Controller placement problem in software defined networks

Hassan Hadi Saleh, Israa Adnan Mishkal, Dheyab Salman Ibrahim
Department of Computer, Science College, University of Diyala, Diyala, Iraq

Article Info

Article history:

Received Jul 5, 2021
Revised Jun 14, 2022
Accepted Jul 1, 2022

Keywords:

Control plane
Controller placement problem
Data plane
Multiple controllers
Single controller
Software-defined networking

ABSTRACT

The requirements for the network are increasing by the expanding and spreading the Internet. The Previous techniques of the network do not meet the modern needs, thus, a new technical presents software-defined networking (SDN). SDN recognizes as a promising new model that separates the control plane (traffic routing and network topology) from the data plane (network architecture layers). The architecture of SDN has some features that find in a single controller or many controllers instance of programmable, flexible, and scalable. In the current SDN, multiple controllers are essential. Therefore, the optimal number of the controllers and their locations is the most significant challenge, known as the controller placement problem (CPP). It deploys the optimal number of controllers within the network while meeting presentations requirements considered conflicting in nature example: credibility, load balancing, latency, energy efficiency, and computation time. Many studies researched the ways to develop solutions for improving scalability, place selection for SDN. This paper presents the CPP and gives a comprehensive review of SDN issues based on the recent well-known research to extract available solution strategies. Finally, it discusses the limitations and future study directions that can support researchers in this field.

Corresponding Author:

Hassan Hadi Saleh
Department of Computer, Science College, University of Diyala
Baquba city, Diyala Governorate, Iraq
Email: hassan.hadi@uodiyala.edu.iq

1. INTRODUCTION

In software-defined networking (SDN), the controller considers the brain of the network and controls the networks and devices. SDN is a networking approach that enables programmatic and dynamic network control. This approach is more in line with cloud computing than popular networks. One of the drivers behind SDN is the desire to move from the old idea of networking to the new, more flexible approach often used in software development forms [1], [2]. In classical network performance, both the control plane and the data plane are closely coupled in the network element. This close coupling of control and data planes means the hardware and the software implementations of a network element that is firmly coupled. Network management becomes a weary, error-prone, and time-consuming process [3], [4]. At the beginning of design SDN, a single controller can face the needs of network control and management due to applying only to small network topologies such as campus networks. With the popularity and the spread of SDN, the number of forwarding tools and requirements increase in the user plane. All these things bring more requirements to controllers' processing capacity. The single controller is unable to gratify the needs of the network and application on account of the presentation and the ability of restriction [5], leading to a single-point bottleneck and fault in high probability. Therefore, it is necessary to research the SDN control plane's scalability. Also, it is useful with a multi-controller placement problem.

In the present years, SDN appeared as a kind of networking, through many standards to ensure the flexibility, cost-effectiveness, and well-organized of the network [2], [6]. The basic principle in its design is shown as decoupling the data plane from the control plane through the network programmability possibility that leads to the network management, dynamic configuration, and control [2], [7], [8]. Especially, the central level of control hides the network complexity of the application developer and delegates the main network control duties to the controller [9]. Controller duties include configuration parameters setting, forming standards, or making traffic rules for forwarding. Accordingly, the data plane deals with forwarding the packets depending on the set of rules created by the controller. Forwarding tools such as switches or routers are managed by the controller, through the rules that support their packet handling role in the network. The use of SDN offers many basic benefits to network service providers, data-centers, cloud, and edge computing [10]-[12]. It has a very large application in many emerging architectures such as optical networks, internet of things (IoT), wireless networks, vehicle ad-hoc networks (VANETs), mobile ad-hoc networks (MANETs), and wireless sensor networks (WSN) [13]. It also encourages researchers to test new ideas and deploy applications in the existing network through the programmability of the control plane. Neither programmability of the control plane nor decoupling control from forwarding is new to the networking community. Active networking solutions such as switch ware [14] and software routing suites such as click modular router [15] and bird internet routing daemon (BIRD) [16] are released prior to SDN that supports programmability [17] of the control plane. Routing control platform (RCP) [18], forwarding and control element separation (ForCES) framework released by internet engineering task force (IETF) [19], 4D design of network architecture [20], path computation element (PCE) architecture [21], [22], ethane [23] and open flow [22] are released prior to SDN that supports separation of control and data planes.

The rest of the paper organizes as follows: section 2 reviews some related works. Section 3 describes the architecture of SDN and the different metrics used to quantify the SDN presentation. Then, it discusses the controller placement approaches in section 4. Finally, section 5 and section 6 respectively present SDN Applications and concluding remarks.

2. RELATED WORKS

Many works considered the issue of controller placement problem (CPP) in SDN. Some of them are outlined here. Kumari and Sairam [22] investigated the CPP and survey this subject from the optimization viewpoint. They classified the literature of CPP into four main parts that contained reliability, latency cost, and multi-objective depending on their objective and analyzed specific algorithms in different application scenarios [24]. The other study studied CPP objectives that include minimizing the latency between controllers and their associated switches, enhancing reliability and resilience of the network, and minimizing deployment cost and energy consumption [25]. Also, Fancy and Pushpalatha [26] investigated the CPP and stated the CPP solution techniques, different metrics of presentations, and features of the various solutions. Ayo *et al.* [27] surveyed the CPP in two scenarios: the smallest number of controllers and the places of the controllers. Generally, solutions establish upon objective functions and their optimum solutions considering many different factors (such as propagation latency between switches and controllers and inter-controllers); and constraints such as the capacity of controllers and switches. To optimize the control layer performance, the authors considered the distance between controller and switch as the performance metric. That enabled minimization of the end-to-end delay, and computation of the optimal number of controllers. However, the proposed solution was topology-dependent on the workload factor of the controller was not considered in depth [28]. Rahman *et al.* [29] considered the average flow requests among switches and switch-to-controller and proposed a heuristic algorithm to dynamically shift the controller load. Indeed, Zebari *et al.* [30] proposed the use of wild card rules in the matching tuple to lessen the workload on the controller. Other researchers also proposed the deployment of distributed controllers as a general approach to obtain the needed expansion. However, the challenges like deriving the optimal number of controllers, their placement in the network, and load distribution remain [31], [32].

Moreover, the study revealed that HTTP applications on the internet covered approximately 47.7% of the internet traffic. They assumed rerouting of traffic using OpenFlow switches to improve resource utilization [5]. A plug-n-server solution analyzed the network status and assign client requests to minimize the controller’s response time [33]. It can be noticed that these surveys studied various view points related to the proposed paper's main issue. Thus, it aims to investigate and analyze the recent papers about CPP in SDN such as to shed the light on the proposed solutions.

3. SDN ARCHITECTURE

In the classical network circumstance, routers carry out a specific algorithm to bring the incoming packages to the following hop. For example, to access one website, permitting the records packages must be relocated using 10 hubs. This means 10 hubs have to operate the transmitting protocol to provide one ask created by the final user. In practice, there are thousands of internet consumers accessing the same web information at the same time, that produces a bunch of processing costs at the hub [33]. In traditional network situations, a modem may carry out two tasks. The first one is the hub regularly upgrading transmitting tables to acquire the standing of the network, and it is generally referred to the management airplane function. The second task is to forward the incoming data packets to the correct location based on the area stashed directing tables. This is generally called forwarding or information airplane procedure. A budget-friendly routing protocol takes choices depend on the transmitting table of the neighborhood tools. Also, it finds out the fastest path between the source and location along minimal time delay, and maximums throughput. If the measurements of the network topology are developed, the routing table measurements additionally increases, which produce a considerable amount of delay considering that transmitting details exchange occasionally to maintain upgrade the network standing. If one of the nodes in the network fails running, the computational issuse on the hub raises significantly; finding a route to another hub follows the same procedure. Since standard routing units have many of the issues, SDN solved the issuse by distributing the control plane and keeping track of degrees of freedom. SDN assigns the control plane to an outside device named operator, which controls all the directing decisions. In this step, hubs do not have to do computationally costly tasks. All methods are carried out in the main operator that considers all the transmitting choices. The controller constantly monitors the global sight of the entire network, which may be through some well-defined APIs [33]. The SDN divides the control plane from the forwarding devices (routers and switches). Therefore, it operates as a logical centralized unit like the network operating system or SDN controller. The SDN architecture is shown in Figure 1. It divides into three layers; namely an application layer at the top, an infrastructure layer at the bottom, and a control layer in the middle.

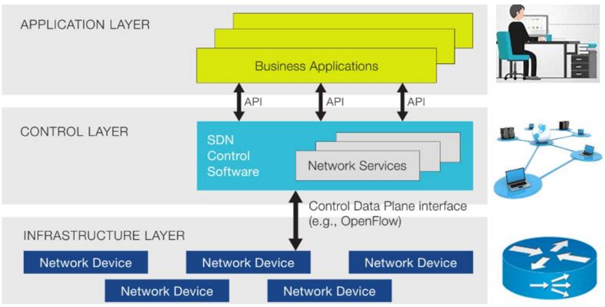

Figure 1. The architecture of SDN

3.1. Application layer (management plane)

It contains networking applications such routing, monitoring, load balancing, and firewalls. The main plane is responsible for defining rules and policies. SDN removes the employment of middleboxes such as load balancers, firewalls, and NAT by performing their basic uses as an application in software on a server [34]. The application layer is located above the control layer.

3.2. Control layer (control plane)

The Control layer gives a link for interaction between the application and control layer. It composes of one or more controllers for controlling infrastructure components. The controllers are basically responsible for maintaining the global view of the network using the data which are collecting by forwarding tools. They are also responsible for sending forwarding rules to switches. The controllers communicate to the forwarding devices using the southbound interface. It main duty is responsibility for programming the forwarding tools. Therefore, it can considere as the brain of the network. Centralized controller(s) resides on this plane [35].

3.3. Infrastructure layer (data plane)

Physical network infrastructure defines in the data plane. Forwarding devices (switches and routers) are connected by each other, either by wired or wireless mediums. It makes up of forwarding elements such as routers, switches, and wireless access points. These devices are responsible for collecting network status information and sending it to the controller [35].

4. CONTROLLER PLACEMENT PROBLEM

CPP is one of the most controversial problems. It can be found in the SDN system when multiple controllers have deployed in the management of the networks as well as the enforcement of network rules in such controllers. The purpose of CPP is to find the required feasible number of controllers and their places in the SDN networks for facing different presentations and QoS needs. Recently, the increase of end systems in new networks has led to the development of the number of forwarding elements (FEs) and network traffics in SDN-based networks. A single controller is unable to control huge networks, so multiple controllers should be used to cover the networks controlling [36]. Accordingly, scalability is one of the most essential topics that need more attention in the control plane. Accordingly, scalability is one of the most essential topics that need more attention in the control plane. It has two methods to controllers' deployment for improving the scalability that contains centralized and distributed schemes [36]. As shown in Figure 2, a controller is the point of the decision-making unit to manage the entire network link bottlenecks between equipment. Figure 3 shows the distributed controllers architecture containing logically centralized controllers' topology. The data path is given by the red line and the control path is shown with dashes in Figure 2 and Figure 3.

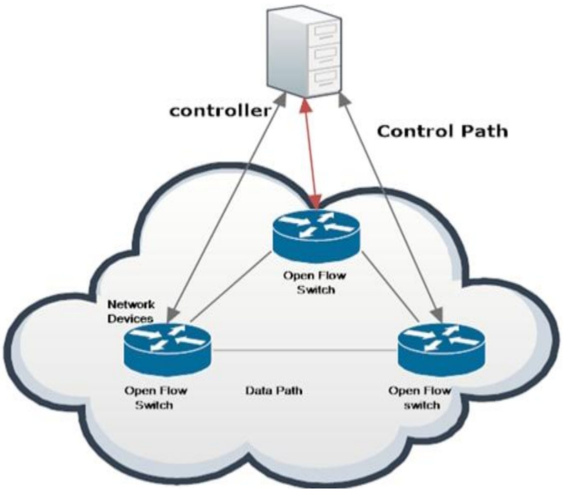

Figure 2. Single-controller architecture in SDN

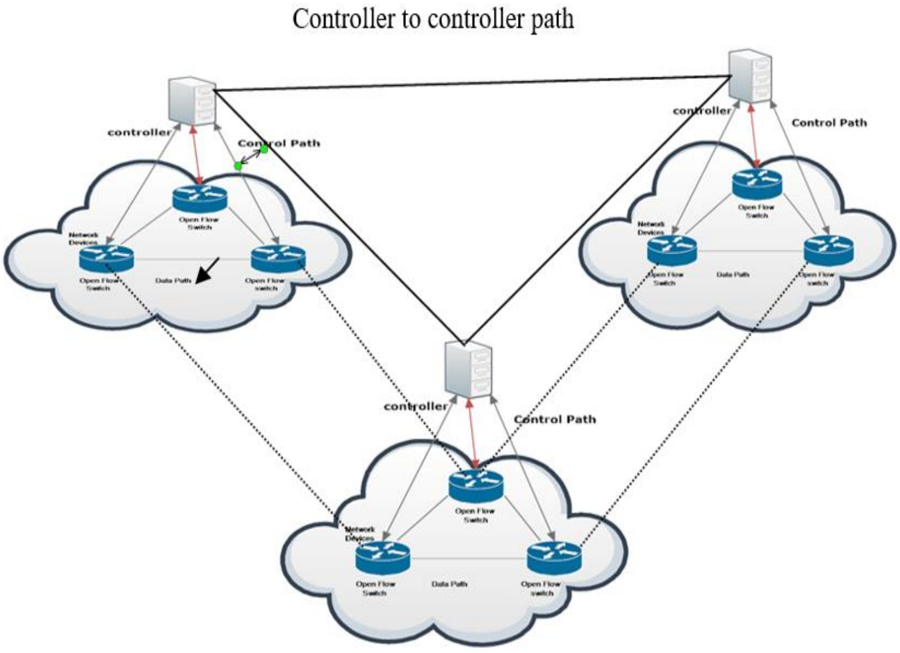

Figure 3. Distributed controllers architecture in SDN

In a distributed architecture of SDN controller, the network performance basically depends on the placement of controllers and the so-called CPP. Therfore, the controller placement is also a controversial in an SDN with a single physical controller, but the issuse has less effect [36]. Heller, Sherwood, and McKeown were the first authors who perform studies on CPP [28]. They formulated the problem as a facility location problem and it was shown to be NP-hard. Since there have been many efforts to place the controllers effectively [37]. The paper [38] points the one controller is considered to be sufficient for network management in most SDN-based networks. One of the problems in single controller system is a single point of failure. On the other hand, in large-scale networks, one controller can induce it to operate as a bottleneck in the network and result in a performance problem. Therefore, in large-scale networks is more than one controller that needs their placement in the network. This point is very important to improve network performance, the delay between FEs and controllers, reliability and robustness of the network, and scalability [39]. Different metrics use to select suitable controllers by network devices. So, it should give answer for the following questions as the controller placement problem to increase the performance of the SDN-based networks [39]:

a. How many controllers are required?
b. Where should the controllers place in the topology?
c. Which metrics should be considered to optimize?

Nowadays, many researches have been applied to solve the CPP problem. Those have assumed various solutions with many different optimization functions like performance, reliability and efficiency. In general, most CPP is modeled as a graph, $G = (V, E, U)$ topology; where V is the set of n switches, E is the set of edges (physical links) among switches or controllers and U is the set of k controllers [2]. Most studies on CPP are basically dealt with designing techniques to solve the value of k and the relation given by $U \rightarrow V$, for instance, it is the shortest path latencies between each pair of nodes when minimizing latency as the only objective in the objective function [6], [7]. Generally, the formulation is an optimization problem that is either data-driven or metric-driven formulated to find the minimum or maximum cost, the optimal number of controllers, switch-controller (SC) latency, or controller synchronization time. The formulation can be also based on hybrid metrics or multi-objective optimization problems about controller location or placement [6].

This model helps search for the number, kinds, and places of controllers which is a key to network performance and QoS. CPP is regarded as a non-deterministic polynomial NP-hard problem, and it is as the same as the facility location problem. Solving CPP needs a clever planning and suitable decisions to achieve optimal location and satisfactory results. In multiple controllers' placement, many factors affect the SDN performance such as reliability, controller load balancing, latency, operational costs, and response time of events. These can be further explored as follows [6].

a. Latency is one of the core factors considered in CPP. It depends highly on the distance between nodes in a network. Propagation and processing are significant to packet propagation. Propagation latency is the response time among the controllers or switches which is influenced by the distance between them. Also, the processing latency is highly affected by the controller's load. In other words, the optimality of the latencies influences both controller-to-controller (CC) and SC delays. Moreover, processing latency increases as the flows when increasing several switches. Additionally, whenever the SC latency exceeds a threshold, the overall latency of the network increases. Therefore, this affects the network availability in terms of responding to network events or pushing forwarding commands to switches on time.
b. Controller load balancing: an increase in the number of switches controlled by a controller increases the controller's load well. If the loading of the controller exceeds its processing ability, it could result in queueing delays or new requests not served from the switches since it has only the capacity to manage a limited number of switches' requests at the moment [6]. That means the controller fails to process requests, and the communication overhead of SC or CC increases. Therefore, it is significant SC assignments well-balanced. However, finding the best placement with minimum controllers and switches assignment is a difficult task when load balancing is considered.
c. Fault tolerance: in the SDN, each switch appoints to a controller, and a controller's loss influences the link connecting the controller and switch since no requests the switch will be given at the controller. As a result, the link loss impacts the controlled switches and restricts some functions of the control plane whenever the SC connection is lost. The switch receives are no new routing directions, and all packets are brought down. Therefore, the best placement that minimizes the number of controllers while ensuring reliability is important.
d. Optimal number of controllers: in this point, an increase in the number of switches is also increases network complexity to meet various network performance needs. According to this point, having various unplanned switches linked to a controller bring about increasing complexity on the controller since it has to process several requests within a given time interval. This needs a great challenge when determining how many controllers should be deployed and their placement to increase network

performance and QoS. However, several approaches such as traversal searching to find optimum performance numbers are considered time-consuming.

e. Cost: optimally and placing controllers in a large-scale network topology has a large effect on overall expenditure, which means that finding an optimal number of controllers to deploy means to put in the considerations all the possible costs such as budget limitations, purchasing, repairing, and maintenance costs [6]. The terms CAPEX and OPEX are considered in the SDN controller placement when minimizing the total cost. Thus, finding the best numbers of controllers to be deployed that minimizes cost is an important factor.

f. Inter-controller communication: in the SDN, inter-controller communication (CC) sustaines by state synchronization to reach global consistency. In this case, controllers communicate when they wish to pass a message to switches controlled by others. Therefore, such communication affects the performance of the end-to-end communication between various switches controlled by various controllers and is a controversial factor in the view of CPP. Generally speaking durings CPP that contains identifying the suitable number of controllers and their places. Also, the key considerations are the performance metrics. In general, during CPP that involves identifying the required number of controllers and their places, the key considerations are the performance metrics. The metrics are used to evaluate the quality of the different controller placement in the network and are critical to its performance, QoS, efficiency, scalability, and reliability [40].

5. SDN APPLICATIONS

An SDN application is a software program that designs to perform a task in an SDN environment. It is an approach to computer networking that does not only allow network administrators to change programmatically, control, initialize, and manage network behavior dynamically through open interfaces but also provides the concept of lower-level functionality. SDN applications also help in expanding and substituting upon functions that are accomplished in the hardware devices of a regular network through firmware. With the help of SDN, the administrators can manage and control the entire network through the abstraction of higher-level functionality. Even though management abstracts, there has to be some kind of communication between the two planes, i.e. the control plane (this system manages how traffic is distributed and where it is being sent) and the data plane (the underlying system helps in controlling traffic destination) [40]. SDN applications divide according to the internal and external environment. Applications that are hosting the rest of the Open Daylight controller software and are deployed internally run inside the container. The applications should be written in the native language which is Java for ODL. Internal SDN applications must also adhere to the execution and design constraints of the controller. It must also execute in the same Java Machine as the controller which means that these types of the application must run locally with the controller.

6. CONCLUSION

This paper presents a quantitative and a qualitative outlook of SDN. This study aims to supply a comprehensive view of SDN and CPP, and the placement of the controller is an essential aspect of the huge-sized SDN. Efficient controller placement enhances the performance metrics like propagation latency, reliability, load distribution, and failure resilience. This study briefly outlines a simple introduction to SDN, its architecture reviews some of the related work and weaknesses found in a single-controller SDN and a multi-controller (distributed) evolution. Therefore, it reveals to ensure scalability and reliability, multiple controllers require for large-scale SDN. Finally, the most significant SDN applications have been explored. A subsequent research effort in this direction could start with constructive comparisons of the so far proposed solutions of CPP. This will assist forward obtaining a reliable solution to CPP in SDN.

REFERENCES

[1] S. Jain *et al.*, “B4: experience with a globally-deployed software defined wan,” *ACM SIGCOMM Computer Communication Review*, vol. 43, no. 4, pp. 3–14, 2013, doi: 10.1145/2534169.2486019.

[2] K. S. Sahoo, D. Puthal, M. S. Obaidat, A. Sarkar, S. K. Mishra, and B. Sahoo, “On the placement of controllers in software-defined-WAN using meta-heuristic approach,” *Journal of Systems and Software*, vol. 145, pp. 180–194, Nov. 2018, doi: 10.1016/j.jss.2018.05.032.

[3] B. A. A. Nunes, M. Mendonca, X.-N. Nguyen, K. Obraczka, and T. Turletti, “A survey of software-defined networking: past, present, and future of programmable networks,” *IEEE Communications Surveys & Tutorials*, vol. 16, no. 3, pp. 1617–1634, 2014, doi: 10.1109/SURV.2014.012214.00180.

[4] D. Kreutz, F. M. V. Ramos, P. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, “Software-defined networking: a comprehensive survey*,” Proceedings of the IEEE*, Jan. 2015, vol. 103, no. 1, pp. 14-76, doi: 10.1109/JPROC.2014.2371999.

[5] M. Khorramizadeh and V. Ahmadi, "Capacity and load-aware software-defined network controller placement in heterogeneous environments," *Computer Communications*, vol. 129, pp. 226–247, Sep. 2018, doi: 10.1016/j.comcom.2018.07.037.

[6] A. A. Ateya *et al.*, "Chaotic salp swarm algorithm for SDN multi-controller networks," *Engineering Science and Technology, an International Journa*l, vol. 22, no. 4, pp. 1001-1012, 2019, doi: 10.1016/j.jestch.2018.12.015.

[7] T. Das and M. Gurusamy, "Resilient controller placement in hybrid SDN/legacy networks," in *2018 IEEE Global Communications Conference (GLOBECOM)*, Dec. 2018, pp. 1–7. doi: 10.1109/GLOCOM.2018.8647566.

[8] S. Wu, X. Chen, L. Yang, C. Fan, and Y. Zhao, "Dynamic and static controller placement in software-defined satellite networking," *Acta Astronautica*, vol. 152, pp. 49–58, Nov. 2018, doi: 10.1016/j.actaastro.2018.07.017.

[9] B. Zhang, X. Wang, and M. Huang, "Multi-objective optimization controller placement problem in internet-oriented software defined network," *Computer Communications*, vol. 123, pp. 24–35, Jun. 2018, doi: 10.1016/j.comcom.2018.04.008.

[10] A. Ali, M. Abbod, M. Khaleel, M. Mohammed, and T. Sutikno, "Large scale data analysis using MLlib," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 19, pp. 1735–1746, Oct. 2021, doi: 10.12928/TELKOMNIKA.v19i5.21059.

[11] J. Liao, H. Sun, J. Wang, Q. Qi, K. Li, T. Li, "Density cluster based approach for controller placement problem in large-scale software defined networkings," *Computer Networks*, vol. 112, pp. 24-35, 2017, doi: 10.1016/j.comnet.2016.10.014.

[12] H. H. Saleh and S. T. Hasson, "Improving communication reliability in vehicular networks using diversity techniques," *Journal of Computational and Theoretical Nanoscience*, vol. 16, no. 3, pp. 838–844, 2019, doi: 10.1166/jctn.2019.7963.

[13] R. Amin, E. Rojas, A. Aqdus, S. Ramzan, D. C. Perez, and J. M. Arco, "A survey on machine learning techniques for routing optimization in SDN," *IEEE Access*, vol. 9, pp. 104582-104611, 2021, doi: 10.1109/ACCESS.2021.3099092.

[14] V. Balasubramanian, M. Aloqaily, and M. Reisslein, "An SDN architecture for time sensitive industrial IoT," *Computer Networks*, vol. 186, 2021, doi: 10.1016/j.comnet.2020.107739.

[15] M. S. Iqbal *et al.*, "Mobile communication (2G, 3G &4G) and future interest of 5G in Pakistan: a review," *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 15, no. 3, pp. 1061-1068, 2020, doi: 10.11591/ijeecs.v22.i2.pp1061-1068.

[16] H. H. Khairi, S. H. Ariffin, N. A. Latiff, K. M. Yusof, M. K. Hassan, and M. Rava, "The impact of firewall on TCP and UDP throughput in an openflow software defined network," *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 20, no. 1, pp. 256–263, 2020, doi: 10.11591/ijeecs.v20.i1.pp256-263.

[17] N. Z. M. Safar, N. Abdullah, H. Kamaludin, S. A. Ishak, and M. R. M. Isa, "Characterising and detection of botnet in P2P network for UDP protocol," *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 18, no. 3, pp. 1584–1595, 2020, doi: 10.11591/ijeecs.v18.i3.pp1584-1595.

[18] Y. A. H. Omer, A. B. A. Mustafa, and A. G. Abdalla, "Performance analysis of round robin load balancing in SDN," in *2020 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE)*, 2021, pp. 1-5, doi: 10.1109/ICCCEEE49695.2021.9429662.

[19] S. Yogarayan, S. F. A. Razak, A. Azman, and M. F. A. Abdullah, "A mini review of peer-to-peer (P2P) for vehicular communication," *Indonesian Journal of Electrical Engineering and Informatics (IJEEI)*, vol. 9, no. 1, pp. 185–197, 2021, doi: 10.52549/ijeei.v9i1.2736.

[20] V. Sivasankarareddy and G. Sundari, "Survey on wireless sensor networks: energy efficient optimization routing algorithms," *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 19, no. 2, pp. 756–765, 2020, doi: 10.11591/ijeecs.v19.i2.pp756-765.

[21] N. McKeown *et al.*, "OpenFlow: enabling innovation in campus networks," *ACM SIGCOMM computer communication review*, vol. 38, no. 2, pp. 69–74, 2008, doi: 10.1145/1355734.1355746.

[22] A. Kumari and A. S. Sairam, "A survey of controller placement problem in software defined networks," *arXiv preprint arXiv:1905.04649*, 2019.

[23] G. Wang, Y. Zhao, J. Huang, and W. Wang, "The controller placement problem in software defined networking: a survey," *IEEE Network*, vol. 31, no. 5, pp. 21–27, 2017, doi: 10.1109/MNET.2017.1600182.

[24] A. K. Singh, S. Maurya, and S. Srivastava, "Varna-based optimization: a novel method for capacitated controller placement problem in SDN," *Frontiers of Computer Science*, vol. 14, no. 3, pp. 1–26, 2020, doi: 10.1007/s11704-018-7277-8.

[25] A. M. A. Jalil, R. Mohamad, N. M. Anas, M. Kassim, and S. I. Suliman, "Implementation of vehicle ventilation system using NodeMCU ESP8266 for remote monitoring," *Bulletin of Electrical Engineering and Informatics (BEEI)*, vol. 10, no. 1, pp. 327-336, 2021, doi: 10.11591/eei.v10i1.2669.

[26] C. Fancy and M. Pushpalatha, "Traffic-aware adaptive server load balancing for software defined networks," *International Journal of Electrical & Computer Engineering (IJECE)*, vol. 11, no. 3, pp. 2211-2218, 2021, doi: 10.11591/ijece.v11i3.pp2211-2218.

[27] I. O. Ayo, M. Adebiyi, O. Okesola, and O. Vincent, "Cloud middleware and services-a systematic mapping review," *Bulletin of Electrical Engineering and Informatics (BEEI)*, vol. 10, no. 5, pp. 2696-2706, 2021, doi: 10.11591/eei.v10i5.3169.

[28] K. Sood, S. Yu, Y. Xiang, and S. Peng, "Control layer resource management in SDN-IoT networks using multi-objective constraint," in *2016 IEEE 11th Conference on Industrial Electronics and Applications (ICIEA)*, 2016, pp. 71–76, doi: 10.1109/ICIEA.2016.7603554.

[29] N. A. A. Rahman, Z. A. Hamid, I. Musirin, N. A. Salim, and M. F. M. Yusoff, "Comparative studies between ant lion optimizer and evolutionary programming in optimal distributed generation placement," *International Journal of Advanced Technology and Engineering Exploration*, vol. 8, no. 75, pp. 236-246, 2021, doi:10.19101/IJATEE.2020.762124.

[30] R. R. Zebari, S. R. Zeebaree, A. B. Sallow, H. M. Shukur, O. M. Ahmad, and K. Jacksi, "Distributed denial of service attack mitigation using high availability proxy and network load balancing," in *2020 International Conference on Advanced Science and Engineering (ICOASE)*, 2020, pp. 174–179, doi: 10.1109/ICOASE51841.2020.9436545.

[31] M. Abdelhaq *et al.*, "The resistance of routing protocols against DDOS attack in MANET," *International Journal of Electrical & Computer Engineering (IJECE)*, vol. 10, no. 5, pp. 4844-4852, 2020, DOI: 10.11591/ijece.v10i5.pp4844-4852.

[32] K. Cabaj, J. Wytrebowicz, S. Kuklinski, P. Radziszewski, and K. T. Dinh, "SDN architecture impact on network security," in *Position papers of the 2014 Federated Conference on Computer Science and Information Systems*, 2014, vol. 3, pp. 143–148, doi: 10.15439/2014F473.

[33] T. Ghrib, M. Benmohammed, and P. S. Pandey, "Automated diagnosis of attacks in internet of things using machine learning and frequency distribution techniques," *Bulletin of Electrical Engineering and Informatics (BEEI)*, vol. 10, no. 2, pp. 950–961, 2021, doi: 10.11591/eei.v10i2.2766.

[34] N. Beheshti and Y. Zhang, "Fast failover for control traffic in software-defined networks," in *2012 IEEE Global Communications Conference (GLOBECOM)*, 2012, pp. 2665-2670, doi: 10.1109/GLOCOM.2012.6503519

[35] Y. A. J. Agudelo, “Scalability and robustness in software-defined networking (SDN),” Ph.D. Thesis, Network Engineering Department, Universitat Polit ́ecnica de Catalunya, Barcelona, 2016.
[36] B. P. R. Killi and S. V. Rao, “Towards improving resilience of controller placement with minimum backup capacity in software defined networks,” *Computer Networks*, vol. 149, pp. 102–114, 2019, doi: 10.1016/j.comnet.2018.11.027.
[37] V. Ahmadi and M. Khorramizadeh, “An adaptive heuristic for multi-objective controller placement in software-defined networks,” *Computers & Electrical Engineering*, vol. 66, pp. 204–228, 2018, doi: 10.1016/j.compeleceng.2017.12.043.
[38] T. Yuan, X. Huang, M. Ma, and J. Yuan, “Balance-based SDN controller placement and assignment with minimum weight matching,” in *2018 IEEE International Conference on Communications (ICC)*, 2018, pp. 1–6, doi: 10.1109/ICC.2018.8422637.
[39] S. Bouzghiba, H. Dahmouni, A. Rachdi, and J.-M. Garcia, “Towards an autonomic approach for software defined networks: An overview,” *Advances in Ubiquitous Networking 2. UNet 2016*, 2016, vol. 397, pp. 149–161, doi: 10.1007/978-981-10-1627-1_12.
[40] A. O. Akpovi, A. O. Adebayo and F. Y. Osisanwo, “Introduction to software defined networks (SDN),” *International Journal of Applied Information Systems (IJAIS)*, vol. 11, no. 7, pp. 10-14, 2016, doi: 10.5120/ijais2016451623.

BIOGRAPHIES OF AUTHORS

Hassan hadi Saleh is Associate Professor at College of Science, University of Diyala, Iraq. He received the B.Sc. degree in computer science from the University of Technology and M.Sc. degree from UOITC, IRAQ. He Holds a PhD degree in Computer Science with specialization in networking and Wireless communication. His research areas are image processing, AI, image analysis and pattern recognition. He has published several scientific papers in national, international conferences and journals. He can be contacted at email: hassan.hadi@uodiyala.edu.iq.

Israa Mishkhal is a lecturer at College of Science, University of Diyala, Iraq. She received the B.Sc. degree in computer science from the University of Diyala in Iraq and M.Sc. degree from Ball State University in United States America. Her research areas are AI, Machine Learning, Big data, image processing, and image analysis and pattern recognition. She has published several scientific papers in national, international conferences and journals. She can be contacted at email: israaadnan@uodiyala.edu.iq.

Dheyab Salman Ibrahim received the B.Sc. degree in computer science from the Mustansiriya University and M.Sc. degree from UOITC, IRAQ. Research areas are wireless networks, wireless sensor networks, and pattern recognition. He has published several scientific papers in national, and international conferences and journals. He can be contacted at email: alnedawyd69@gmail.com.