IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 8, 2020, accepted December 18, 2020, date of publication December 24, 2020, date of current version January 6, 2021.

Digital Object Identifier 10.1109/ACCESS.2020.3047210

Heuristic Resource Allocation Algorithm for Controller Placement in Multi-Control 5G Based on SDN/NFV Architecture

ABEER A. Z. IBRAHIM[1,2], (Member, IEEE), FAZIRULHISYAM HASHIM[1,2], (Member, IEEE), NOR K. NOORDIN[1,2], (Senior Member, IEEE), ADUWATI SALI[1,2], (Senior Member, IEEE), KEIVAN NAVAIE[3], (Senior Member, IEEE), AND SABER M. E. FADUL[4], (Member, IEEE)

[1]Department of Computer and Communication Systems Engineering, Faculty of Engineering, Universiti Putra Malaysia, Serdang 43400, Malaysia
[2]Research Centre of Excellence for Wireless and Photonics Network (WiPNET), Faculty of Engineering, Universiti Putra Malaysia, Serdang 43400, Malaysia
[3]School of Computing and Communications, Lancaster University, Lancaster LA1 4YW, U.K.
[4]Department of Electrical and Electronics Engineering, Faculty of Engineering, Universiti Putra Malaysia, Serdang 43400, Malaysia

Corresponding authors: Abeer A. Z. Ibrahim (abeerazibrahim@gmail.com) and Fazirulhisyam Hashim (fazirul@upm.edu.my)

This work was supported in part by the Organization for Women in Science for the Developing World (OWSD) and the Swedish International Agency (SIDA) under grant No. 3240291613 and also in part by the ongoing research collaboration between Universiti Putra Malaysia (UPM) and Lancaster University through the H2020-MSCA-RISE ATOM Project under Grant No. 690750.

ABSTRACT The integration of Software Defined Networking (SDN) and Network Function Virtualization (NFV) is considered to be an efficient solution that enables the forecasting of highly scalable, optimal performance of 5G networks by providing an effective means of network functionality. The distributed multi-controller architecture approach is an emerging strategy that primarily aims to support network functions performed through the application of a control plane, to provide versatile network traffic management. However, the management of resource allocations across multiple data centers is an important issue that still affects 5G core networks. Using such a strategy in 5G core networks requires the controllers to be correctly located, in order to improve network reliability and cost-effectiveness. Thus, to address the controller placement problem (CPP) in a distributed 5G network, we proposed an efficient, heuristic multi-objective optimization approach, using dynamic capacitated controller placement problem (DCCPP). It is based on the *K*-center problem, to solve the capacitated controller placement problem (CCPP), which acts as a resource location problem, in which the location and number of controllers can be allocated to maximize resources. A Greedy Randomized Search (GRS) algorithm was used to solve the dynamic assignment of nodes to controllers to achieve load balancing. The design of the heuristic method provides proper load balancing, efficient cost management, and network resource management, as compared to the basic CCPP model. The results indicate that the allocation and the optimum number of controllers under an effective decentralized policy could achieve a higher degree of efficiency through resource assignment in such a densified network.

INDEX TERMS 5G, SDN, controller placement problem, resource assignment, heuristic, optimization.

I. INTRODUCTION

The advent of the fifth generation (5G) has created an exponential increase in traffic volume, accompanied by the immense use of applications and various service characteristics, which have added to the complexity of network management and orchestration. This poses imminent challenges to all aspects of the 5G wireless network design [1].

The associate editor coordinating the review of this manuscript and approving it for publication was Maurice J. Khabbaz.

The Network Software and Virtualization concept embodies the Software Defined Networking (SDN) and Network Function Virtualization (NFV), which are enabling techniques aimed at solving and reconfiguring the complexity of the network, along with efficient resource sharing [2]. SDN provides operational intelligence by decoupling the network control functions from the data layer devices to support advanced automation for the 5G network management [3]. At the same time, NFV provides a useful abstraction of the functionalities of the network services. Moreover, it offers a new scalable

infrastructure to accommodate a wide range of network functions [4], [5].

The integration of NFV and SDN is run at a cloud-based architecture leveraging model for the 5G core network architecture (5G-CN) [6]. It facilitates the sharing of physical network resources and chaining functions between different sections of a virtual network. VNFs are designed to support all network traffic functions performed through a centralized control plane for 5G-CN management and orchestration requirements by providing a wide range of interconnected services [7].

The centralized server is meant to call upon the controllers for the network to be flexible enough to endorse the versatile policy configurations for managing the CNs and rapid deployment of new functionalities [8]. However, the deployment of network functions in a single controller has a drawback that is sufficiently linked to the optimum performance, that affects the guaranteed level of quality of service (QoS), necessary for large-scale 5G network management [9]. In this context, the adoption of multiple controllers for 5G networks in the control layer tackles flexibility, scalability and performance degradation [10]. In addition, unexpected network demands, and dynamic changes of topology create different load distributions amongst the controllers, as well as uneven coordination for controlling event problems. Therefore, maintaining a group of controllers needs careful attention to the allocation of resources and the control planes and management [11].

Nevertheless, the problem of network resource allocation in the distributed multi-control architecture of 5G based SDN is one of the most critical strategic challenges, requiring proper planning and optimization of both the control plane and the physical infrastructure layers. It targets proper resource allocation management of the control plane to guarantee adequate latency and bandwidth requirements, that provide a sufficient QoS, even if part of the system has a failure tolerance or load balancing capability issue [12], [13]. The solution is meant to find the best position for the number of controllers, as well as the assignment of switches to each controller, to achieve a reliable connection between the controllers and the network's physical components, to prevent any controllers from overloading. The mapping and placement of SDN controllers and switches are known as controller placement problem (CPP) [14]. It is important to assign user requests to different cluster controllers to balance the workload between them. When decreasing the overall response time for offloaded tasks, significant consideration should be given toward planning the required demand for traffic volume and service request control tools [15].

In this framework, several preliminary techniques have recently addressed the issues for CPP and dynamic switch assignment, as well as switch migration approaches. Each controller has a limited capacity in terms of handling the volume of traffic requests [16]. As traffic varies, the switches are dynamically planned and assigned to various controllers, as shown in [17] and [18]. Other researchers have tackled the issue based on load balancing [19], [20] and switch migration schemes [21]. Therefore, the aforementioned studies provided heuristics techniques for the CPP as a resource location problem, in which the metric number of switches focused on the impact of the optimal selection of the positioning of the controller locations and determined the weight of the switches based on the latency or distance. However, most of the previous works dealt with CPP in a single domain. They concentrated on load balancing without calculating the exact number of controllers against the network traffic load, as well as for evaluating the efficiency of assigning resource management.

The fundamental concept of this paper is to present the management and control framework for 5G-based SDN network architecture through efficient network planning and optimization for 5G-CNs. The methodology presented seeks to implement a framework as a solution for the CPP in the distributed architecture, to find trade-offs between the number of controllers, and determine the dynamic load assignment cost. In other words, the optimized model should maximize the performance of the network, manage the deployment costs incurred, and maintain the required load balancing between the network components, all while ensuring high network resource utilization.

The CPP is considered to be an NP-hard problem. Accordingly, heuristics is a method of tackling the optimization problem effectively. In particular, the applicability of a specific capacitated facility location (CFL) problem based on the K-center algorithm is investigated by developing a capacitated CPP (CCPP) model.

In this paper, the key contribution is to examine the location of dynamic traffic flow controllers based on the number of controllers and switch-to-controllers assignment for different average traffic situations within the network. The load metric was limited to the impact of optimal placement selection and guarantees that each partition was capable of shortening the maximum end-to-end latency. Therefore:

- The dynamic capacitated controller placement problem algorithm (DCCPP) is proposed to determine the allocation of controllers in the distributed control layers under dynamic traffic. Subsequently, a demonstration is shown how this layer can fulfill the specifications of the adaptive load balancing and the management of the resource in two applications.
- An optimal solution is also developed for both the location and the number of controllers using CCPP based on the generalization of K-center algorithm and Graph Theory.
- The resource scheduling efficiency is then investigated to measure the quality of the switch-to-controller assignment, which is handled by the controller. In the context of the switch assignment, the basic principles of the Greedy Randomized Search (GRS) algorithm are utilized based on the use of the neighborhood specified through the construction of demand points.

Thus, our proposed (DCCPP) attempts to answer the questions for the following sub-problems: 'How to measure the load of controllers and determine when to perform switch migration'; and 'How to define a trade-off between controller assignment costs, and the load balancing ratio.'

A detailed analysis of optimal network architecture is also provided, with a systematic assessment carried out across different topologies under various parameter settings, in a consistent manner. Our method is based on the reasoning above. In this work, we aim to ensure that the algorithm conforms to the 5G network requirements to ensure that the network variables are completely realized and that their values are optimally modified to achieve an exemplary network configuration.

This paper is organized into five sections. Following an introduction, the integration of SDN and NFV into the 5G-CN is investigated, and the resource management of the CPP is then explained in Section II. Related literature is presented in Section III. The proposed algorithm and the controller placement, and the allocation problem formulation are described in Section IV. The proposed solutions for dynamic capacitated controller placement problems are discussed in Section V. The model performance evaluation is provided in Section VI. Finally, a conclusion is given in Section VII.

II. INTEGRATION OF SDN AND NFV INTO 5G CORE NETWORK

This section presents a systematic redefinition of the 5G wireless network control/management functions used in the 5G-CN review. In particular, it describes the general design of the 5G core. In this work, we focus on a distributed model applied to a specified multi-control layer that demonstrates the efficiency of SDN and NFV technologies in 5G.

A. KEY ELEMENTS FOR 5G CORE NETWORKS ARCHITECTURE

The 5G network continues to advance toward reconfiguring the legacy network by enabling intelligence systems to be operated effectively across both access and core network based on SDN and NFV. It will also serve a wide range of networks, highly agile network controls, and cloud allocations to meet the needs of both network diversified traffic operations and big data demands [22], [23]. Based on SDN, the control plane functions are split from the forwarding capability of the physical layer elements (e.g., firewalls and routers) and reassigned to the centralized SDN controller. The control plane virtualization is configured by adding new network functions (NFs) to the software base instead of modifying each hardware switch [24].

Dynamic deployment of NFs can be implemented to achieve satisfactory separation of the resource slices. NF chaining can be implemented to provide flexibility in the 5G network infrastructure and allows virtualized services to be resourced in the 5G core cloud network, which can be replicated across different networks [25]. Figure 1 shows a layered architecture and generic SDN/NFV for 5G-CN.

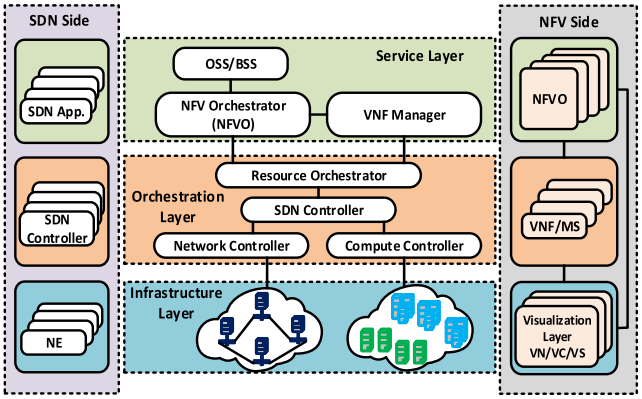

FIGURE 1. General integrated SDN/NFV layer architecture [6].

Therefore, the 5G-based SDN/NFV is projected to be a multi-tenant solution. Many autonomous network operators and service providers use the same physical structure and computing platform [26]. Consequently, there is a logical virtual layer solution between the control plane and the forward paths on the NE, which defines the path forward to navigate the virtual network [27].

B. MULTI-CONTROLLER ARCHITECTURE BASED ON 5G

The control plane's controller is a software program that runs on a high-speed virtual machine (VM) in the cloud or data center (DC) for real-time network implementation. It enables functions related to the control and management of 5G components, such as routers and switches, to be demonstrated hierarchically. It also handles the processes involved in the infrastructure planning, routing, and security applications through a set of interfaces with the VM network application [28].

A detailed multi-control architecture for the 5G-CN is presented in Figure 2. It has two main planes; the 5G control and management plane and the 5G physical infrastructure plane. Centralized control in such a 5G network with many

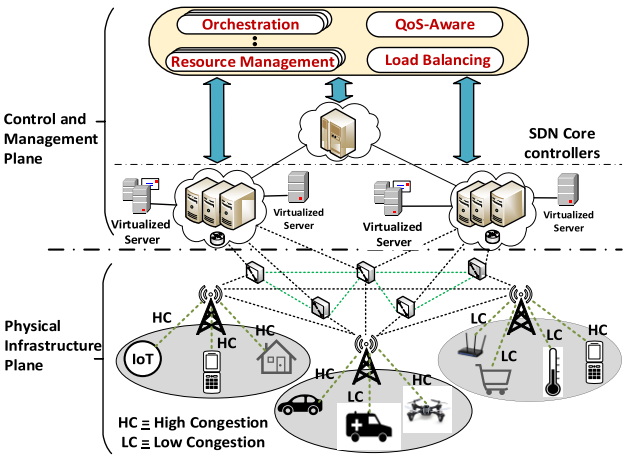

FIGURE 2. The multi-control architecture for 5G core network.

alternatives signaling paths in the congested network results in increased demand and a large amount of overhead [29]. This illustrates that using a centralized controller would adversely affect the entire network, rendering it insufficient for a single controller. However, the controller should manage all network traffic generated by the 5G network, but its limited resources would become a single point of failure [30].

Therefore, the deployment of a hierarchical multi-controller structure is seen as a potential solution that avoids the bottleneck, i.e., the controllers. It also maintains the demand for traffic loads to be distributed amongst multiple containers at the edge for traffic management of the growing capacity in order to achieve scalability. However, resource management remains an open issue in the 5G wireless network [31].

Our control layer of the model consists of two layers: a core control layer and cloud controllers. The core control layer technically acts as a centralized controller, where multi-controllers from various suppliers for multi-application configurations have been further improved. This improvement includes the high processing capacity of the controller clusters equipped with advanced multi-threading technologies and constructive traffic management systems [32]. In this regard, the deployment of the multi-layer architecture of a cloud orchestrator, or a distributed DC infrastructure, enables automated deployment and coordination of new systems, resources, and end-to-end services across edge networks [33].

On the other hand, the cloud controllers must have a clear vision of the entire network at each core. It is responsible for managing each local root controller in a centralized manner, effectively monitoring and scheduling the area's resources to accommodate different Internet functions.

The SDN local controllers in the control layer also distribute the flow control resource decision-making through each infrastructure subnetwork switch, pushing the computation storage to the network's edge [34]. They also provide NEs with rules that regulate their handling of packets to avoid a substantial amount of signaling generated between remote gateway applications and offload control over accessible network resources [35]. In addition, the edge server gathers data from nodes continuously. Depending on the specific case to optimize the 5G-CN, it sets out an optimal routing path according to the states of the cells. Such an edge controller paradigm for a 5G-CN controller has become a trend that brings computational services, infrastructure, and facilities closer to the terminal devices [36].

Indeed, coordination between the network components and their controllers is influenced by the structure of the distributed control plane, along with the number and location of multiple controllers within the SDN network, to satisfy the QoS constraint. This allows the management of connectivity services that produce some side effects in terms of the latency between the nodes, and their controllers, which is essential. It provides the ability to properly define the end-to-end (E2E) services [37].

Although several deployments of controllers solve the issue of capacity, they also increase the cost of utilization. Thus, the best way to reduce the overall network costs (CapEX /OpEX) is by minimizing the number of functions and the number of controllers. Such integrated converged resource management can provide an energy-aware and efficient allocation of resources means.

C. SCOPE OF 5G NETWORKS RESOURCE MANAGEMENT

The 5G-CNs must have an efficient resource management technique to optimize resource allocation and schedule users to support increased demand for network capacity and user experience. Taking advantage of the virtualization and programmability of the core network functions ensures the effective use of network resources for sharing, services amongst slices of the shared resource pool, and flexible 5G core scheduling [38].

The resource management decision process is distributed through network edge nodes that allow re-configuration based on the feedback of the network state. More concisely, this ensures the continuous aggregation of edges, and the allocation of bandwidth, computing, and storage capacity, to virtual connectivity elements. This leads to lower latencies and achieves the required reduction in operating costs. It also ensures improved network management for the user experience regarding the quality key approaches to reduce implementation and operating costs. The network storage resource is better used if the load is distributed more uniformly across the network (WAN or cloud), allowing traffic to be offloaded and preventing a bottleneck in the core network [39].

Besides, by integrating advanced routing protocols and traffic engineering algorithms, this strategy offers increased coverage for diversified 5G networks in lieu of load balancing and on-demand deployment with high service criteria for developing an efficient model. However, in a multi-controller architecture, different flow configuration models are feasible, as the controller-based core network has precisely a robust network vision. It is responsible for determining the rules for the maintenance and management of thousands of sub-layer DCs for providing optimized solutions for the overall productivity of high-capacity resources and functions [40].

III. RELATED WORKS

The integration of SDN and NFV is implemented using a combination of software on the IT servers and properly configuring the OpenFlow switches for the scalability of the control plane to minimize the cost of running network services. Therefore, researchers introduced NFV and VNF chaining in the SDN-based architecture and control plane frameworks and developed an efficient mapping of the virtual network resources [41].

Work in [28] was based on the same architecture for incorporating SDN and NFV into mobile devices. It implemented a dynamic deployment of the mobile gateway for load traffic balancing. It focused on addressing resource allocation through an orchestration controller, managing the virtual

function chaining and services, maintaining the performance, and preserving the network resources, maximizing controller reliability and load balancing capabilities in the dense WAN networks. However, the above works were considered to only focus on latency metrics for the shortest physical distance between the controllers and switches.

The implementation of controllers within a network control plane, single or multiple controllers, is based on the size of the network that to be deployed. At the same time, the cost-reduction framework for CPP is described in [42]. The deployment used specified latency as a metric to determine the best location and the number of controllers in the control plane for optimal network architecture design. Contrastingly, in [43], the mathematical derivation of the model was used to reduce the costs of the SDN network by considering different parameters and the interconnection of all network components. However, the abovementioned CPP research optimization literature ignored network traffic analysis, which is a vital element of the existing networks, in general.

The latency requirements of the switch-controller and inter-controller with the capacity of the controllers were taken into account to meet the traffic load management of the switches. Analyzing the SDN-WAN fault tolerance meant that the load balancing constraint amongst the controllers in the sub-domains in terms of their demand and availability was a key element in the distributed network. The load balancing played a crucial role in enhancing QoS and customer experience [3]. An assessment of the Pareto Optimum Set of the Controller Placement Optimization approach (POCO) paradigm for the controller's resilience often identified a trade-off between the controller's load balancing, and network traffic, as suggested in [44].

The SDN-based framework assigned switches to the controller when other service efficiency requirements were met. As an extension of this work is the dynamic controller location heuristic in the SDN [45]. The method was investigated using the K-medoid CPP and evaluated via Pareto's optimum solution. Similar studies in [20] suggested a CCPP based on the K-means and a matching algorithm for switch assignment. However, these methods focused on finding a trade-off based on the average latency and did not consider network resource flow traffic analysis.

The purpose of the dynamic configuration was to distribute the load amongst the controllers based on the traffic load generated by the controllers themselves [46]. It is advantageous to move a switch from a heavily loaded controller to a lightly loaded one. Thus, work in [47] introduced a dynamic slave controller assignment method in case of a network failure due to overloaded controllers. To avoid a network crash, it is essential to assign and migrate switches between the slave and a master controller for load balancing. The controller utilization is measured by the number of switches (flows) managed by the controller based on network traffic fluctuations and how they handle network traffic.

The proposed algorithms in [48] followed a clique-based approach, using the graph theory to identify high-quality solutions heuristically. It was then evaluated for actual WAN topologies, and the resultant effects were extensively examined for local network control over the entire partition and stability of the control plane. An elastically distributed controller architecture was proposed in [49]. Their solutions proposed to include mapping of resources between the switch and the controller, with the inclusion of a new controller for balancing the load. However, the authors only assumed delay in the emulation environment. The work in [50] developed a dynamic optimization algorithm for optimizing controllers and the best switch allocations for large-scale networks enabled by SDN. The developed algorithm was based on the Slap Swarm Optimization Algorithm (SSOA), which includes chaotic maps to optimize the performance of the algorithm. However, the optimization model was used to only optimized the minimum number of controllers without considering the analysis of traffic for load balancing.

IV. THE CONTROLLER PLACEMENT AND DYNAMIC ALLOCATION PROBLEM FORMULATION

Throughout this section, we introduce a general optimization framework for describing CPP in a distributed system compatible with the mapping of controllers for large-scale implementation of the SDN and NFV in the 5G-CN, as set out in Section (II) above.

The planning involved two types of decisions; (1) placing a controller at a location in subnetworks, installing and turning on the servers to maintain a local set of switches for various network forwarded traffic, (2) a dynamic controller management approach focused on the demand sites for each controller, depending on the flow assignment and capacity required by each controller. This focused primarily on latency, load balancing, and robustness. We considered the solution viable if the service level agreement was accepted. The optimum feasible solution tried to reduce the linear combination of total installation costs by a detailed analysis of the DCCPP in terms of allocation. This often-considered assignment algorithm and the reassignment switch algorithm.

A. NETWORK MODEL DESCRIPTION

A network representation of the problem is presented as an undirected graph $G(N, E)$. $N\{N \in (C_j, S_i)\}$ represents a set of network nodes and E represents a physical communication link between a group of network elements, respectively. The network nodes consist of a set of controllers (C_j) and switches (S_i). The controllers could be located at demand sites or separately across various candidate locations $j(j \in C_j, j = 1, 2, 3, \ldots, m)$ in a network. Besides, $S(i \in S, i = 1, 2, 3, \ldots, n)$ is a set of demand points that identify switches or other network physical resource elements. Simultaneously, d_{ij} denotes the shortest distance between cluster controllers and their nearest switches in the subnetworks. For example, from the perspective of the controller, it is more

able to automatically handle the allocated topology bringing it closer to switches, minimizing the control traffic overhead and latency.

The connectivity between switches and controllers is critical since it forms part of the frequent flow profile and requires little bandwidth resources. Notice that as controllers respond to the dynamic requests from nodes in their service area, all controllers can be characterized by the existing capacity constraints and the total resources that each controller can manage through migrating controllers and reassign switches are inevitable [17]. Table 1 summarizes the primary notations used in our model.

TABLE 1. The primary notations used in the model.

Notation	Description
E	Set of communication links in the network
N	Set of Network Nodes (controllers and switches)
C_j	Set of controllers or servers
S_i	Set of switches
C_K	Set of cluster controllers
i	The index represents a set of servers or demand points
j	The index represents a set of controllers
C_{UtiK}	Cost of locating the controller
C_{Assig}	Assignment cost
$f_r(d_i)$	Demand requested by the node i
$f_c(d_j)$	Demand accommodated at the controller
λ_i	Flow rate requested by switch i *(kbps)*
λ_{ij}	The flow cost between $(i.j)$
X_{ij}	The decision matrix of assigning demand to controllers
Y_j	The decision if controller located at j
d_{ij}	Distance between controller j and NE i
K	The integer represents the number of controllers
ς	Controller capacity *(kbps)*
$\eta_{resource}$	The efficiency of resource scheduling for control plane

In each feasible region, the decision recorded is to decide Y_j the position of controllers amongst the switches and X_{ij} assign switches to controllers. A subset of feasible location solutions was chosen for the core network, and a necessary minimum number of controllers used was specified. Then, the positioning strategies for all available switches were established. So, the efficiency can be maximized in such a way that the requirements of the QoS are still met. The distance between the nodes and their respective centers is an important design parameter for dealing with center allocation problems. Therefore, for the purpose of load balancing and reliability, each controller should maintain the same workload for the input services, the upper bound for the number of centers, and the total load for the output of the K sets. Given the above definitions of the network model description, the DCCPP problem can be formulated as follows

$$\text{Minimize } (C_{UtiK} + C_{Assig}) \tag{1}$$

The optimization model reflects two objective functions, the cost-utility function and the assignment cost function, to minimize the overall network resource cost. The first term of the objective function (1) is the cost-utility function C_{UtiK}. It reflects the overall fixed cost of locating controllers by covering all demand request nodes. The second term is the assignment cost C_{Assig} represents the weighted flow traffic generated at the switch and the delay from the switch to the controller.

B. CONTROLLER LOCATION COST

We considered the controller utilization cost as an operating cost toward the maintenance and communication costs as an initial assumption. The setup of the infrastructure and maintenance costs of the controllers were almost static for the different network service providers. The number of servers was related to the number of computational resources, i.e., CPU cores, needed for the NFV functions chains, i.e., virtual gateways, or SDN function chains, i.e., controllers [7]. Since our main focus was the network design, the cost of locating a minimum number of controllers is defined as:

$$\text{Minimize } C_{UtzK} = \sum_{j \in m} Y_j \tag{2}$$

$$\text{Subject to : } \sum_{j,i \in N} Y_j \geq 1, \quad \forall j, i \in N, \tag{3}$$

$$\sum_{j \in J} Y_j = K, \tag{4}$$

$$X_{ij} \geq y_j, \quad i \in S_i, \quad j \in C_j, \tag{5}$$

$$\sum_{i \in n} X_{ij} = 1, \quad \forall i \in n, \tag{6}$$

$$X_{ij} - Y_j \leq 0 \quad j \in K, \quad i = 1, 2, 3, \ldots, n, \tag{7}$$

$$Y_j, X_{ij} = 0, 1 \quad j \in K, \quad i = 1, 2, 3, \ldots, n, \tag{8}$$

Equation (3) assured that there is only one controller for each control domain. Constraint (4) divided the network into the $K(K \subset j)$ service regions. Note that K is the minimum number of controllers required for each demand sites within the minimum delay grantee. While constraint (5) guarantees a given NE, which is assigned to the nearest active controller or server itself. Constraints (6) and (7) were the assignment, or demand satisfaction constraints, which ensured that each NE was assigned precisely to exactly one controller, j, and the requested demand was satisfied by the located controller in the cluster. This confirmed that each switch was attached to the established controller within the distance limits. Finally, equation (8) defines the auxiliary decision variables as binary.

The controllers would then be in a good location to appropriately adapt their resource utilization to provide the bandwidth required and the optimum response time for the forwarding devices.

C. END TO END NETWORK DELAY

The latency between the controllers and their associated switches is the most crucial design factor for some delay-sensitive applications and determines the efficiency of the 5G network. The end-to-end network delay consists of three components: Packet sending delay (τ_i), propagation delay (τ_{ij}), and processing latency (γ_j). According to [49], the processing latency for NFV and SDN switches is minimal (microseconds), compared to the network propagation latency of a widespread core network topology, which is in the order of milliseconds.

The calculation of the propagation delay between the controllers and their NE on a graph depends on the location chosen to achieve a minimum distance between the demand to be assigned and the controllers on a given network. The travelling flow latency cost is defined as the minimum delay between the controllers and switches path to elaborate on the shortest path. The algorithm examined the weighted distance from the node to the near center in such a way that satisfied the following condition, as in:

$$\text{Minimize } D_{NW} = (Max(d_{ij})) \tag{9}$$

$$\text{Subjected to : } \tau_{NW} \leq T_{threshold}\ \tau_{NW} \geq 0, \quad j \in C_K \tag{10}$$

Equation (9) shows that the essence of reducing latency was an important principle through which we can accurately identify the required number of controllers by monitoring the minimum latency $T_{threshold}$ in order to minimize costs. Constraint (10) focuses on the network delay bound constraint, which declares the types of decision variables and their restrictions. Higher latency typically refers to switches failing to implement flow rules in time, influencing network reliability and availability.

D. DYNAMIC ASSIGNMENT COST

The controller is responsible for scheduling interconnecting services throughout the distributed network. Therefore, assigning the switch to the core controllers requires bandwidth and delay measurement before the average workload is exceeded. The SDN controller handles and controls several request-demand messages (*PACKET IN*) from switches. The controller then calculates the flow routing path to initiate a traffic flow exchange. If the requested controller accepts the request, it returns either the approved acceptance *ACK* or the denied *ACK*. Incoming traffic can be handled dynamically with a flow table structure.

On the other hand, the SDN controller informs the nodes how to distribute the flow through the (*PACKET OUT*) message. Subsequently, the controller load consists of the (*PACKET IN*) and (*PACKET OUT*) arrival flows. As we consider all the control domains that the flow passes through, traffic thus consists of all concerned requests for flow setup and reply-responses. However, in the selected path, the capacity of usable bandwidth is shared by multiple switches and is assigned by the controller. Overloading the controller leads to increase response delay of network events and failure of the controller [51]. However, the total delay from the source to the destination depends on the path diversity. The performance metrics, such as response time, is calculated based on the elapsed time between the transmitted data packets and the successful receipt of the data packets themselves.

The workload in a service region depends on both the average arrival rate of flow (λ_i) that is requested for by each switch and the travel time (τ_{ij}) between the nodes i and j, respectively. We modelled the switch weight $\lambda_{ij}(t)d_{ij}$ as the number of new flows per unit of distance, as in:

$$\lambda_{ij}(t) = \sum_{i \in n} \lambda_i(t) \tag{11}$$

Then, the total amount of average arriving flows requested by the switches and accommodate at the controller is:

$$f_c(d_j) = \sum_{i \in n} \lambda_{ij}(t) f_r(d_i), \tag{12}$$

To this end, the controller load is given by:

$$L_C(C_j) = \sum_{i,j \in N} f_c(d)_j \tag{13}$$

The controller load is defined as capacity to handle the number of flow events from all current switches to their domain controllers. The higher the number of flows (or switches) that the active controller manages, the more it is utilized to achieve the best cost to obtain an equal and balanced resource distribution between the controllers under the constraint capacity.

Thus, it is crucial for such bottleneck problems to balance the load using a dynamic load balancing approach that is proposed for clustered controllers based on the idea of a switch reassignment mechanism. The approach helps in the mitigation of unnecessary signaling and resource utilization. Therefore, each partition load is adjusted, and splitting occurs across the number of switches in the controller management domain and the size of the flow to be processed by each switch. As a result, neighboring domain controllers in the sub-networks try to create a destination to change the resource itself and cooperate in a distributed manner to decide the available resources providing the required bandwidth for the allocation process feedback on the network's state [4].

The aim is to decrease the resource cost by assigning the total number of servers available (e.g., in the DCs deployed). The sum of the switches' weight cost should therefore be reduced at a single DC location. Note that since the control plane requires more resources (or memory) to adapt to dynamic flow profiles, the system needs to be reconfigured.

The load-balancing algorithm iteratively adjusts the traffic flow splitting ratios so that traffic can be diverted from the maximum utilization link in the network. Similarly, in the multi-control 5G-CN, the used and available data flow resources are not local concepts but are linked to the adjacent nodes at the end of the path. Therefore, to extend the CPP problem, the dynamic assignment traffic allocation problem

is formulated as a linear programming model. Both the number of required servers and the resource assignment cost C_{Assig} are minimized according to the following equation:

$$\text{Minimize } C_{Assig} = \sum_{i=1}^{n}\sum_{j=1}^{m}\lambda_{ij}(t)d_{ij}X_{ij} + \sum_{J\in K} f_c(d_j)Y_{ijK} \tag{14}$$

Equation (14) is subjected to the previous constraints (4), (5), (6), and (7). All the switches must be allocated and controlled by their fixed controllers in a subdomain $Y_K = 1$, $\forall K \in C$. They have to satisfy the demand request send to the controller, as follows:

$$\sum_{j\in J}\lambda_i X_{ij} = f_r(d_i) \quad for\ i = I\ and\ j = J, \tag{15}$$

Otherwise, constraint (15) follows the flow balance constraint, which enforces a total load of each located controller to satisfy its desired switch requests.

$$\sum_{j\in K} f_r(d_i)X_{ij} \geq f_c(d_j), \tag{16}$$

Both constraints (15) and (16) are related to the flow variables and the assignment decision, which belongs to the same controller. The requested flow by a switch can be set by the controller j, only if it is available on the controller, j. The constraints forbid the delivery of *PACKET_IN* from the controller to the switch if the controller is overloaded. Through the model, constraint (17) is a capacity constraint for controller capability as described in:

$$\sum_{i\in N} f_r(d_i)X_{ij} \leq \varsigma_j Y_j. \tag{17}$$

Ensuring the number of switch demands X_{ijK} is assigned to a given controller, which does not exceed its maximum upper-bounded nominal capacity ς_j.

Solving the objective function in (14) results in balancing the load according to the amount of traffic to migrate between the network devices while minimizing the cost amongst all paths in the network. It also specifies the incoming traffic flow correlated with the demand, of which it originates from the switch and enters the node (λ_{ij}) where the controller is situated. It must be equal to the demand fraction served by that controller, as well as the outgoing traffic flow leaving that node since this demand is optimally assigned to another controller. The pseudo-code for Algorithm 1 is given in Table 2. The placement algorithm is summarized in Algorithm 1.

For each problem mentioned above, we attempted to construct a subgraph $G(N, E, K)$ == $(\bigcup_{j=1}^{K} N_j, \bigcup_{j=1}^{K} E_j)$ corresponding to the given completed edge-weighted graph structure, such that the length of the longest edge constraint by (5) and (6) and the maximum path capacity of an edge $E(i, j) \in (2\varsigma_j \leq f_r(d_i))$ was minimized. Accordingly, in each feasible region, the dominated set switch $S_{ijK} \subset S_{i\in n}$ was to reside in the nearest available controller $C_j \subset C_k$. The edges' weights corresponded to the shortest path distances and satisfied the $d_{ij} \leq d_{ijk}$ values of the decision binary variables $X_{ij} \in \{0, 1\}$. It is necessary to ensure that, at $\sum_{j\in m}\varsigma_{jk} \geq \sum_{J\in K} f_c(d_j)$ the location of the controllers provided a feasible solution by having an adequate cumulative capacity to meet all of the necessary demands and certain boundary criteria that needed to be specified. Firstly, the transfer should not impose the capacity constraint of the controller, and secondly, the minimum number of controllers that met the above constraints need to be known. The optimal solution for the optimization model can therefore be set to the minimum cost-network load ($f_r(d_i)$) with a lower bound, the minimum quantity of ς_j, and the demands ($f_c(d_j)$) of the network flow problem shown. Moreover, as the dynamic traffic changes, the value of the K controllers also changes dynamically. Then it is mapped to the SDN switches for the maximum number of controllers

TABLE 2. The Pseudo-code for DCCPP *K*-center Algorithm.

Algorithm 1:	Dynamic Capacitated Controller Placement Algorithm (DCCPP)
Input	Graph (N, E), ς_j , K , $f_i(d_i), \lambda_i$
1:	**Initialize** Construct G_K
2:	**for** each $j \in m$ **do**
3:	Generate clusters of regions $K(C_1, C_2, C_3 ... C_K)$
4:	**if** $\max\_N < N$ && $Map\_flag==1$
5:	$deploy \leftarrow nodes.\min(d_{ij})$
6:	**else** $get\_dis\tan ce(d_{ij}), calaculate\_delay$
7:	$Update\ centers$
8:	**end**
9:	**end**
	Scheduling and assign nodes to near centers according to flow to construct traffic matrix in (13).
10:	**Start**
11:	**for** $i = 1 : Itre$ **do**
	For each demand node in the given region request demand, arrival times
12:	$Nodes[(r(d_i)] = Nodes[\min f_r(d_i), \max f_r(d_i)]$
	Test the constraint (15)
13:	$blocked\_data \leftarrow (i, j), [K]$
14:	$blocked\_data(after\_rescheduling) = blocked\_data$
15:	Calculate assignment cost
	for $K = 1 : m$ **do**
16:	$Nodes(C_K) \leftarrow size(i, j)$
17:	$calculate\_capacity\ \varsigma_j$
18:	$locate\_K\_controllers \leftarrow K(i, j)$
19:	**end**
20:	**if** $sum(block\_data) = 0$
21:	$no\_blocked \leftarrow no\_need\_for\_rescheduling$
22:	**else**
23:	$blocked\_data\_after\_scheduling$
24:	**end**
25:	$calculate\_efficiency$
26:	**end**
output	$best\_number\_of\ controllers$, $blocked\_data$, $\eta_{resourceAssign}$

in terms of the upper bounds. Both methods can use the average delay of the decision-levels (location, allocation, and capacity). Based on the constraints (16) and (17), we can see that such a property is still held for, which establishes the dynamic assignment of switches. In the following, we provided some definitions for the algorithm design.

1) THE LOAD BALANCES RATIO

The main load of the controller comes from the processing of the flow requests by the switches. In order to sustain uniformly distributed traffic across all switches, any controller on a subnetwork must compute resources while the switches generate requests and maintain their load. It covers the traffic volumes provided by the controller. Before reassigning the switch, we need to identify the threshold that uses the controller load to determine the network factor or assignment factor. The threshold value β can be expressed as the difference between the load value of the controller and its capacity is:

$$\beta = \frac{1}{\varsigma_j(C_K)}\left(\sum_{j\in K}(\varsigma_j(C_K) - L_c(C_j)_K)\right) \tag{18}$$

This threshold reduces the disparity between the number of switches assigned to the maximum load controller and the minimum load controller. The framework then sends invitations based on the threshold set out in (18). If the demand requested by the switch exceeds the load threshold of the controller, it will be overloaded, and the overall demand will be blocked. On this basis, the blocking meets the criteria set out in (19), and the blocking is then specified in accordance with the following conditions:

$$\left\{\begin{array}{l} \varsigma_j(C_K) > \beta, \ underprovision \\ \varsigma_j(C_K) < \beta \ , overloaded \\ \varsigma_j(C_K) = \beta \ , balanced \end{array}\right\} \tag{19}$$

The controller uses the blocking amount to reassign the necessary demand nodes to another controller on the network. Rerouting traffic from heavily loaded controllers to middle-boxes is primarily responsible for analyzing and tracking, or dropping, malicious traffic. This process ensures the reliability and resilience of the network. We also described the resource blocking ratio as one of the techniques assessed to maintain a lack of connection, or link interruption, due to the controller capacity overload. The general working concept of the load calculation module was discussed in Algorithm 1.

2) THE RESOURCE SCHEDULING EFFICIENCY

Obviously, each controller can only serve a limited number of switches. The network resource is best used if the load is more equitably distributed and balanced across the network. The goal of the switch assignment cost strategy is to assign certain switches between network controllers until the controllers exceeded their load based on higher assignment performance. Therefore, the resulting topology is analyzed in terms of the efficiency of the resource scheduling at the controller to measure the performance of the number of active controllers demands requested by the switches, concerning the capacity of the cluster's controller.

$$\eta_{\mathrm{Re}\, sorceAssig} = \frac{\sum\limits_{i,j\in N} f_c(d_j)}{\sum\limits_{j\in K}\varsigma_j(C_K)} \tag{20}$$

In view of the role of the controllers, the idea here is to develop solutions in a variety of ways in terms of resource performance. Certain switches may have a higher preference depending on their distance when assigned to the controller. Among the resulting availability and blocking ration, Algorithms 2 and 3 have optimum assignment costs by minimizing variable server costs. The pseudo-code of Algorithms 2 and 3 for scheduling and rescheduling are shown in Tables 3 and 4, respectively.

TABLE 3. The pseudo-code for resource scheduling.

Algorithm 2	Scheduling Algorithm (Assignment)
Input	$\lambda_{ij_i}, \varsigma_j(C_K), C_i, r(d_i)$
1:	**Start** *scheduling*
	while in distance traffic Matrix
	Such that $X_{ijK} = 1$
2:	**for** $i = 1:n$ **do**
	$X_{ijK} \leftarrow \lambda_{ij}$
3:	$f_c(d_j) = f_c(d_j)_K + \lambda_i(i)$
	Select all demand assigned **do** scheduling
	Compute load balancing
4:	**if** $L_c(C_J) \geq \varsigma_j$ **then**
5:	For each controller $f_c(d_j)_K = f_c(d_j)$
6:	$active\_controller = C_K$
7:	**else** $blocked\_data \leftarrow blocked\_data + \lambda_i$
8:	**end**
	end if;
9:	**end while;**
	$i \leftarrow K;$
10:	**Return** *scheduling ended*
Output	$Balance\_ratio, active\_data, C_K$

V. THE PROPOSED SOLUTION APPROACH FOR DYNAMIC CAPACITATED CONTROLLER PLACEMENT PROBLEM

This section presents an approximate solution for the DCCPP in the 5G-CN based SDN and NFV, which is analytically formulated as a heuristic NP-hard problem [44]. Our strategy for the proposed Algorithm 1 requires utilizing the center in each subnetwork of the desired clusters $\{(C_1, C_2, \ldots, C_K), C_j, j \in C_K\}$. Concurrently, assigning the switches to each center on the minimum accepted delay level ensures scalability and load balancing based on the available objective functions. In this context, such a solution is referred to as capacitated K-center problems.

TABLE 4. The pseudo-code for resource re-scheduling.

Algorithm 3	Rescheduling Algorithm (Reassignment)
Input	$active\_data\ f_c(d_j)$, $blocked\_data$, d_{ij}, ς_j, K
1:	**Start** *scheduling*
2:	**for** $i = 1 : blocked\_data$ **do**
3:	$f_c(d_j)(i) = blocked\_data(i)$
4:	**if** $f_c(d_j) > 0$ **then**
5:	**for** $j = 1 : m$
6:	$remain\_data(C_K) = \varsigma_{jk} - f_c(d_j)_K$
7:	**if** $f_c(d_j) < remain\_data(C_K)$
8:	$blocked\_data(i) = 0$
9:	$f_c(d_j)_K \leftarrow f_r(d_i) + d_j$
10:	$C_K \leftarrow K$
11:	**end**
12:	**end**
13:	**If** $blocked\_data(i) > 0$
14:	**end**
15:	**end**
16:	display the full rescheduling
Output	*Nodes number,* C_K

Given the location of the controllers in Algorithm 1, there are essentially three levels of decisions for a feasible region; the number of controllers, the assigned demand nodes and the capacity to deliver optimum location costs, which is achieved by reducing the number of variable costs of the nodes. Though, all capacity and demand constraints have been relaxed due to the assignment problems in the first and second cases of the DCCPP.

Consequently, once the controllers are configured or bipartite domains are established, the forwarding device may be optimally assigned to the network. The algorithm calculates the weighted distance between each point and the center of the cluster by choosing the controller to become the centroid for each cluster. The deployment between switch and controller $[\lambda_{ij} d_{ij}]$ forms a traffic matrix. These switches are allocated to the nearest controller based on the used standard minimum cost delay function $X_{ij} \in \{d(C_j, S_i)\}$ and the flow request technique. The shortest path distance is then calculated based on the coordinates and the adjacent matrix. The node with the highest end-to-end latency to the centroid is selected as the second initial center until all the centroids are located together. The centroids are then recalculated as a means of all points assigned to them with an initial partition based on a modified capacitated K-center algorithm. After a number of iterations, the algorithm repeats from solving the above model described by the constraints (3)–(8) across all centers of the network C_K, $(j = 1, 2, 3, \ldots, K)$, which is constructed as making sure no point is left out. At this point, the first two levels of decisions $\{Y_j\}$, and $\{X_{ij}\}$ which have been initially obtained, are then used to construct the service regions. Such clustering considers only the location of the nodes assigned to each controller and can thus deliver arbitrarily low imbalanced results. The final possible locations are those with a minimum sum of costs.

There is no a priori information on the number of controllers to be located in the CCPP. Our algorithm began with one controller and implemented an algorithm that raised the number of controllers by one unit for each iteration sequence. The algorithm stopped when there were no changes to the objective function by picking many controllers. This procedure evaluated the location selection process, in which each iteration of the solution minimized the sum of the fixed location of node costs.

The assignment cost of the switches to the controllers was determined by the weight of the link or the shortest distance to the controller.

The closed pseudo-computing code for assigning or scheduling switches to the controllers is seen for Algorithm 2 in Table 3. In this case, the GRS technique was used to solve a switch assignment problem. This heuristic approach is an iterative process, in which each iteration consisted of two phases; construction and a local search. The construction phase offered a viable solution to the objective of the respective traffic forwarding problem. The algorithm related to the interconnected path between the nodes in the network determined the controller location's neighborhood structure. This included the traffic matrix. Generally, the neighborhood focused on the demand point (switches). The nearest controller represented the controller in the clusters and switches were assigned to the controller when considering all delays and loads.

Scheduling began with the input of the original traffic assignment matrix in Algorithm 2. The loop from line 2 to line 11 assigned a positive input of the actual traffic matrix to one of the controller mode matrices, C_K, in the decomposition. The controllers in line (4) then examined the capacity constraints related to the control and satisfaction links. To ensure that the controller capacity was not exceeded, each controller tested the threshold level β for normal operation levels. However, if there was a variation in the controller's capacity, it would be overloaded and subjected to extra data blocking. Otherwise, a new switching mode matrix is initialized in line (7) of Algorithm 2 to accommodate all unassigned entries and conflicts associated with the drop. The neighboring area under generation is then completed in lines 13–15 of Algorithm 1 by the blocking matrix. The algorithm repeats the search process for all the nodes in the request list, as the solution varies. If the solution is improved (lower bound), the algorithm preserves the current assignment or restores the feasible assignment to retain the optimum cost.

The integrated nature of the problem does not mean that such sub-problems (position and assignment) cannot be isolated, where the location of the controller reflects decision-making at the strategic level, and the switch assignment stage provides flexibility at the operational stage. Using a decision-supporting multi-objective optimization

workflow provides scope for flexible management decision-making.

At this stage, the rescheduling of the blocked nodes is carried out and oriented via the assigning principle. A negotiation process is initiated between the controller and the assigned switch, based on the traffic volume and the minimum distance specified beforehand. Secondly, when the controller gets *ACK* messages from its neighbor, it decides on one of its migration switches and sends a notification of migration to its neighbor. The next cluster controller selection often depends on estimating the shortest distance and load balancing ratio in (18) to ensure the latency threshold and the controller bandwidth limits as desired.

The rescheduling process can be carried out by modifying the roles of the controllers in order to set the traffic flowing along with the new $d_{ijK} \in L_C$ from the demand point to the next controller. However, the next controller is determined by the minimum distance and the corresponding nominal arc capacity $\sum_{J \in K} f_c(d_j)_K Y_{ijK} \leq \beta$. Assuming that all controllers have an equivalent capacity (ς_j) to serve whatever is assigned to them. Therefore, for load balancing and reliability to be handled (so-called capacitated constraints), the assignment of each request to its nearest located controllers can be gradually optimized until the best possible assignment is achieved. In this case, if the demand surpasses the capacity, it means that the controller has reached the maximum processing capacity and is overloaded. To this end, a few nodes will be blocked (dropped) or sent to the nearest cluster, or redundant controllers will depend on the layout of the distributed control plane.

The accuracy of the results can be further enhanced by executing the algorithm code several times for each iteration. Besides that, in both cases, we assumed the balanced version of these problems. A potential way to solve this situation, where $\sum_{i \in I} f_r(d_i) \geq \sum_{j \in K} \varsigma_j$ is to bound the maximum number of nodes, is assigned to a single controller by constraints 17 and 19.

Hence, Algorithm 1 aims to determine the number and location, where a cost function depends on the vertices. For Algorithm 2, the objective is to pick a set of centers where the total cost is at most K, such that the distance is minimized. If the solution improves (lower objective), it keeps the new location. Otherwise, it restores the initial location. Then, it repeats the procedure for all demand points to be located. Algorithm 3 is responsible for rescheduling the overloaded switches to neighboring controllers.

Following a procedure that has been generalized for all graphs, G, the solution with a minimum center is the final solution for a balanced K-center problem. In all instances of the algorithms, we considered the optimal versions of these problems.

VI. MODEL PERFORMANCE EVALUATION

In this section, the network reflected the actual physical properties of a communication link to calculate parameters that evaluated its performance. Therefore, the findings focused on four evaluated criteria; delay cost, network load cost, number of controllers, and network size.

A. SIMULATION MODEL

In the simulation, we considered the random traffic and location scenarios for different networks consisting of several SDN nodes randomly deployed in a 2000 km $\times$ 2000 km square area as in [51]. The location of the nodes followed a uniform random distribution in the simulation region. A node can request demand from 0 up to 100 kbps/req within the network under service region. The processing capacity of the controller is set to 1800 k flows/s as adapted from [20]. We considered that all controllers had the same functionalities. Besides, we identified the minimum and maximum controller capacity required to satisfy all specifications. The controllers can be located at any of these node degree locations, depending on the calculation of the node values concerning the number of node neighbors.

The empirical results of the model conducted over the various WAN topologies architecture are presented. The link weights are set to become the propagation delay tool, which measured the distance between the controllers and the switches. The proposed algorithm was then implemented and tested in MATLAB using an Intel Core i7/Gen 10 processor and 12GB of RAM.

B. ASSIGNMENT COST AND DELAY

Testing was carried out in a simulated environment with different topologies, across 25, 34, 42, 54, 61, 100 and 150 nodes. The number of K controllers ranged from 1 to 8. The focus was on quantifying the trade-off between the load balancing rate using the switch assignment method and the task cost. Then, we evaluated the switch assignment cost for each process.

Simulations were conducted to assess the behavior of the networks when the number of controllers varied. We evaluated our proposed algorithm's performance regarding the adoption of total assignment costs and end-to-end latency. It is compared to the previous proposed CCPP methods, K-mean algorithm and *Kuhn-Munkres* (K-M) for capacity matching strategy in [43], and the capacitated K-median based on the minimum-cost flow algorithm in [52].

To meet this purpose, we first calculated the total cost location value using the normal capacitated location based on the K-center method to obtain an optimum number of controllers. The problem of the trained positioning of the controller was considered to reduce the delay of the control paths and the load of the controllers. Other techniques, however, measured the switch migration and minimized load balancing.

Figure 3 displays the assignment costs of the three methods. Our method DCCPP and the two other methods, 2 and 3, as well as the number of controllers used over various network sizes. As shown, the assignment costs increased with the number of controllers used. Our approach showed that an optimal number of controllers and a range of network sizes

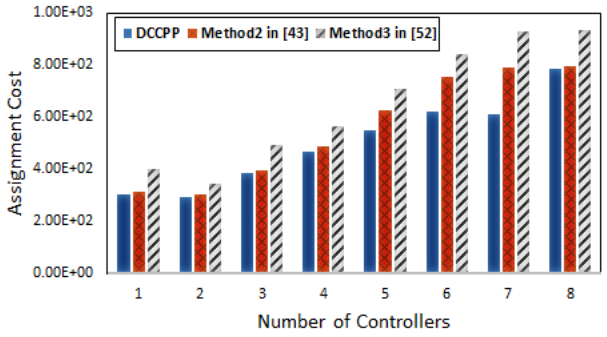

FIGURE 3. Assignment cost for different three methods.

could be achieved at a low cost. The process of locating the central controllers in the DCCPP was performed directly from the first step. However, the other two methods used to find the controller depend on the search steps.

In comparison, our approach selected the number of controllers with minimal iterations, compared to other methods that picked the number of clusters after several iterations. Many iterations meant more burden toward the controller output, which, at the same time, leads to a computationally expensive system. To sum up, in our method, the correct number of controllers can be determined to prove that our assignment algorithm was effective and accurate, compared to other CCPP assignment algorithms.

The average delay and number of controllers were quantified over various network sizes. Simultaneously, the proposed algorithms performance improvements were evaluated under a realistic network model, as shown in Figure 4. Our method, the DCCPP, could achieve a lower delay than the other two methods. As the K value increased, the average delay of three approaches decreased. However, the *K-center* locates the nodes resulting in producing good clustering from the first steps of the algorithm, while, the *K-median* searches for good clustering resulting in consuming time to locate the nodes. In addition, to balance the number of switches assigned to each controller, certain switches were assigned to another controller.

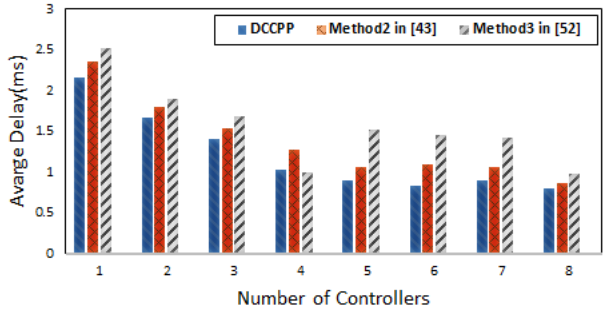

FIGURE 4. The trade-off between the network average delay and number of controllers over different networks for different three methods.

Figure 5 shows the trade-off and dominant effects of the average network delay and the average number of

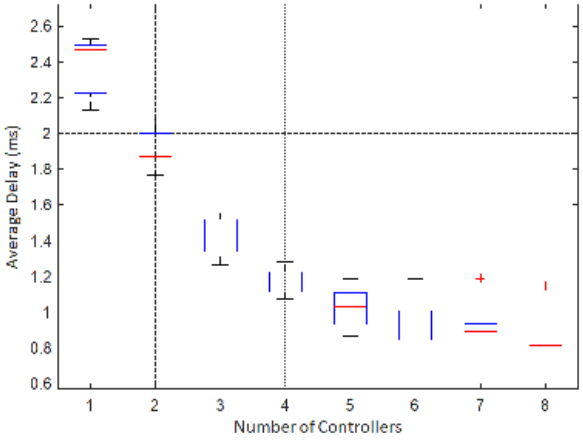

FIGURE 5. The average network delay and number of controllers over different sizes of the network.

controllers and their locations for the same group of topologies. Each boxplot corresponded to a set of minimum and maximum delays (short distances) for each network, which were grouped according to their sizes. For example, the selection of the optimal controllers was calculated at a minimum value for the average delay.

As the number of controllers increased, the delay reduced since the nodes were allocated to a minimum distance. In other words, all controllers were assigned during the minimum average delay. Although the delays with fewer controller numbers were high, the nodes were located far off from the controllers.

Therefore, we presented the results for these different topologies to investigate the impact of the deployment of the controllers across a variety of controllers (the preselection K for 1 to 8 controllers). The results focus on four evaluated criteria the delay cost, network load cost, controller center resources cost, controller numbers, and the size of the ranging network.

Firstly, the algorithm located points based on the distance. After allocating controllers in clusters and assigning the switches to their nearest controllers based on the capacity, the switches started to request data from controllers within the cluster. Therefore, if the number of switches associated with a specific controller capacity was met, there was no need for rescheduling. It noted that the trend analysis to determine the optimum resource balance between controllers and network nodes must be performed in various situations, as the optimum ratio can vary for each particular situation.

C. CONTROLLER LOAD BALANCING

In the DCCPP, the selection of the optimum controllers was determined at the minimum value of the average delay and the capacity. This was done in consideration of the management and deployment costs of the controllers.

Figure 6 illustrates the relationship between the number of controllers before rescheduling and load balancing ratio (β). The ratio β indicates the number of blocked data that reflects

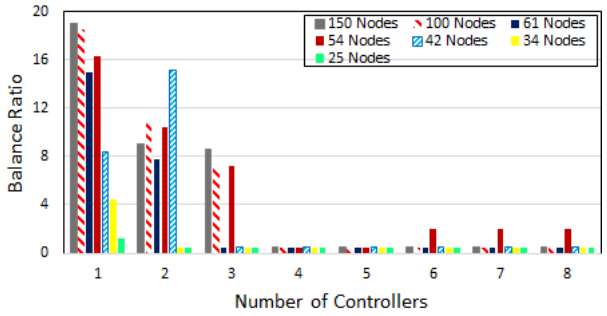

FIGURE 6. The balance ratio and number of controllers over different network sizes.

the difference between the load of the controller and its capacity. Blocking data gradually declined to a balanced level (near zero), with the increase in the number of controllers. The controller's resources, including storage and limited capacity to schedule switches, could accommodate and maintain a limited number of OpenFlow switches.

During an overload decision, some nodes may be blocked. The blocking rate gradually declines with an increase in the controllers. In the case of the topologies mentioned above, load balancing is desirable to provide a better load distribution between the controllers. However, the scheduling efficiency depends on the blocking ratio (balance index), which decreases with an increasing number of controllers over different network sizes.

For the different number of controllers, the balance factor is very different. For example, at $K = 1$ and $K = 2$, the higher balance factor shows a higher blocking rate of data. But, when $K > 3$, all cases' balance factor is near zero, which means that there is no overload at the controllers. The blocking data then declines gradually to become zero, with an increase in the number of controllers. At this point, the load is well balanced amongst controllers.

The resulting configuration of the topology was also evaluated in terms of resource performance (rescheduling efficiency) for overload cases. The performance was defined as the amount of demand requested by the switches for the capacity of the cluster controller. Figure 7 describes the relationship between many controllers, the performance of the

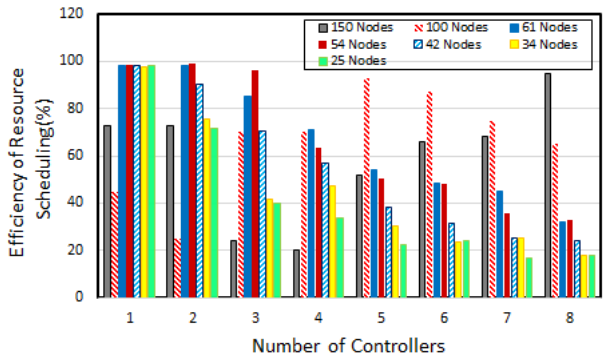

FIGURE 7. Impact of the resource performance over different topologies and number of controllers before rescheduling of resources.

assigning switches, and the blocking ratio under a balanced factor. In this work, we have defined the efficiency of the scheduling ratio, from (20), to be between 10% and 100% of the data traffic demand over several iterations.

D. SELECTION OF OPTIMUM NUMBER OF CONTROLLERS

The optimal number of controllers was selected for optimum efficiency. However, the performance varied based on the distance between the nodes. Some nodes were very far off from the controller, resulting in poor resource scheduling, contributing to low performance.

Consider a scenario with a number of 54 nodes (medium size network), where, typically, the selected control number is picked with a higher efficiency before the rescheduling is approximately set at 80%. This optimum number of controllers maximizes performance and minimizes network load costs, as discussed in Figure (2). This observation extends to different sizes of topologies, ranging from small (25, 42, 54) to wide (61, 100, 150) nodes and all the number of controllers used. In our methodology, the added rescheduling of resources would compromise for more reliability and balancing of resources. For Algorithm 3 within rescheduling, each node can search for the next server on the line, based on the shortest distance and the available capacity.

Figure 8 shows the relationship between the number of controllers and resource scheduling efficiency. The selection of controllers for each size of the network was decided with optimal resource efficiency. For example, for a network with a size of 100 nodes with an efficiency of 92%, the corresponding number of controllers is 5. However, there are 3 controllers for 54 nodes, 1, 2 and 3 controllers for a 95% performance level.

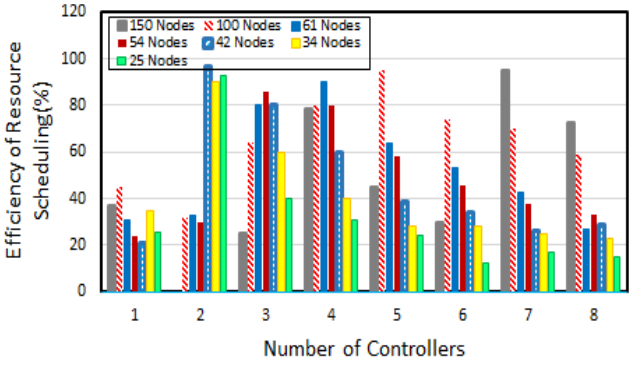

FIGURE 8. The optimal number of controllers at maximum resource efficiency before rescheduling.

In this case, the optimum number of controllers should be chosen for other considerations, such as the lower balance factor seen in Figure 9. Locations that optimize performance and minimize the expense of network loads to find an optimum number of controllers are seen in Figure 9. The execution of the rescheduling of resources would also compromise for much better reliability and minimal blocking of resources.

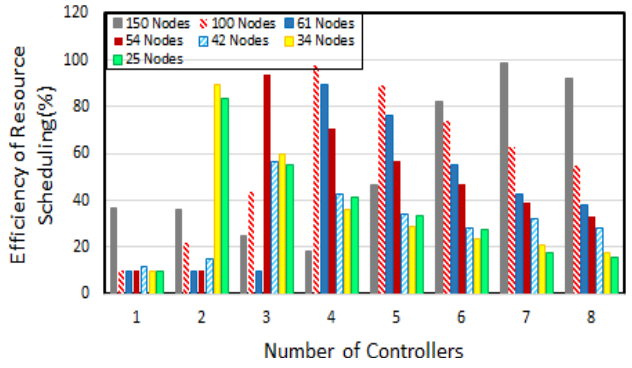

FIGURE 9. **The optimal number of controllers at maximum resource efficiency after rescheduling.**

Nevertheless, a higher number of controllers raises the deployment costs incurred, maintaining the controllers required. More controllers also mean a waste of resources, particularly for a densified network.

VII. CONCLUSION

In this paper, the integration of SDN and NFV for efficient 5G-CN was presented. We developed a resource management algorithm to identify the controller placement in distributed 5G SDN NFV-based network architecture. The optimal solution for both the location and the number of controllers under dynamic traffic was achieved through the proposed DCCPP based on the generalization of K-center algorithm and Graph Theory. Also, the GSR heuristic was used for switch assignments and scheduling of resource allocation problems. We also investigated the resource scheduling efficiency to measure the quality of the switch-to-controller assignment handled by the controller. Our framework achieved the proposed management scheduling algorithms to meet the load balancing and optimal resource management cost in the distributed control layer. The results indicated that the allocation and the optimum number of controllers under an effective decentralized policy could achieve a higher resource assignment efficiency to accomplish an exemplary network configuration.

REFERENCES

[1] E. O'Connell, D. Moore, and T. Newe, "Challenges associated with implementing 5G in manufacturing," *Telecom*, vol. 1, no. 1, pp. 48–67, Jun. 2020.

[2] M. S. Kumar and J. Prabhu, "Analysis of network function virtualization and software defined virtualization," *Int. J. Inform. Vis.*, vol. 1, no. 4, pp. 122–126, 2017.

[3] S. Manzoor, Z. Chen, Y. Gao, X. Hei, and W. Cheng, "Towards QoS-aware load balancing for high density software defined Wi-Fi networks," *IEEE Access*, vol. 8, pp. 117623–117638, 2020.

[4] Q. Wang, G. Shou, Y. Liu, Y. Hu, Z. Guo, and W. Chang, "Implementation of multipath network virtualization with SDN and NFV," *IEEE Access*, vol. 6, pp. 32460–32470, 2018.

[5] D. Li, S. Wang, K. Zhu, and S. Xia, "A survey of network update in SDN," *Frontiers Comput. Sci.*, vol. 11, no. 1, pp. 4–12, Feb. 2017.

[6] M. S. Bonfim, K. L. Dias, and S. F. L. Fernandes, "Integrated NFV/SDN architectures: A systematic literature review," *ACM Comput. Surv.*, vol. 51, no. 6, pp. 1–39, 2019.

[7] A. Basta, A. Blenk, K. Hoffmann, H. J. Morper, M. Hoffmann, and W. Kellerer, "Towards a cost optimal design for a 5G mobile core network based on SDN and NFV," *IEEE Trans. Netw. Service Manage.*, vol. 14, no. 4, pp. 1061–1075, Dec. 2017.

[8] R. Mijumbi, J. Serrat, J.-L. Gorricho, S. Latre, M. Charalambides, and D. Lopez, "Management and orchestration challenges in network functions virtualization," *IEEE Commun. Mag.*, vol. 54, no. 1, pp. 98–105, Jan. 2016.

[9] I. A. Lbachir, R. Es-salhi, I. Daoudi, and T. Saida, "Towards an autonomic approach for software defined networks: An overview," in *Proc. Int. Symp. Ubiquitous Netw.*, 2016, pp. 149–161.

[10] M. Shariat, Ö. Bulakci, A. De Domenico, C. Mannweiler, M. Gramaglia, Q. Wei, A. Gopalasingham, E. Pateromichelakis, F. Moggio, D. Tsolkas, B. Gajic, M. R. Crippa, and S. Khatibi, "A flexible network architecture for 5G systems," *Wireless Commun. Mobile Comput.*, vol. 2019, nos. 1530–8669, pp. 1–19, Feb. 2019.

[11] J. Lu, Z. Zhang, T. Hu, P. Yi, and J. Lan, "A survey of controller placement problem in software-defined networking," *IEEE Access*, vol. 7, pp. 24290–24307, 2019.

[12] T. Chin, M. Rahouti, and K. Xiong, "Applying software-defined networking to minimize the end-to-end delay of network services," *ACM SIGAPP Appl. Comput. Rev.*, vol. 18, no. 1, pp. 30–40, Apr. 2018.

[13] Y. Zhou, K. Zheng, W. Ni, and R. P. Liu, "Elastic switch migration for control plane load balancing in SDN," *IEEE Access*, vol. 6, pp. 3909–3919, 2018.

[14] B. P. R. Killi and S. V. Rao, "Controller placement in software defined networks: A comprehensive survey," *Comput. Netw.*, vol. 163, Nov. 2019, Art. no. 106883.

[15] A. Jalili, M. Keshtgari, and R. Akbari, "A new framework for reliable control placement in software-defined networks based on multi-criteria clustering approach," *Soft Comput.*, vol. 24, no. 4, pp. 2897–2916, Feb. 2020.

[16] G. Yao, J. Bi, Y. Li, and L. Guo, "On the capacitated controller placement problem in software defined networks," *IEEE Commun. Lett.*, vol. 18, no. 8, pp. 1339–1342, Aug. 2014.

[17] M. He, A. Basta, A. Blenk, and W. Kellerer, "Modeling flow setup time for controller placement in SDN: Evaluation for dynamic flows," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2017, pp. 1–7.

[18] T. Wang, F. Liu, J. Guo, and H. Xu, "Dynamic SDN controller assignment in data center networks: Stable matching with transfers," in *Proc. IEEE INFOCOM*, Apr. 2016, pp. 1–9.

[19] G. Li, X. Wang, and Z. Zhang, "SDN-based load balancing scheme for multi-controller deployment," *IEEE Access*, vol. 7, pp. 39612–39622, 2019.

[20] T. Wang, F. Liu, and H. Xu, "An efficient online algorithm for dynamic SDN controller assignment in data center networks," *IEEE/ACM Trans. Netw.*, vol. 25, no. 5, pp. 2788–2801, Oct. 2017.

[21] K. S. Sahoo, D. Puthal, M. Tiwary, M. Usman, B. Sahoo, Z. Wen, B. P. S. Sahoo, and R. Ranjan, "ESMLB: Efficient switch migration-based load balancing for multicontroller SDN in IoT," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 5852–5860, Jul. 2020.

[22] G. Nencioni, R. G. Garroppo, A. J. Gonzalez, B. E. Helvik, and G. Procissi, "Orchestration and control in software-defined 5G networks: Research challenges," *Wireless Commun. Mobile Comput.*, vol. 2018, pp. 1–18, Aug. 2018.

[23] T. Taleb, A. Ksentini, and R. Jäntti, "'Anything as a service' for 5G mobile systems," *IEEE Netw.*, vol. 30, no. 6, pp. 84–91, Nov. 2016.

[24] S. Clayman, E. Maini, A. Galis, A. Manzalini, and N. Mazzocca, "The dynamic placement of virtual network functions," in *Proc. NOMS IEEE/IFIP Netw. Oper. Manag. Symp.*, May 2014, pp. 1–9.

[25] P. Schulz, M. Matthe, H. Klessig, M. Simsek, G. Fettweis, J. Ansari, S. A. Ashraf, B. Almeroth, J. Voigt, I. Riedel, A. Puschmann, A. Mitschele-Thiel, M. Muller, T. Elste, and M. Windisch, "Latency critical IoT applications in 5G: Perspective on the design of radio interface and network architecture," *IEEE Commun. Mag.*, vol. 55, no. 2, pp. 70–78, Feb. 2017.

[26] P. Yang, N. Zhang, Y. Bi, L. Yu, and X. S. Shen, "Catalyzing cloud-fog interoperation in 5G wireless networks: An SDN approach," *IEEE Netw.*, vol. 31, no. 5, pp. 14–20, Sep. 2017.

[27] B. Han, J. Lianghai, and H. D. Schotten, "Slice as an evolutionary service: Genetic optimization for inter-slice resource management in 5G networks," *IEEE Access*, vol. 6, pp. 33137–33147, 2018.

[28] A. Ksentini, M. Bagaa, and T. Taleb, "On using SDN in 5G: The controller placement problem," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2016, pp. 1–6.

[29] S.-C. Lin, P. Wang, I. F. Akyildiz, and M. Luo, "Towards optimal network planning for software-defined networks," *IEEE Trans. Mobile Comput.*, vol. 17, no. 12, pp. 2953–2967, Dec. 2018.
[30] A. A. Z. Ibrahim and F. Hashim, "An architecture of 5G based on SDN NV wireless network," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 14, no. 2, pp. 725–734, 2019.
[31] S. H. Yeganeh and Y. Ganjali, "Kandoo: A framework for efficient and scalable offloading of control applications," in *Proc. HotSDN*, 2012, pp. 19–24.
[32] E. J. Kitindi, S. Fu, Y. Jia, A. Kabir, and Y. Wang, "Wireless network virtualization with SDN and C-RAN for 5G networks: Requirements, opportunities, and challenges," *IEEE Access*, vol. 5, pp. 19099–19115, 2017.
[33] D. Tuncer, M. Charalambides, S. Clayman, and G. Pavlou, "Adaptive resource management and control in software defined networks," *IEEE Trans. Netw. Service Manage.*, vol. 12, no. 1, pp. 18–33, Mar. 2015.
[34] G. Wang, Y. Zhao, J. Huang, and W. Wang, "The controller placement problem in software defined networking: A survey," *IEEE Netw.*, vol. 31, no. 5, pp. 21–27, Sep. 2017.
[35] T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta, and D. Sabella, "On multi-access edge computing: A survey of the emerging 5G network edge cloud architecture and orchestration," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 3, pp. 1657–1681, 3rd Quart., 2017.
[36] P. Agyapong, M. Iwamura, D. Staehle, W. Kiess, and A. Benjebbour, "Design considerations for a 5G network architecture," *IEEE Commun. Mag.*, vol. 52, no. 11, pp. 65–75, Nov. 2014.
[37] H. Zhang, N. Liu, X. Chu, K. Long, A. H. Aghvami, and V. C. M. Leung, "Network slicing based 5G and future mobile networks: Mobility, resource management, and challenges," *IEEE Commun. Mag.*, vol. 55, no. 8, pp. 138–145, Aug. 2017.
[38] S. Han, I. Chih-Lin, G. Li, S. Wang, and Q. Sun, "Big data enabled mobile network design for 5G and beyond," *IEEE Commun. Mag.*, vol. 55, no. 9, pp. 150–157, Sep. 2017.
[39] R. Guerzoni, R. Trivisonno, I. Vaishnavi, Z. Despotovic, A. Hecker, S. Beker, and D. Soldani, "A novel approach to virtual networks embedding for SDN management and orchestration," in *Proc. NOMS IEEE/IFIP Netw. Oper. Manag. Symp.*, May 2014, pp. 1–7.
[40] G. Wang, Y. Zhao, J. Huang, and Y. Wu, "An effective approach to controller placement in software defined wide area networks," *IEEE Trans. Netw. Service Manage.*, vol. 15, no. 1, pp. 344–355, Mar. 2018.
[41] R. Guerzoni *et al.*, "A novel approach to virtual networks embedding for SDN management and orchestration," in *Proc. IEEE/IFIP NOMS–IEEE/IFIP Netw. Oper. Manag. Symp. Manag. Softw. Defin. World*, 2014.
[42] G. Wang, Y. Zhao, J. Huang, and Y. Wu, "An effective approach to controller placement in software defined wide area networks," *IEEE Trans. Netw. Serv. Manag.*, vol. 15, no. 1, pp. 344–355, 2018.
[43] R. Chai, Q. Yuan, L. Zhu, and Q. Chen, "Control plane delay minimization-based capacitated controller placement algorithm for SDN," *EURASIP J. Wireless Commun. Netw.*, vol. 2019, no. 1, pp. 1–17, 2019.
[44] D. Hock, M. Hartmann, S. Gebert, M. Jarschel, T. Zinner, and P. Tran-Gia, "Pareto-optimal resilient controller placement in SDN-based core networks," in *Proc. 25th Int. Teletraffic Congr. (ITC)*, 2013.
[45] S. Lange *et al.*, "Specialized heuristics for the controller placement problem in large scale SDN networks," in *Proc. 27th Int. Teletraffic Congr. (ITC)*, 2015, pp. 210–218.
[46] M. He, A. Basta, A. Blenk, and W. Kellerer, "How flexible is dynamic SDN control plane?" in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, 2017, pp. 689–694, .
[47] T. Hu, P. Yi, Z. Guo, J. Lan, and Y. Hu, "Dynamic slave controller assignment for enhancing control plane robustness in software-defined networks," *Future Gener. Comput. Syst.*, vol. 95, pp. 681–693, Feb. 2019.
[48] M. Tanha, D. Sajjadi, R. Ruby, and J. Pan, "Capacity-aware and delay-guaranteed resilient controller placement for software-defined WANs," *IEEE Trans. Netw. Service Manage.*, vol. 15, no. 3, pp. 991–1005, Sep. 2018.
[49] A. Dixit, F. Hao, S. Mukherjee, T. V. Lakshman, and R. Kompella, "Towards an elastic distributed SDN controller," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 43, no. 4, pp. 7–12, Sep. 2013.
[50] A. A. Ateya, A. Muthanna, A. Vybornova, A. D. Algarni, A. Abuarqoub, Y. Koucheryavy, and A. Koucheryavy, "Chaotic salp swarm algorithm for SDN multi-controller networks," *Eng. Sci. Technol., Int. J.*, vol. 22, no. 4, pp. 1001–1012, Aug. 2019.
[51] F. Al-Tam and N. Correia, "On load balancing via switch migration in software-defined networking," *IEEE Access*, vol. 7, pp. 95998–96010, 2019.
[52] N. Cai, Y. Han, Y. Ben, W. An, and Z. Xu, "An effective load balanced controller placement approach in software-defined WANs," in *Proc. MILCOM IEEE Mil. Commun. Conf. (MILCOM)*, Nov. 2019, pp. 361–366.

ABEER A. Z. IBRAHIM (Member, IEEE) received the B.Sc. degree in electrical and electronic engineering from Alzaiem Alazhari University, Khartoum, Sudan, and the M.Sc. degree in communication engineering from Alzaiem Alazhari University, in 2003. She is currently pursuing the Ph.D. degree in wireless networking engineering with the Research Centre of Excellence for Wireless and Photonics Network (WiPNET), University Putra Malaysia (UPM).

From 2003 to 2017, she was a Lecturer with the Academy of Engineering Sciences, Sudan. From 2019 to 2020, she was a Visiting Researcher with the Info-Lab, Department of Computing and Communications, Lancaster University, U.K. Her research interests include future wireless networking, software-defined networks, network planning, and optimization. She was awarded the Ph.D. Fellowship from OWSD, in 2017.

FAZIRULHISYAM HASHIM (Member, IEEE) received the M.Sc. degree from Universiti Sains Malaysia and the Ph.D. degree in wireless communication networks engineering from the University of Sydney, Australia.

He is currently an Associate Professor and a Researcher with the Wireless and Photonic Network Research Center of Excellence (WiPNET), Universiti Putra Malaysia. He has authored over 140 technical papers (conference and journals) in the area of wireless communications and networks and holds two patents. His research interests include network security and quality of service of next-generation mobile networks, software-defined networks, blockchain, green communication systems, cognitive networks, and wireless sensor networks. He is a member of the Association for Computing Machinery (ACM). He was the Chair of IEEE Malaysia Young Professionals (YP), from 2013 to 2014, and of IEEE Malaysia Communications and Vehicular Technology Society (ComSoc/VTS) Joint Chapter, from 2015 to 2016. Under his stewardship ComSoc/VTS Malaysia won various awards notably the 2016 ComSoc Chapter of the Year Award and Chapter Achievement Award, and the 2016 IEEE Malaysia Outstanding Chapter Award.

NOR K. NOORDIN (Senior Member, IEEE) received the B.Sc. degree in electrical engineering from The University of Alabama, USA, in 1987, the M.Eng. degree from Universiti Technologi Malaysia, and the Ph.D. degree from Universiti Putra Malaysia. She is currently working as a Professor and also the Dean of Engineering Faculty at Universiti Putra Malaysia. She has published more than 300 journals, book chapters, and conference papers. She has led many research projects. Her research interests include wireless communication and network systems.

ADUWATI SALI (Senior Member, IEEE) received the B.Eng. degree in electrical electronics engineering from the University of Edinburgh, U.K., in 1999, the M.Sc. degree in communications and network engineering from Universiti Putra Malaysia, in 2002, and the Ph.D. degree in mobile and satellite communications from the University of Surrey, U.K., in 2009. She was the Deputy Director of the Research Management Centre (RMC), Universiti Putra Malaysia, from 2016 to 2019. She is currently a Professor with the Department of Computer and Communication System Engineering and a Researcher with the Wireless and Photonic Network Research Center of Excellence (WiPNET), Universiti Putra Malaysia. Her research interest includes mobile and satellite communication systems.

KEIVAN NAVAIE (Senior Member, IEEE) is currently with the School of Computing and Communications, Lancaster University, U.K. His research interests include provisioning dependable connectivity and positioning to intelligent cyber-physical systems. He is a Fellow of the IET, a Senior Fellow of the HEA, and a Chartered Engineer in U.K. He currently serves on the Editorial Board of the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, the IEEE COMMUNICATIONS LETTERS, and the IEEE COMMUNICATIONS SURVEYS & TUTORIALS.

SABER M. E. FADUL (Member, IEEE) received the B.Eng. (Hons.) and M.Sc. degrees in electrical and electronics engineering (Control and Automation Engineering) from the Sudan University of Science and Technology, Khartoum, Sudan, in 2001 and 2008, respectively, and the Ph.D. degree in control and automation engineering from the Faculty of Engineering, Universiti Putra Malaysia, Malaysia, in 2020.

Since 2018, he has been working as a Researcher with UPM and the Malaysia Automotive Robotics and IoT Institute (MARii), Cyberjaya, Malaysia. His research interests include the development of electric vehicle powertrain, data-driven system control, optimal control, optimization, and computational simulation.