

Received July 15, 2020, accepted August 15, 2020, date of publication August 18, 2020, date of current version August 28, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3017667

NFV Provisioning in Large-Scale Distributed Networks With Minimum Delay

MOHAMMED J. F. ALENAZI¹, (Senior Member, IEEE),
ABDULRAHMAN ALMUTAIRI¹, (Member, IEEE),
SALEH ALMOWUENA¹, **ABDUL WADOOD¹**, AND
EGEMEN K. ÇETINKAYA^{1,2}, (Senior Member, IEEE)

¹Department of Computer Engineering, College of Computer and Information Sciences, King Saud University, Riyadh 11451, Saudi Arabia

²Verizon, Basking Ridge, NJ 07920, USA

Corresponding author: Mohammed J. F. Alenazi (mjalenazi@ksu.edu.sa)

This work was supported by the Research and Development (R&D) Program (Research Pooling Initiative), Ministry of Education, Research Pooling Initiative-King Saud University, Riyadh, Saudi Arabia.

ABSTRACT Network function virtualization (NFV) and software-defined networking (SDN) are two technologies that have emerged to reduce capital and operational costs, and to simplify network management. In this paper, we propose an SDN-based system that provisions virtual network functions (VNFs) to minimize round trip time (RTT) delay and synchronization delay requirements. Our system uses graphic-theoretic approaches to place newly requested VNFs including four centrality functions – betweenness, degree, closeness, and Katz. The system performance is evaluated using two random graph topologies representing the physical and logical structures. The impact of increasing the number of deployed VNFs is considered. The results indicate that the degree and Katz selection methods mostly provide the minimum RTT for physical networks, whereas the betweenness selection provides minimum RTT values for logical networks. Moreover, the closeness selection method provides the best synchronization delay for both logical and physical networks.

INDEX TERMS Software-defined networking, network function virtualization, VNF placement, network performance, clustering, k -means, Gabriel graph, Waxman graph.

I. INTRODUCTION AND MOTIVATION

As the Internet scales up in terms of users and demand, a large number of new network devices are installed and upgraded on a daily basis. Scaling up network resources is becoming an expensive burden on network administrators, in order to keep up with the exponential growth of the Internet workload. Scalability cost and effort is making network provisioning and management increasingly more challenging. Another major challenge encountered by administrators responsible for networks composed of equipment provided by different vendors is interoperability. Performance, reliability, and durability are usually the main points of focus for network vendors [1], rather than ensuring interoperability with other vendor network components. Such challenges have resulted in higher capital expenditure (CapEx) and operational expenses (OpEx) [2]. Network function

virtualization (NFV) and software-defined networking (SDN) have emerged as new technologies to address the network provisioning and interoperability problems while minimizing cost.

Figure 1 presents a typical SDN/NFV architecture. NFV is a new technology that decouples the network functions such as routers, IDSs, and firewalls from network device hardware. A virtual network function (VNF) is the software implementation of network functions. NFV and VNF overcome several challenges associated with the use of specialized network hardware or middleboxes to provide new services. Specialized network hardware can be expensive, requires specialized operations personnel, has high energy costs, does not allow the addition of new functionality, and has short life-cycles [3]. These challenges not only raise the CapEx and OpEx of several service providers but also increase the inflexibility of management drastically [2].

The emergence of SDN technologies has led to a significant paradigm shift in the field of network

The associate editor coordinating the review of this manuscript and approving it for publication was Kezhi Wang.

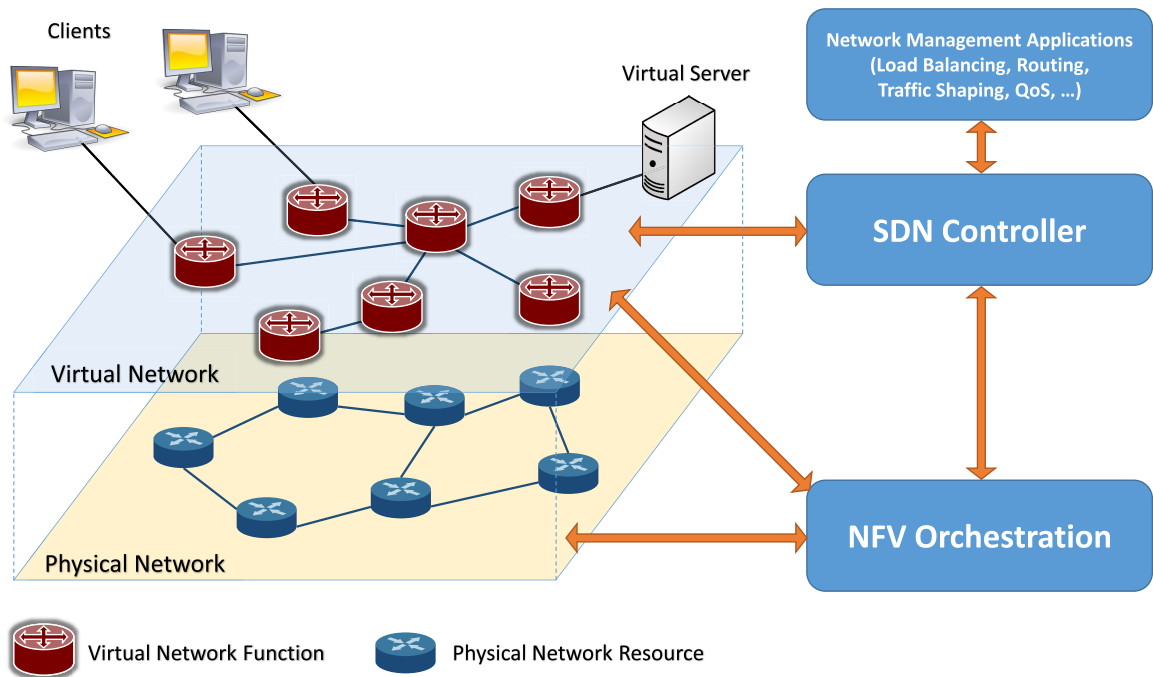


FIGURE 1. NFV/SDN architecture.

infrastructures [4]–[6]. In particular, by allowing the logical centralization of feedback control, decisions are based on a global view of the network. For example, controlling and managing virtual switches (e.g., Open vSwitch [7]) using an SDN controller (e.g., OpenDaylight [8]). This eases network optimization and enforces consistency of network policies. Such features supported by SDN provide a favorable environment for developing innovative applications, becoming an interesting research topic for both academia and industry.

The emerging SDN and NFV technologies have been introduced to improve several aspects of network services and operations. For instance, several approaches have been presented to improve network service orchestration (NSO) [9]–[11]. In addition, SDN has been used along with NFV to improve smart home networking [9]. For future networks, an architectural framework that integrates SDN and NFV for service provisioning has been introduced [12].

Another innovative utilization of SDN and NFV is the application of the two technologies in backbone networks. These core networks require low-latency systems to meet user demands in terms of high demands for throughput, bandwidth, quality of service (QoS), and end-to-end delay [13]–[15]. Most networked applications such as video-on-demand or cloud gaming rely on the backbone networks to provide on-time services [16]. Utilizing SDN/NFV to minimize the network delay can improve the overall user experience in real-time applications [17]–[19]. Thus, our objective in this study is to design a system that uses NFV to provide network services to meet network delay requirements while using SDN to manage and configure the newly deployed VNFs.

The contribution of this study is threefold. First, an NFV-SDN dynamic provisioning system is introduced to provide the locations of VNFs minimizing the overall system delay. This is crucial for real-time and mission-critical applications. Second, the proposed approach is evaluated within the context of four centrality metrics—node degree, node betweenness, node closeness, and Katz. Third, the system is applied to logical and physical networks to study the performance of the proposed system on network delays.

The remainder of this paper is organized as follows. A brief theoretical background of random graphs and centrality metrics is presented in Section II. Relevant related research work is then discussed. The proposed NFV-SDN dynamic provisioning system is explained in Section III. This includes a description of the k -VNFs selection algorithm used. In Section IV, the evaluation protocol used is described, including the dataset used in Subsection IV-A. In Section V, the obtained results are presented and discussed. Finally, concluding remarks and future research directions are provided in Section VI.

II. BACKGROUND AND RELATED WORK

In this section, a background on random graphs and centrality metrics is provided, in addition to discussing related research work.

A. RANDOM GRAPHS

In this section, we present two random graph models that are used to generate the dataset. Next, we present and discuss four graph-theoretic centrality node metrics that are used to provide NFV services.

1) WAXMAN GRAPHS

The Waxman model provides a probabilistic means of connecting nodes in a graph [20]. For two nodes $\{u, v\}$ with a Euclidean distance $d(u, v)$ between them, the probability of connecting them is:

$$P(u, v) = \beta e^{-\frac{d(u,v)}{L\alpha}} \quad (1)$$

where $\beta, \alpha \in (0, 1]$ and L is the maximum distance between any two nodes. Increasing β increases the link density and a large value of α corresponds to a high ratio of long links to short links. In this study, the Waxman model node locations are uniformly distributed. It has been established that Waxman graphs exhibit mesh-like properties of logical-level networks [21].

2) GABRIEL GRAPHS

Gabriel graphs are useful in modeling graphs with geographic connectivity that resemble grids [22], [23]. In a Gabriel graph, two nodes are connected directly if and only if there are no other nodes that fall inside the circle whose diameter is provided by the line segment joining the two nodes. The node locations are generated randomly using a uniform distribution with a range of $[0, 1]$ for both the x -axis and y -axis. It has been established that Gabriel graphs exhibit grid-like properties of physical-level networks [21].

B. NODE CENTRALITY

In this subsection, four node-centrality metrics are discussed: node degree, node betweenness, node closeness, and Katz. The significance and application of each node metric is discussed within the context of computer networks.

1) DEGREE

The degree centrality is one of the simplest and yet most commonly used centrality metrics. It can be defined as the number of link incidents to a node, representing the connectivity significance of a node [24]. Degree centrality is a local metric, as it depends only on the number of links locally connected. In communication networks, nodes with a high degree centrality are considered to be more significant than nodes with a lower degree centrality, as they provide connectivity to a larger number of links. The algorithmic complexity to determine the degree of a node is $O(n)$, in which n represents the number of nodes.

2) BETWEENNESS

Betweenness is a centrality graph metric that can be used for both nodes and links. Node betweenness is defined as the number of shortest paths through a node. In contrast, link betweenness is defined as the number of shortest paths through a link. Betweenness is considered to have a global significance as the betweenness value is impacted by the structure of the entire graph [25]. The algorithmic complexity to determine node betweenness is $O(nm)$, in which

n represents the number of nodes and m represents the number of links [26].

3) CLOSENESS

Closeness is a node centrality metric that measures the mean distance from a node to other nodes [24], [27]. In communication networks, closeness indicates the efficiency of the diffusion of a message in a network. The closeness is a node centrality metric that measures the mean distance from the node to other nodes [24], [27]. The algorithmic complexity to determine the closeness for a given graph is $O(n^3)$, in which n represents the number of nodes [28].

4) KATZ

Katz centrality measures the topological centrality of a node that helps to discover its relative influence on the network [29]. The Katz centrality is similar to the degree metric. Whereas the node degree measures the number of neighbors, the Katz centrality captures the significance of the neighbors [27], [29]. The Katz centrality (x_i) of node i is calculated as:

$$x_i = \gamma \sum_j A_{ij} x_j + \theta \quad (2)$$

where A_{ij} is the adjacency matrix of the graph G and γ and θ are control parameters for the Katz centrality. γ is used to control the effect of immediate neighborhood centrality and θ controls the initial centrality value. The algorithmic complexity to determine the closeness for a given graph is $O(n^3)$, in which n represents the number of nodes [30].

C. RELATED WORK

Several SDN/NFV approaches have been proposed to improve network provisioning and management. In this subsection, research work related to SDN/NFV is discussed.

Xu *et al.* introduced an algorithm for efficient NFV-enabled multicasting in SDNs [31]. Using an online algorithm, network throughput was improved via dynamic admissions of NFV-enabled multicast requests with no prior knowledge of future request patterns. Experimental and simulation tools were used to demonstrate performance gains over existing heuristics.

An SDN-based approach was presented by Ejaz *et al.* to support balancing mechanisms for a particular network while deploying SDN controllers as VNFs [32]. Their approach included a backup virtual SDN controller that is triggered to work once a certain threshold is exceeded. All hosts are notified of the presence of the new SDN controller; thus, new requests are balanced among all available SDN controllers. The approach was experimentally evaluated using Mininet, based on the fat-tree topology for a data center, with OpenDaylight as the main SDN controller. The results indicated performance improvement in terms of throughput and delay.

A novel system was proposed by Jawad *et al.* to reactively configure routers through SDN while using NFV to

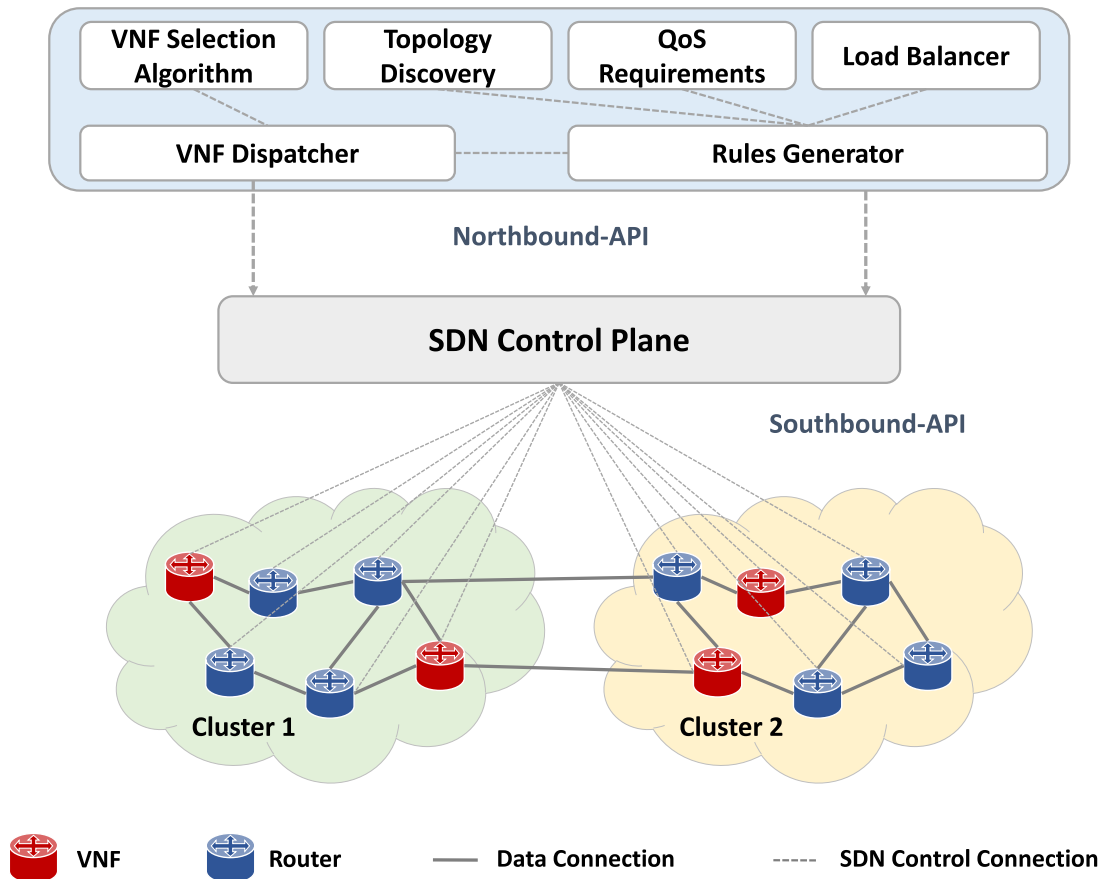


FIGURE 2. NFV provisioning system.

dynamically deploy network services [33]. The characteristics of the Internet of Radio Light (IoRL), including large bandwidth and location estimation accuracy, offered by its intelligent home IP gateway were exploited. A new service for IoRL clients was introduced to stream video from their nearest VNF, minimizing the end-to-end delay. Experimental evaluation indicated high throughput with no packet loss and an average jitter of 0.03 ms

Zarca *et al.* defined ANASTACIA, a network security management system to provide security and privacy in NFV/SDN-enabled Internet of Things (IoT) [34]. The system was tested against distributed denial-of-service (DDoS) and IoT malware attacks. The results confirmed that the system could automatically monitor, detect, react, and mitigate IoT cyberattacks. The system can apply the appropriate security policies as needed based on the type of attack detected. The overall network delay was minimized.

An investigation of the optimal placement of NFV middleboxes was undertaken by Ma *et al.*, considering the different middlebox traffic dependency relations [35]. In their work, the authors introduced a graph-theoretic formulation to model the traffic-aware placement of the interdependent middleboxes problem with the primary objective of load-balancing the deployed VNFs. It was found that the NFV problem is

NP-hard. A traffic and space aware routing heuristic was introduced along with an SDN-based prototype of a system to utilize such a heuristic. Extensive scale simulations were performed to demonstrate the effectiveness of the proposed system.

Wang *et al.* proposed a system to provide multipath routes among NFV components by utilizing SDN [36]. The proposed system consists of control and data planes. The control plane is responsible for functional components, which include multipath and flow splitting under network virtualization. With software design and NFV technology, network services were provided to ensure efficient computing and storage capabilities while implementing a multipath network for high throughput and resilience. For evaluation, open platform for NFV (OPNFV) was used as an experimental platform. The results indicated that flow splitting using multipath had improved the network performance while balancing the traffic load on the selected paths.

A system was proposed by Mouradian *et al.* to dynamically provide network services using NFV and SDN technologies as a response to disastrous events [37]. NFV was utilized to upgrade the pre-existing gateway and deploy gateway functions while exploiting SDN to reuse the same gateway services for various applications. The system services were

provisioned as VNFs and were chained automatically by SDN controllers. The results indicated that the system imposes an overhead on the overall management and orchestration; however, it provides significant performance gains. Moreover, the results illustrated the advantages of reusing and updating a pre-existing gateway.

Al-Kaseem *et al.* implemented a proof-of-concept testbed for the use of SDN/NFV technologies in cloud-based 6LoWPAN gateways [38]. The implementation included integrating SDN and NFV in the IEEE 802.15.4-based network, which is characterized by low-power and low-data rate sensor nodes. The results indicated that the SD-NFV approach improved the network discovery process time by 60% and the lifetimes of the nodes by 65% as compared to a baseline 6LoWPAN network.

All the related approaches discussed here focus mainly on improving the functionality of network operations after deployment, whereas in our study, the main objective is to improve the network performance prior to deployment. We provide a solution to place VNFs such that the overall network delay is minimized.

III. NFV-SDN DYNAMIC PROVISIONING SYSTEM

In this section, the proposed NFV-SDN dynamic provisioning system (NSDPS) is described. First, the system architecture is presented, including the system's components and their interactions. This is followed by a description of the proposed VNF locator algorithm.

A. SYSTEM COMPONENTS

The system consists of six main components, as depicted in Figure 2, which are outlined as follows:

- **The VNF selection algorithm component** is responsible for selecting the number and location of VNFs to satisfy the quality of service requirements. The pseudo-code of the VNF Locator Algorithm is presented in Algorithm 1.
- **The QoS requirements component** provides the set of QoS requirements for the network based on the used applications. For example, if the network is mainly used for real-time applications such as games or video conferencing, the minimum delay among all the network entities is set to ensure that such applications work correctly.
- **The topology discovery component** aims to detect newly added network resources and their connectivity. This information is stored as a graph $G = (N, L)$, where N represents the set of detected network resources, whereas L represents the set of links connecting these network resources.
- **The load balancer component** is responsible for distributing the load among VNFs to minimize the network services processing time.
- **The VNF dispatcher component** is responsible for deploying on-demand new VNFs based on QoS requirements. The component minimizes the number of

deployed VNFs to reduce overall network operation costs. For example, if the network encounters a sudden surge in traffic, the VNF dispatcher deploys more VNFs to handle such a surge.

- **The SDN rules manager component** creates and manages SDN switching rules, which are used to configure the network connectivity of deployed VNFs.

B. VNF LOCATOR ALGORITHM

In this section, the VNF locator algorithm used in the proposed system is described. It is a greedy algorithm used to locate k -VNFs locations based on a given objective. Although the algorithm is based on graph-theoretic centrality functions, the system proposed is generically designed to be compatible with any centrality function. Four objective functions are used – node degree, node betweenness, closeness, and Katz. The pseudocode of the algorithm is presented in Algorithm 1.

Algorithm 1: k -VNF Locator Algorithm

Functions:

centrality(N) := compute graph-theoretic centrality of nodes N

clustering(c, G) := determine c clusters of a graph G based on nodes locations

VNFSelect($k, attr$) := select k nodes based on attributes

Input:

G_i := input graph

Loc := nodes locations

c := number of required clusters

k := number of required VNFs for each cluster

Output:

selectedVNFs := a list of the selected controllers

begin

selectedVNFs = []

clusters = k -clustering(c, G_i)

for cluster **in** clusters **do**

centrality = []

for n **in** cluster.nodes() **do**

centralityValue = centrality(G_i)

centralityValue.append(n , centralityValue, Loc)

end

bestCandidateVNF =

VNFSelect(k , centralityValue)

selectedVNFs.add(bestCandidateVNF)

end

return selectedVNFs

The algorithm uses three functions: centrality(G), clustering(c, G), and VNFSelect($k, attr$). The centrality(G) function computes the graph-theoretic centrality of nodes N within a weighted graph. The Euclidean distance between two nodes is set as the link weight. This function can represent any graph-theoretic centrality function such as node degree, betweenness, or closeness. The clustering(c, G) function

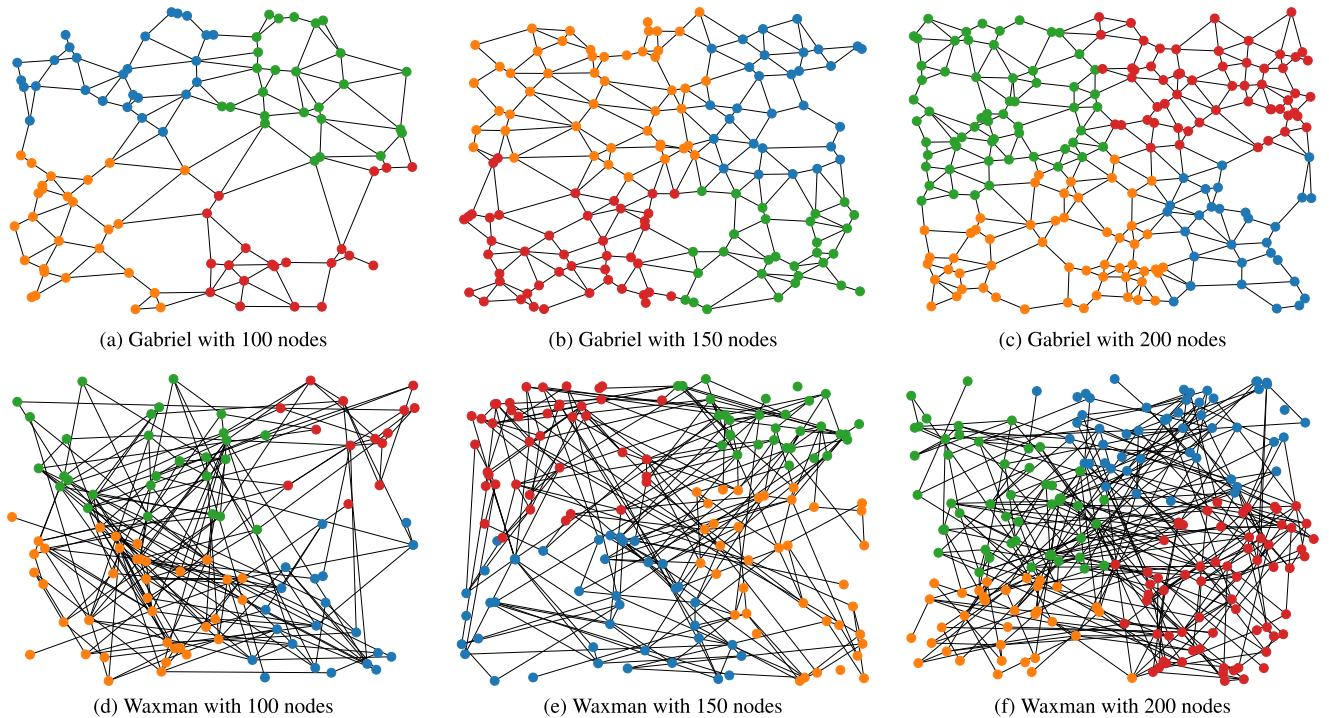


FIGURE 3. Gabriel and Waxman random graphs with different numbers of nodes.

determines c clusters of a graph G based on node locations. The clustering function is used to divide the nodes into c regions to ensure that the VNFs are geographically distributed. In this study, the k -means function is used to cluster the provided nodes based on their Euclidean distances. The selection function $\text{VNFSelect}(k, attr)$ returns the node with the best centrality value provided that it has not selected before. The algorithmic complexity of our algorithm is $O(n\mathcal{F})$, where n represents the number of nodes, and \mathcal{F} represents the centrality selection function, presented in Section II-B. The degree selection method makes the least expensive option for our algorithm, which yields $O(n^2)$. The Katz and closeness centrality functions incur the most expensive algorithmic complexity of $O(n^3)$, which makes the overall algorithmic complexity of our algorithm to $O(n^4)$. When the betweenness centrality is used, the algorithmic complexity becomes $O(n^2m)$, in which n represents the number of nodes and m represents the number of links.

As an example, to illustrate how this algorithm interacts with other system components, assume that the network is divided into four clusters ($c = 4$), and that five VNFs need to be deployed in each cluster ($k = 5$). Such parameters are provided by the network administrator using QoS requirements. They will be passed to the VNF locator algorithm to determine the locations of k VNFs for each cluster with a total of $c \times k$ VNFs. Thus, in this example, 20 locations will be sent to the VNF dispatcher for deployment. Finally, the SDN rules manager constructs SDN rules to configure the network connectivity of the deployed VNFs.

IV. EVALUATION METHODOLOGY

In this section, the methodology used to evaluate the proposed system is described. In particular, Subsection IV-A discusses the random graphs used as a dataset and their parameters, and Subsection IV-B presents the two performance metrics used to evaluate the proposed system. Graph modeling and performance metrics computations are done using the NetworkX Python library [39].

A. DATASET

For the dataset, two random graph models are used – Gabriel and Waxman, which are discussed in Section II-A. For each random graph model, graphs of the following orders are generated—100, 150, and 200 nodes. The nodes are generated with random positions within a $100 \text{ km} \times 100 \text{ km}$ area to model the backbone network of a city. The number of clusters is set to 4 clusters for all experiments. The resulting random graphs are depicted in Figure 3, with node color representing the cluster to which the node belongs to.

B. PERFORMANCE METRICS

The geometric Gabriel and Waxman graph models studied in this paper are unweighted—they do not consider node and link weights; therefore, we capture delay performance in terms of propagation delay in our simulation model. The propagation delay between two nodes is computed as the shortest path length divided by the propagation speed, which is assumed to be $2 \times 10^8 \text{ m/s}$. Based on the propagation delay,

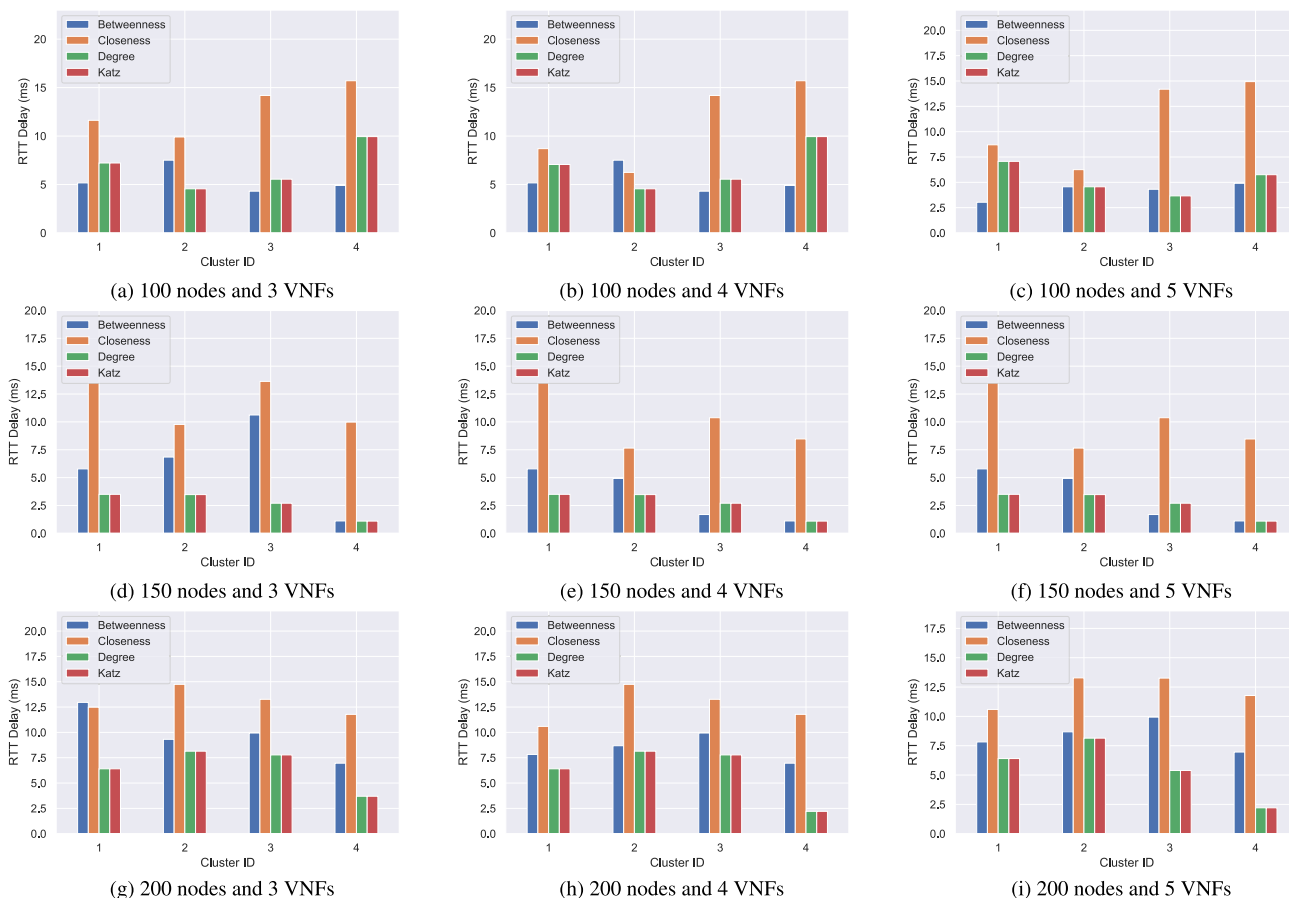


FIGURE 4. RTT delay for Gabriel graphs with different network sizes and different numbers of VNFs.

two performance metrics are used to evaluate the proposed system, round trip time delay and synchronization delay.

1) ROUND TRIP TIME DELAY

Round trip time (RTT) is considered in this work as a measure of the average propagation delay via the shortest paths between all network elements and their closest VNF in a given cluster. In a distributed virtualized network with multiple available VNF entities, it can be crucial for network elements to connect to the nearest VNF and minimize the RTT. For example, if the VNF is a dynamic host configuration protocol (DHCP) server, other network elements such as routers and hosts need to communicate with the DHCP server frequently to obtain the IP addresses, making RTT a significant measure.

2) SYNCHRONIZATION DELAY

In a distributed virtualized network, there are more than one VNF entities available to serve the network. Usually, such VNFs require to be synchronized, e.g., in the case of multiple virtual routers running a distributed routing algorithm such as internal border gateway protocol (iBGP). Minimizing the time needed to synchronize VNFs is a significant aspect of network design and can positively affect

network performance. The synchronization delay measures the average propagation delay via the shortest path between all VNFs in a given cluster. It captures the average time needed to exchange state information for a particular set of VNFs.

V. RESULTS AND DISCUSSION

The results obtained from implementing and evaluating the proposed NFV-SDN dynamic provisioning system are presented and discussed in this section.

A. EVALUATION OF RTT DELAY

Gabriel and Waxman random graphs are applied to the proposed system with different parameters to evaluate the network’s RTT delay. The system is configured to select the location of the VNFs based on betweenness, closeness, degree, and Katz. The RTT evaluation results for Gabriel with different graph order (100, 150, and 200) nodes are presented in Figure 4.

For Gabriel graphs with 100 nodes, it can be observed that the betweenness selection centrality method mostly provides minimum RTT values. For example, the RTT value of the betweenness method is approximately 5 ms for clusters 1, 3, and 4. However, for cluster 2, the degree and Katz selection

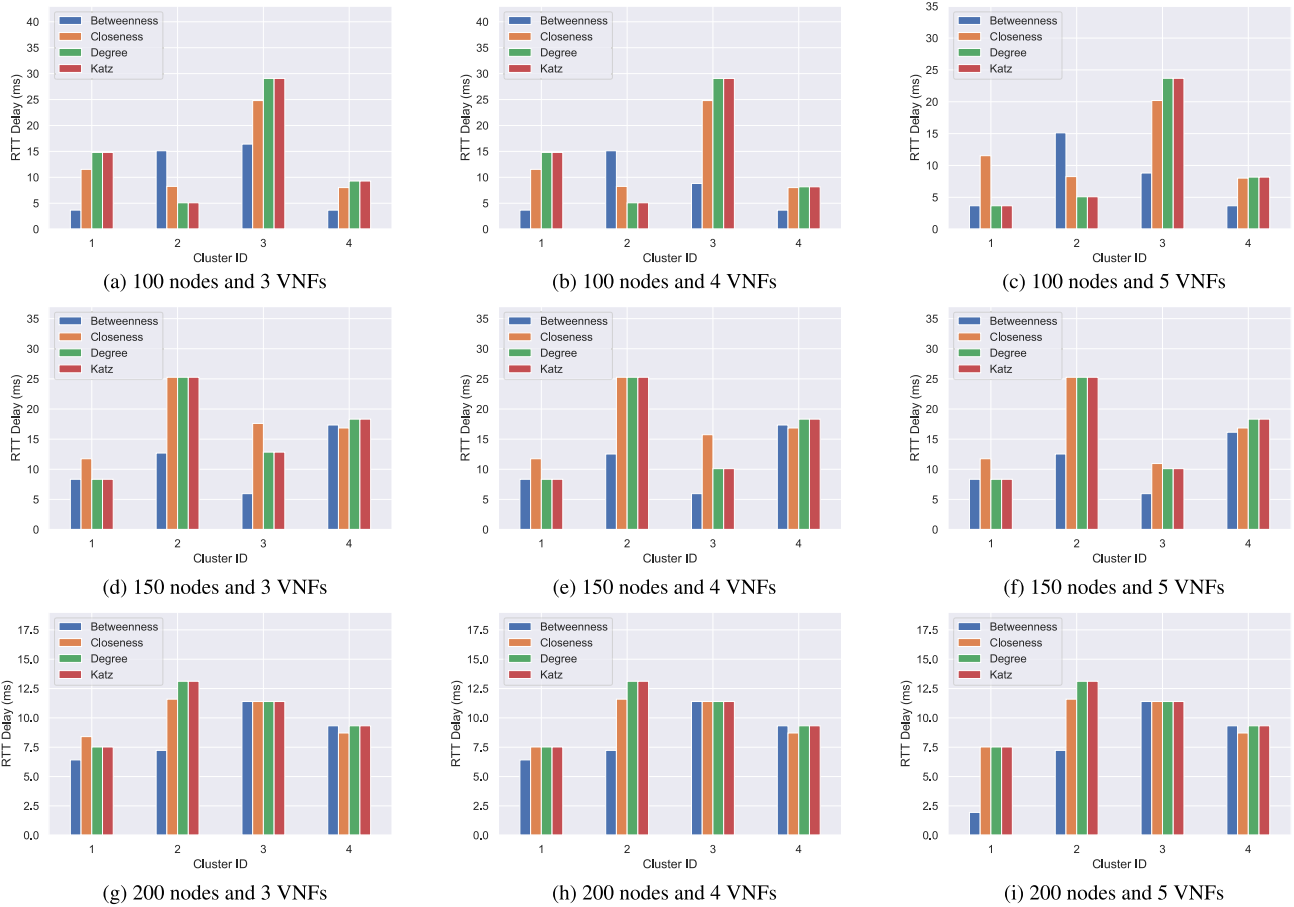


FIGURE 5. RTT delay for Waxman graphs with different network sizes and different numbers of VNFs.

method provides the minimum RTT of approximately 4 ms. In addition, the results indicate that the closeness selection method causes the largest RTT, with approximately 100% increase compared to the other methods.

For Gabriel graphs with 150 nodes, the results indicate that the degree and Katz selection methods mostly provide the minimum RTT when deploying 3, 4, and 5 VNFs for each cluster. In addition, the results indicate that the betweenness selection method causes slightly higher RTT values than the degree and Katz selection methods. Moreover, the results indicate that the closeness selection method yields the highest RTT values.

For Gabriel graphs with 200 nodes, the results indicate that the degree and Katz selection methods consistently provide the minimum RTT when deploying 3, 4, and 5 VNFs for each cluster. In addition, the results indicate that the betweenness selection method yields slightly higher RTT values than the degree and Katz selection methods. Further, the results indicate that the closeness selection method leads to the largest RTT values.

Considering all the graph sizes, it can be observed that the degree and Katz selection methods mostly provide the minimum RTT when deploying 3, 4, and 5 VNFs for each

cluster. The similarity between degree and Katz results show that they exhibit similar graph-theoretic properties for Gabriel graphs. Moreover, the results indicate increasing the node degree decreases the RTT values. This is mainly because if the VNF is selected based on node degree, it guarantees that the VNF has more connections than other candidate nodes. In addition, if the VNF is selected based on Katz centrality, it guarantees that the VNF is connected to neighbors with more connections than other candidate nodes. Thus, it can be concluded that the degree and Katz selection methods are good candidates for placing VNFs to minimize the RTT for a disturbed network with a grid structure (e.g., physical backbone networks or smart-grid systems).

The RTT evaluation results for the Waxman model with different graph orders (100, 150, and 200) are presented in Figure 5. For Waxman graphs with 100 nodes, it can be observed that the betweenness selection centrality method mostly provides the minimum RTT values for different numbers of deployed VNFs. For example, the RTT value of the closeness method is approximately 100 ms for clusters 1, 3, and 4. However, for cluster 2, the degree and Katz selection methods provide a minimum RTT of approximately 6 ms.

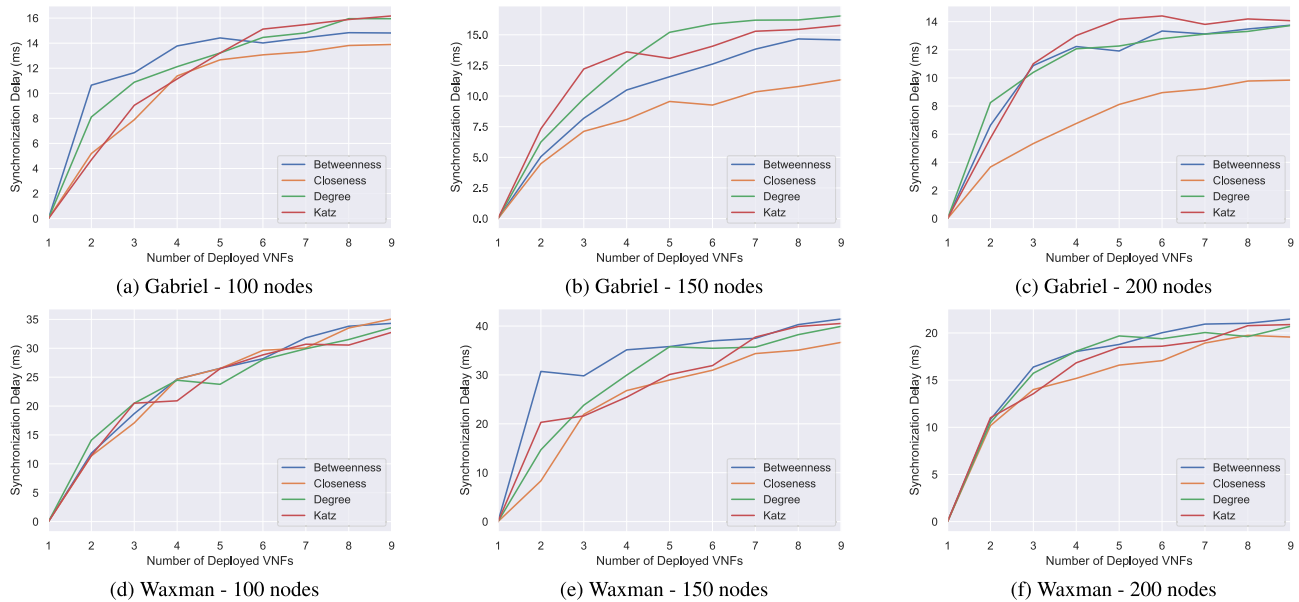


FIGURE 6. Synchronization delay for random graphs with different network sizes and different numbers of VNFs.

For Waxman graphs with 150 nodes, the results indicate that the betweenness selection methods mostly provide the minimum RTT when deploying 3, 4, and 5 VNFs for each cluster. In addition, the results indicate that the degree and Katz selection methods demonstrate slightly higher RTT values, whereas the closeness selection method provides the largest RTT values. For Waxman graphs with 200 nodes, the results indicate that all the selection methods provide similar RTT delay results. However, the betweenness selection methods provide slightly better results than the other selection methods.

Considering all the Waxman graph sizes, it can be observed that the betweenness selection method mostly provides the minimum RTT through deploying 3, 4, and 5 VNFs for each cluster. The results indicate when the node betweenness values increases, the RTT values increases. This is mainly because if the VNF is selected based on node betweenness, it guarantees that the VNF has more shortest paths passing than any other candidate nodes. Thus, it can be concluded that the betweenness selection method is a good candidate for placing VNFs to minimize the RTT for a distributed network with a tree-like structure (such as routing networks). In addition, we observe the selection method with the minimum RTT values for Waxman and Gabriel graphs are different. This indicates that the topological type of the network requires different centrality selection method to minimize RTT values.

B. EVALUATION OF SYNCHRONIZATION DELAY

Gabriel and Waxman random graphs are applied to the proposed system along with various parameters to check the network's synchronization delay. The number of deployed VNFs is varied between 1 and 9 to study the network's

synchronization delay as the number of VNFs increases. The number of clusters is set to 4 for each experiment.

The results for the Gabriel random graphs of orders 100, 150, and 200 nodes are presented in Figure 6a, Figure 6b, and Figure 6c, respectively. It can be clearly seen that the delay is zero when the number of deployed VNFs is 1, as there is no need to exchange status messages. However, as the number of deployed VNFs is increased, the synchronization delay among them increases. The results indicate that the closeness selection method consistently provides the lowest synchronization delay values, whereas the betweenness selection method causes the highest delay for 100 and 200 nodes. In contrast, the Katz method causes the worst synchronization delay for 150 nodes.

The results for the Waxman random graphs of orders 100, 150, and 200 nodes are presented in Figure 6d, Figure 6e, and Figure 6f, respectively. Similar to Gabriel graphs, the synchronization delay is zero when the number of deployed VNFs is 1, as there is no need to exchange status messages. However, as the number of deployed VNFs increases, the synchronization delay increases. The results indicate that all the selection methods yield similar results with 100 nodes. However, for 150 and 200 nodes, it can be clearly seen that the closeness selection method provides the lowest synchronization delay values.

Compared to Gabriel graphs, it can be observed that Waxman graphs causes almost double the synchronization delay as the number of deployed VNFs increases due to its graph connectivity structure. For example, for 100 nodes, Gabriel graphs yield a delay of approximately 16 ms for 9 VNFs, whereas Waxman graphs cause a delay of approximately 33 ms for all centrality methods. Moreover, the results indicate that the difference in delay between the two graph types

for 150 and 200 nodes decreases as the number of VNFs deployed increases.

The results indicate that regardless of topological type, the closeness metrics mainly yields minimum synchronization delays for both the Waxman and Gabriel graphs. This is because synchronization delay is minimized when the selected VNFs are geographically closer to each other. The highest-closeness nodes tend to exist in the center of the graph, which in turn are close to all other nodes. Therefore, the selected VNFs, based on closeness, are actually geographically closer to each other. As a result, it can be concluded that the closeness selection method provides the lowest synchronization delay values, as it selects VNFs that are close to each other, whereas the other methods select nodes in a distributed manner. The closeness selection method is suitable for distributed systems that have frequent state exchange messages such as firewalls and distributed routing.

VI. CONCLUSION AND FUTURE WORK

The emergence of NFV/SDN technologies has the significant effect of minimizing operational costs for network deployment and management. The NFV technology deploys virtual network functions such as routers, IDSs, and firewalls in a short amount of time compared to the deployment of physical equipment. The SDN technology provides standard APIs to configure and manage the network connectivity of middleboxes regardless of the vendor operating system. Thus, NFV/SDN technologies provide a flexible solution for network administrators and designers while minimizing installation and operation costs. The network can be easily built from diverse combinations of network equipment vendors without worrying about configuration and management complexity. In this study, an SDN-based system is introduced to provide network functions for backbone networks. The objective of the system is to minimize RTT and synchronization delays for the entire network. The system uses graph-theoretic centrality measures to place newly requested VNFs such that the overall delay is minimized. Four centrality functions are used: betweenness, degree, closeness, and Katz. For evaluation, the system is applied to random networks with physical and logical structures. The performance evaluation results indicated the impact of increasing the number of deployed VNFs. The system performs better when the degree and Katz selection methods are used in terms of minimum RTT for physical networks. The betweenness selection provides minimum RTT values for logical networks. In addition, the system provides the best synchronization delay for both logical and physical networks when the closeness selection method is used. For future work, we plan to deploy the proposed system in the smart-city environment and study the performance impact of such deployment compared to baseline systems.

REFERENCES

- [1] 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, p. 114, 2019.
- [2] E. Hernandez-Valencia, S. Izzo, and B. Polonsky, "How will NFV/SDN transform service provider opex?" *IEEE Netw.*, vol. 29, no. 3, pp. 60–67, May 2015.
- [3] R. Mijumbi, J. Serrat, J.-L. Gorricho, N. Bouten, F. De Turck, and R. Boutaba, "Network function virtualization: State-of-the-Art and research challenges," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 236–262, 1st Quart., 2016.
- [4] J. Matias, J. Garay, N. Toledo, J. Unzilla, and E. Jacob, "Toward an SDN-enabled NFV architecture," *IEEE Commun. Mag.*, vol. 53, no. 4, pp. 187–193, Apr. 2015.
- [5] M. Alsaedi, M. M. Mohamad, and A. A. Al-Roubaiey, "Toward adaptive and scalable OpenFlow-SDN flow control: A survey," *IEEE Access*, vol. 7, pp. 107346–107379, 2019.
- [6] M. J. F. Alenazi and E. K. Çetinkaya, "Resilient placement of SDN controllers exploiting disjoint paths," *Trans. Emerg. Telecommun. Technol.*, vol. 31, no. 2, pp. 1–17, Feb. 2020.
- [7] (Apr. 2020). *Open vSwitch*. Accessed: Apr. 27, 2020. [Online]. Available: <http://www.openswitch.org/>
- [8] (Apr. 2020). *The OpenDaylight Project*. Accessed: Apr. 27, 2020. [Online]. Available: <https://www.opendaylight.org/>
- [9] N. F. Saraiva De Sousa, D. A. Lachos Perez, R. V. Rosa, M. A. S. Santos, and C. Esteve Rothenberg, "Network service orchestration: A survey," *Comput. Commun.*, vols. 142–143, pp. 69–94, Jun. 2019.
- [10] R. Martínez, A. Mayoral, R. Vilalta, R. Casellas, R. Muñoz, S. Pachnicke, T. Szyrkowiec, and A. Autenrieth, "Integrated SDN/NFV orchestration for the dynamic deployment of mobile virtual backhaul networks over a multilayer (Packet/Optical) aggregation infrastructure," *J. Opt. Commun. Netw.*, vol. 9, no. 2, p. A135, 2017.
- [11] R. Muñoz, R. Vilalta, R. Casellas, R. Martínez, T. Szyrkowiec, A. Autenrieth, V. López, and D. López, "Integrated SDN/NFV management and orchestration architecture for dynamic deployment of virtual SDN control instances for virtual tenant networks," *IEEE/OSA J. Opt. Commun. Netw.*, vol. 7, no. 11, pp. B62–B70, 2015.
- [12] Q. Duan, N. Ansari, and M. Toy, "Software-defined network virtualization: An architectural framework for integrating SDN and NFV for service provisioning in future networks," *IEEE Netw.*, vol. 30, no. 5, pp. 10–16, Sep. 2016.
- [13] N. Huin, B. Jaumard, and F. Giroire, "Optimization of network service chain provisioning," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Paris, France, May 2017, pp. 1–7.
- [14] T. Kurimoto, S. Urushidani, H. Yamada, K. Yamanaka, M. Nakamura, S. Abe, K. Fukuda, M. Koibuchi, H. Takakura, S. Yamada, and Y. Ji, "SINET5: A low-latency and high-bandwidth backbone network for SDN/NFV era," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Paris, France, May 2017, pp. 1–7.
- [15] M. Birk, G. Choudhury, B. Cortez, A. Goddard, N. Padi, A. Raghuram, K. Tse, S. Tse, A. Wallace, and K. Xi, "Evolving to an SDN-enabled ISP backbone: Key technologies and applications," *IEEE Commun. Mag.*, vol. 54, no. 10, pp. 129–135, Oct. 2016.
- [16] V. Lopez, J. M. Gran Josa, V. Uceda, F. Slyne, M. Ruffini, R. Vilalta, A. Mayoral, R. Muñoz, R. Casellas, and R. Martínez, "End-to-End service orchestration from access to backbone," *J. Opt. Commun. Netw.*, vol. 9, no. 6, p. B137, 2017.
- [17] J. W. Guck, A. Van Bemten, and W. Kellerer, "DetServ: Network models for real-time QoS provisioning in SDN-based industrial environments," *IEEE Trans. Netw. Service Manage.*, vol. 14, no. 4, pp. 1003–1017, Dec. 2017.
- [18] R. Kumar, M. Hasan, S. Padhy, K. Evchenko, L. Piramanayagam, S. Mohan, and R. B. Bobba, "End-to-End network delay guarantees for real-time systems using SDN," in *Proc. IEEE Real-Time Syst. Symp. (RTSS)*, Paris, France, Dec. 2017, pp. 231–242.
- [19] H. Nam, K.-H. Kim, J. Y. Kim, and H. Schulzrinne, "Towards QoE-aware video streaming using SDN," in *Proc. IEEE Global Commun. Conf.*, Austin, TX, USA, Dec. 2014, pp. 1317–1322.
- [20] B. M. Waxman, "Routing of multipoint connections," *IEEE J. Sel. Areas Commun.*, vol. 6, no. 9, pp. 1617–1622, Dec. 1988.
- [21] E. K. Çetinkaya, M. J. F. Alenazi, Y. Cheng, A. M. Peck, and J. P. G. Sterbenz, "A comparative analysis of geometric graph models for modelling backbone networks," *Opt. Switching Netw.*, vol. 14, pp. 95–106, Aug. 2014.
- [22] K. R. Gabriel and R. R. Sokal, "A new statistical approach to geographic variation analysis," *Syst. Biol.*, vol. 18, no. 3, pp. 259–278, 1969.

- [23] D. W. Matula and R. R. Sokal, "Properties of Gabriel graphs relevant to geographic variation research and the clustering of points in the plane," *Geographical Anal.*, vol. 12, no. 3, pp. 205–222, Sep. 2010.
- [24] L. C. Freeman, "Centrality in social networks conceptual clarification," *Social Netw.*, vol. 1, no. 3, pp. 215–239, Jan. 1978.
- [25] T. Opsahl, F. Agneessens, and J. Skvoretz, "Node centrality in weighted networks: Generalizing degree and shortest paths," *Social Netw.*, vol. 32, no. 3, pp. 245–251, Jul. 2010.
- [26] U. Brandes, "A faster algorithm for betweenness centrality," *J. Math. Sociol.*, vol. 25, no. 2, pp. 163–177, 2001.
- [27] M. E. J. Newman, *Networks: An Introduction*, 1st ed. New York, NY, USA: Oxford Univ. Press, 2010, ch. 7, p. 199.
- [28] U. Kang, S. Papadimitriou, J. Sun, and H. Tong, "Centralities in large networks: Algorithms and observations," in *Proc. SIAM Int. Conf. Data Mining*, Mesa, AZ, USA, Apr. 2011, pp. 119–130.
- [29] L. Katz, "A new status index derived from sociometric analysis," *Psychometrika*, vol. 18, no. 1, pp. 39–43, Mar. 1953.
- [30] Y. Zhang, Y. Bao, S. Zhao, J. Chen, and J. Tang, "Identifying node importance by combining betweenness centrality and Katz centrality," in *Proc. Int. Conf. Cloud Comput. Big Data (CCBD)*, Shanghai, China, Nov. 2015, pp. 354–357.
- [31] Z. Xu, W. Liang, M. Huang, M. Jia, S. Guo, and A. Galis, "Efficient NFV-enabled multicasting in SDNs," *IEEE Trans. Commun.*, vol. 67, no. 3, pp. 2052–2070, Mar. 2019.
- [32] S. Ejaz, Z. Iqbal, P. Azmat Shah, B. H. Bukhari, A. Ali, and F. Aadil, "Traffic load balancing using software defined networking (SDN) controller as virtualized network function," *IEEE Access*, vol. 7, pp. 46646–46658, 2019.
- [33] N. Jawad, M. Salih, K. Ali, B. Meunier, Y. Zhang, X. Zhang, R. Zetik, C. Zarakovitis, H. Koumaras, M.-A. Kourtis, L. Shi, W. Mazurczyk, and J. Cosmas, "Smart television services using NFV/SDN network management," *IEEE Trans. Broadcast.*, vol. 65, no. 2, pp. 404–413, Jun. 2019.
- [34] A. Molina Zarca, J. B. Bernabe, R. Trapero, D. Rivera, J. Villalobos, A. Skarmeta, S. Bianchi, A. Zafeiropoulos, and P. Gouvas, "Security management architecture for NFV/SDN-aware IoT systems," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8005–8020, Oct. 2019.
- [35] W. Ma, J. Beltran, D. Pan, and N. Pissinou, "Placing traffic-changing and partially-ordered NFV middleboxes via SDN," *IEEE Trans. Netw. Service Manage.*, vol. 16, no. 4, pp. 1303–1317, Dec. 2019.
- [36] 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.
- [37] C. Mouradian, N. T. Jahromi, and R. H. Glioth, "NFV and SDN-based distributed IoT gateway for large-scale disaster management," *IEEE Internet Things J.*, vol. 5, no. 5, pp. 4119–4131, Oct. 2018.
- [38] B. R. Al-Kaseem and H. S. Al-Raweshidy, "SD-NFV as an energy efficient approach for M2M networks using cloud-based 6LoWPAN testbed," *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1787–1797, Oct. 2017.
- [39] A. A. Hagberg, D. A. Schult, and P. J. Swart, "Exploring network structure, dynamics, and function using NetworkX," in *Proc. 7th Python Sci. Conf. (SciPy)*, Pasadena, CA, USA, pp. 11–15, Aug. 2008.



MOHAMMED J. F. ALENAZI (Senior Member, IEEE) received the B.S. and M.S. degrees in computer engineering and the Ph.D. degree in computer science from The University of Kansas, in 2010, 2012, and 2015, respectively. He is currently an Associate Professor of computer engineering with King Saud University. His research interests include, but are not limited to, design, implementation, and analysis of resilient and survivable networks, network routing design and implementation, development and simulation of network architectures and protocols, performance evaluation of communication networks, algorithmic graph approach for modeling networks, and mobile ad hoc networks (MANET) routing protocols.



ABDULRAHMAN ALMUTAIRI (Member, IEEE) received the M.S. and Ph.D. degrees from Purdue University, West Lafayette. He is currently an Assistant Professor with the Department of Computer Engineering, King Saud University, Saudi Arabia. His research interests include network security and cloud computing security.



SALEH ALMOWUENA received the B.Eng. degree in computer engineering from King Saud University, Saudi Arabia, in 2006, the M.A.Sc. degree in electrical and computer engineering from the University of Victoria, Canada, in 2010, and the Ph.D. degree in computing science from Simon Fraser University, Canada, in 2016. He is currently an Assistant Professor with the Department of Computer Engineering, King Saud University. He is also working on improving the bandwidth usage and power consumption caused by adaptive video streaming over heterogeneous cellular networks. His research interests include data aggregation in the IoT, configuration of single frequency networks, and software defined networking.



ABDUL WADOOD received the Ph.D. degree in signal and image processing from the University of Poitiers, France, in 2011. He is currently an Associate Professor with the Department of Computer Engineering, College of Computer and Information Sciences, King Saud University. His research interests include color image watermarking, multimedia security, privacy, and biometric template protection.



EGEMEN K. ÇETINKAYA (Senior Member, IEEE) received the B.S. degree in electronics engineering from Bursa Uludağ University, Bursa, Turkey, in 1999, the M.S. degree in electrical engineering from the University of Missouri-Rolla, in 2001, and the Ph.D. degree in electrical engineering from The University of Kansas, in 2013. From 2014 to 2018, he was an Assistant Professor of electrical and computer engineering with the Missouri University of Science and Technology. He held various positions with Sprint as a Support, a System, and a Design Engineer from 2001 to 2008. Since 2018, he has been a Principal Member of the Technical Staff with Verizon focusing on development of the 5G network and orchestration architecture. His research interest includes resilient networks. He is a member of ACM.

...