm (NSGA-II)—to meet the QoS constraint based on application type, either maximizing the path reliability or minimizing the delay. The system consists of three parts that are network architecture that consists of SDN switches, fog nodes, IoT devices and links associated with QoS parameters, i.e., bandwidth and transmission delay. Then, the communication model and provisioning model. In order to find the link reliability, the system uses machine learning techniques, i.e., KNN algorithm. While finding the optimal path as a tradeoff objective, the system uses the NGSA-II algorithm. An evaluation is conducted using Mininet with two types of applications: delay-sensitive that requires a path with less delay, and computational-intensive, which requires a reliable path. The results show that the system manages to meet each application QoS constraint by selecting required path according to optimal objective. Moreover, less delay and less packet loss are achieved compared to the baseline.

The software defined and multi-access edge computing vehicular network architecture is developed to reduce the latency in vehicular networks [16]. The system uses fuzzy logic for clustering vehicles and OpenFlow algorithm to update flow tables. The architecture consists of four layers, which are: access layer, that consists of forwarding devices. Then the forwarding layer, which consists of high-speed SDN switches. Then multi-edge computing layer that has local controllers that perform resources allocation, and the MEC application platform system, which performs cellular resource allocation, caching in-vehicle message modules, communication decision modules and multimedia broadcast multicast service computing modules. Finally, the control layer which contains global SDN controller. Using fuzzy logic for clustering the vehicles by head vehicle based on velocity, position and signal quality. After that, the SDN client switched flow tables are updated. The simulation was performed in OMNET++ with INET framework using real-time traffic via SUMO. The results show that proposed system manages to achieve the highest packet delivery ratio and lowest average E2E delay compared to the baselines.

Similarly, an offloading mechanism is proposed in [17] that works dynamically by considering nodes selection prior to offloading. The system is composed of fog nodes, which are connected to the SDN switch and whose information is gathered by fog agents. Additionally, core switches that contain several SDN switches that are controlled via SDN controller. As well, the controller that manages both fog nodes and network via the fog orchestration and SDN controller. Finally, the service layer, which includes fog support applications. On the other hand, a dynamic offloading service is triggered when it receives a call from fog orchestration, which is performed when a node sent an offload request to fog orchestration as a result of an overloading condition. However, the decision of which node will receive the offload that was is made after filtering the nominated nodes and ordered it based on the network status and available resources. The evaluation was conducted via the Minint emulation tool and Python, and iPref was used for traffic generation. Selection time, time to transfer data to the offloading node, throughput and available resources ratio are the evaluation metrics. The results showed that the proposed work was outperformed in minimizing the selection time, thus minimizing the delay. As well, it provides better throughput and less response time. As well, an artificial intelligent approach is proposed to overcome the delay and reliability issues in SDN-based IoV-Fog computing [18]. ARTNet architecture consists of the IoV layer which contains onboard devices, the fog layer, which contains base stations and RSU as the static models to provide processing and storage facilities. As well, the architecture contains an AI-based SDN that consists of three modules that are intelligent agents, deep learning and big data analytics. The experimental results show that the proposed algorithm outperforms in reducing the power consumption and latency with respect to the time slot and number of vehicles, especially when the number of IoV is increased. In addition to that, IoV architecture is proposed in [19], which consists of multiple fog clusters that contain several RSUs. RSUs gather information regarding roads and vehicles, then transform this information to RSU controller, which performs decision-making towards resource allocations in the fog layer. While the SDN controller resides in the cloud to obtain global view and forward status of the network with forwarding policies and rules to the RSU controller. Then, the resource allocation model is constructed, which considers several factors, which are the power consumption, delay, load balancing and time execution stability. After that, the proposed algorithm is developed with integration with the evolutionary algorithm by using hieratical clustering features to enhance the many-objective optimization algorithm. The experimental results show that the proposed algorithm outperforms in time execution stability and load balancing objectives.

Even though previous studies followed OpenFlow architecture, however, in [20], a multi-layer SDN Fog paradigm was proposed in order to organize the SDN controllers, locally and remotely, in fog computing using P4 and P4Runtime to overcome certain difficulties imposed by the OpenFlow protocol. Controllers can be built locally near edge devices in order to ease communication, while remote SDN controllers are centralized to control traffic, among other functions that require a global view of the network. Figure 1 represents the relationship between each category.

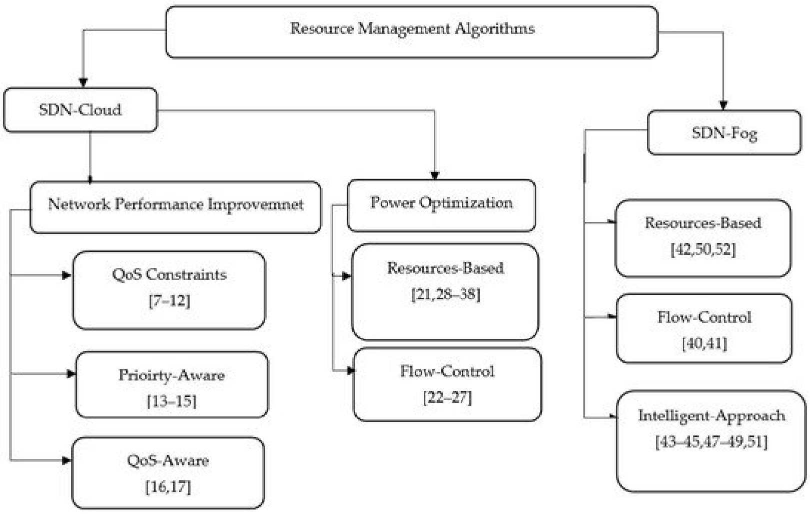
Resource Management Algorithms
SDN-Cloud
SDN-Fog
Network Performance Improvemnet
Power Optimization
QoS Constraints [7–12]
Prioirty-Aware [13–15]
QoS-Aware [16,17]
Resources-Based [21,28–38]
Flow-Control [22–27]
Resources-Based [42,50,52]
Flow-Control [40,41]
Intelligent-Approach [43–45,47–49,51]

Figure 1. Taxonomy of reviewed research based on the main objective and research strategy.

References

1. Culver, P. Software Defined Networks, 2nd ed.; Morgan Kaufmann: Burlington, MA, USA, 2016.
2. Nunes, B.A.A.; Mendonca, M.; Nguyen, X.-N.; Obraczka, K.; Turletti, T. A survey of software-defined networking: Past, present, and future of programmable net-works. IEEE Commun. Surv. Tutor 2013, 16, 1617–1634.
3. Leon-Garcia, A.; Bannazadeh, H.; Zhang, Q. Openflow and SDN for Clouds. Cloud Serv. Netw. Manag. 2015, 6, 129–152.
4. Arivazhagan, C.; Natarajan, V. A Survey on Fog computing paradigms, Challenges and Opportunities in IoT. In Proceedings of the 2020 International Conference on Communication and Signal Processing (ICCSP), Chennai, India, 28–30 July 2020; pp. 385–389.
5. Schaller, S.; Hood, D. Software defined networking architecture standardization. Comput. Stand. Interfaces 2017, 54, 197–202.
6. Gonzalez, N.M.; Carvalho, T.C.M.D.B.; Miers, C.C. Cloud resource management: Towards efficient execution of large-scale scientific applications and workflows on complex infrastructures. J. Cloud Comput. 2017, 6, 13.
7. Okay, F.Y.; Ozdemir, S.; Demirci, M. SDN-Based Data Forwarding in Fog-Enabled Smart Grids. In Proceedings of the 2019 1st Global Power Energy and Communication Conference (GPECOM), Nevsehir, Turkey, 12–15 June 2019; pp. 62–67.
8. Bellavista, P.; Giannelli, C.; Montenero, D.D.P. A Reference Model and Prototype Implementation for SDN-Based Multi Layer Routing in Fog Environments. IEEE Trans. Netw. Serv. Manag. 2020, 17, 1460–1473.
9. Kadhim, A.J.; Seno, S.A.H. Maximizing the Utilization of Fog Computing in Internet of Vehicle Using SDN. IEEE Commun. Lett. 2018, 23, 140–143.

10. Sellami, B.; Hakiri, A.; Ben Yahia, S.; Berthou, P. Deep Reinforcement Learning for Energy-Efficient Task Scheduling in SDN-based IoT Network. In Proceedings of the 2020 IEEE 19th International Symposium on Network Computing and Applications (NCA), Cambridge, MA, USA, 24–27 November 2020; pp. 1–4.
11. Casas-Velasco, D.M.; Villota-Jacome, W.F.; Da Fonseca, N.L.S.; Rendon, O.M.C. Delay Estimation in Fogs Based on Software-Defined Networking. In Proceedings of the 2019 IEEE Global Communications Conference (GLOBECOM), Waikoloa, HI, USA, 9–13 December 2019; pp. 1–6.
12. Frohlich, P.; Gelenbe, E.; Nowak, M.P. Smart SDN Management of Fog Services. In Proceedings of the 2020 Global Internet of Things Summit (GIoTS), Dublin, Ireland, 3 June 2020; pp. 1–6.
13. Amadeo, M.; Campolo, C.; Ruggeri, G.; Molinaro, A.; Iera, A. Towards Software-defined Fog Computing via Named Data Networking. In Proceedings of the IEEE INFOCOM 2019—IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Paris, France, 29 April–2 May 2019; pp. 133–138.
14. Boualouache, A.; Soua, R.; Engel, T. Toward an SDN-based Data Collection Scheme for Vehicular Fog Computing. In Proceedings of the ICC 2020—2020 IEEE International Conference on Communications (ICC), Dublin, Ireland, 7–11 June 2020; pp. 1–6.
15. Akbar, A.; Ibrar, M.; Jan, M.A.; Bashir, A.K.; Wang, L. SDN-Enabled Adaptive and Reliable Communication in IoT-Fog Environment Using Machine Learning and Multiobjective Optimization. IEEE Internet Things J. 2021, 8, 3057–3065.
16. Nkenyereye, L.; Nkenyereye, L.; Islam, S.M.R.; Kerrache, C.A.; Abdullah-Al-Wadud, M.; Alamri, A. Software Defined Network-Based Multi-Access Edge Framework for Vehicular Networks. IEEE Access 2019, 8, 4220–4234.
17. Phan, L.-A.; Nguyen, D.-T.; Lee, M.; Park, D.-H.; Kim, T. Dynamic fog-to-fog offloading in SDN-based fog computing systems. Future Gener. Comput. Syst. 2021, 117, 486–497.
18. Ibrar, M.; Akbar, A.; Jan, R.; Jan, M.A.; Wang, L.; Song, H.; Shah, N. ARTNet: AI-based Resource Allocation and Task Offloading in a Reconfigurable Internet of Vehicular Networks. IEEE Trans. Netw. Sci. Eng. 2020, 1.
19. Cao, B.; Sun, Z.; Zhang, J.; Gu, Y. Resource Allocation in 5G IoV Architecture Based on SDN and Fog-Cloud Computing. IEEE Trans. Intell. Transp. Syst. 2021, 1–9.
20. Ollora Zaballa, E.; Franco, D.; Aguado, M.; Berger, M.S. Next-generation SDN and fog computing: A new paradigm for SDN-based edge computing. In Proceedings of the 2nd Workshop on Fog Computing and the IoT, Sydney, Australia, 8 April 2020; pp. 9:1–9:8.