IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received May 28, 2020, accepted June 9, 2020, date of publication June 18, 2020, date of current version June 30, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3003263

A Novel Strategy to Achieve Bandwidth Cost Reduction and Load Balancing in a Cooperative Three-Layer Fog-Cloud Computing Environment

MIRZA MOHD SHAHRIAR MASWOOD[1], (Member, IEEE),
MD. RAHINUR RAHMAN[1], (Member, IEEE),
ABDULLAH G. ALHARBI[2], (Member, IEEE),
AND DEEP MEDHI[3], (Fellow, IEEE)
[1]Department of Electronics and Communication Engineering, Khulna University of Engineering & Technology, Khulna 9203, Bangladesh
[2]Department of Electrical Engineering, Faculty of Engineering, Jouf University, Sakaka 72388, Saudi Arabia
[3]Department of Computer Science Electrical Engineering, University of Missouri–Kansas City, Kansas City, MO 64110, USA

Corresponding author: Mirza Mohd Shahriar Maswood (mmnt7@mail.umkc.edu)

ABSTRACT Recently, IoT (Internet of Things) has been an attractive area of research to develop smart home, smart city environment. IoT sensors generate data stream continuously and majority of the IoT based applications are highly delay sensitive. The initially used cloud based IoT services suffers from higher delay and lack of efficient resources utilization. Fog computing is introduced to improve these problems by bringing cloud services near to edge in small scale and distributed nature. This work considers an integrated fog-cloud environment to minimize resource cost and reduce delay to support real-time applications at a lower operational cost. We first present a cooperative three-layer fog-cloud computing environment, and propose a novel optimization model in this environment. This model has a composite objective function to minimize the bandwidth cost and provide load balancing. We consider balancing load in both links' bandwidth and servers' CPU processing capacity level. Simulation results show that our framework can minimize the bandwidth cost and balance the load by utilizing the cooperative environment effectively. We assign weight factors to each objective of the composite objective function to set the level of priority. When minimizing bandwidth cost gets higher priority, at first, the demand generated from the traffic generator sensors continues to be satisfied by the regional capacity of layer-1 fog. If the demand of a region goes beyond the capacity of that region, remaining demand is served by other regions layer-1 fog, then by layer-2 fog, and finally by the cloud. However, when load balancing is the priority, the demand is distributed among these resources to reduce delay. Link level load balancing can reduce the queueing delay of links while server level load balancing can decrease processing delay of servers in an overloaded situation. We further analyzed how the unit bandwidth cost, the average and maximum link utilization, the servers' resources utilization, and the average number of servers used vary with different levels of priority on different objectives. As a result, our optimization formulation allows tradeoff analysis in the cooperative three-layer fog-cloud computing environment.

INDEX TERMS Fog computing, IoT, optimal resource management, load balancing, task offloading.

I. INTRODUCTION

Internet of Things (IoT) devices such as home voice controllers, smart TVs, smart locks in smart homes, road traffic and air quality monitors in smart cities, continuous glucose monitor, and fitness bands in healthcare are only a few examples, which are introduced in recent years. A huge growth in the number of IoT devices equipped with sensors is observed recently. These sensors collect the data from these devices. Later, the collected raw data is used to produce aggregated

The associate editor coordinating the review of this manuscript and approving it for publication was Asad Waqar Malik.

information and to take automated decision. It is projected that 26 billion IoT devices are to be installed by 2020 [1], and 75.44 billion by 2025 [2]. This is a fivefold increase from 2015 to 2025.

These sensing devices also come with restrictions, e.g., low computational capability, energy, memory, and storage capacity. Furthermore, many of the IoT applications need prompt response in situations like fire in a home, failure of emergency home appliances, patient requiring emergency medical assistance, and so on. To address these issues, cloud computing is projected to satisfy the computational and storage need. However, cloud is usually many hops away from these sensors and thus, could result in higher cost and delay such as due to higher propagation and transmission delay. A new computing paradigm named fog computing, a branch of Mobile Edge Computing, is introduced to supplement cloud to address the needs of IoT services [3]. In a fog environment, facilities of cloud is brought closer to the users and IoT devices but in a small scale. With the advent of fog computing, tasks with higher computational complexity but less delay sensitivity can be sent to the cloud. Then, the tasks with the opposite requirement can be served by a nearby fog server.

Fog computing architecture is often considered as a three-layered architecture [4] i.e., fog layer-1, fog layer-2, and cloud. Furthermore, fog layer-1 is divided into different regions as shown in Fig. 1. Service requests generated from the sensors can be served by these resources. Since, layer-1 is divided into different regions, if services are limited to be provided by one region, then two possibilities might arise: one, fog servers of this region might be overloaded, and two, there might be idle resources available in other regions.

Overloaded fog servers within a region may not be able to maintain Quality of Service (QoS) and may incur higher delay. Furthermore, if the regional fog servers of layer-1 are the only source of providing service, then, there is a high probability that the requests generated from that region may face blocking at a overloaded demand situation in that region. Thus, an efficient and cooperative resources management through a load balancing scheme is required to reduce the delay and the probability of requests' blocking. However, this scheme might face higher bandwidth cost as the regional fog servers are often single hop away from the sensors. Therefore, if we need to route the traffic to the fog servers of other regions or even to layer-2 or cloud servers, extra bandwidth cost will be added. Furthermore, such a situation might also arise when some links are highly overloaded, which results in a higher queuing delay. Thus, a joint optimization scheme is required to balance the computational load among all the available servers as well as to reduce the bandwidth cost and link level overloads.

A Software Defined Network (SDN) controller is a favorable network element for achieving this optimization [5]–[7]. An SDN controller has absolute control over the network by separating the control plane and the data plane. The control plane is used for OpenFlow communication from the SDN controller to OpenFlow switches to generate flow table [6]. The data plane is used for data transmission from one device to another. The SDN controller has two interfaces, northbound and southbound. The northbound interface provides Application Programming Interface (API) to support different applications developed by fog service providers. The southbound interface is used for open flow communication. To the best of our knowledge, no work has considered these issues jointly in a fog-and-cloud computing paradigm so far.

Our approach focuses on minimizing bandwidth cost and load balancing. In doing so, we capture the latency for selecting fog against cloud through different weights for the bandwidth cost per unit of flow. Secondly, an efficient load balancing scheme can implicitly influence reduction in queuing delay at the link level. Also, at the server level, processing of a request would be faster if it is not overloaded. Therefore, by combining both link and server level load balancing, the overall delay is also reduced.

The novel contributions of this work beyond the state-of-the-art, in terms of the optimal resource provisioning in a cooperative fog environment, are as below:

- We propose a novel Mixed-Integer Linear Programming (MILP) based optimization model to minimize a composite objective function. Our model can minimize the bandwidth cost of establishing paths from a cluster point (CP) to a server. It also considers load balancing jointly both at the network and the server level.
- We use two types of resources in our approach: network resources (bandwidth) and server resources (CPUs' processing capacity). Thus, our model is a unified model on resource optimization between the network and the servers.
- A series of systematic studies is conducted using different values of the weight factors associated with each goal of composite objective function. Then, we present how the average link utilization, maximum link utilization, bandwidth cost, servers' resources utilization, and average number of servers used vary. In this process, we use both homogenous and heterogeneous bandwidth and servers' resource requirement.
- Using this study, we further analyze how the goal of cooperative fog computing is achieved.

The rest of the paper is organized as follows. We discuss the related works and the uniqueness of our work compared to these works in Section II. In Section III, the system assumption and model formulation is presented. In Section IV, simulation study setup and result analysis are shown. Finally, Section V, concludes the paper along with some future direction to extend this work.

II. RELATED WORK

Fog computing can utilize edge network resources, core network resources, and cloud resources [8]. A fog computing orchestration framework, which supports IoT applications,

TABLE 1. Comparison of related works from different perspectives.

Parameters Reference Papers	Link level load balancing	Server level load balancing	Bandwidth cost minimization	Delay/ latency minimization	Energy Consumption minimization	Number of fog layers	Cooperative fog computing	Is SDN based?
[5] Zahid et al.	No	Yes	No	Yes	No	One	No	Yes
[6] He et al.	No	Yes	No	Yes	No	One	Yes	Yes
[7] Kadhim at el.	No	Yes	No	Yes	No	One	Yes	Yes
[16] Chen at el.	No	No	No	Yes	No	One	No	No
[17] Meng at el.	No	No	No	No	Yes	One	Yes	No
[18] Khattak at el.	No	Yes	No	Yes	No	One	Yes	No
[19] Xu at el.	No	Yes	No	No	No	Two	Yes	No
[20] Mostafa at el.	No	Yes	No	Yes	No	One	Yes	No
[21] Dong at el.	No	No	No	Yes	Yes	One	Yes	No
[22] Mushunuri at el.	No	No	No	Yes	Yes	One	No	No
[23] Deng at el.	No	No	No	Yes	Yes	One	No	No
[24] Ningning at el.	Yes	Yes	No	No	No	One	Yes	No
Our proposed	Yes	Yes	Yes	No	No	Two	Yes	Yes

is presented in [9], [10]. Several critical challenges associated with fog computing such as architecture, interfacing and programming, offloading of computation, optimal provisioning of resources, security and privacy problems are discussed broadly in [11]–[13]. Fog computing brings opportunities to provide quality and prompt services in different areas; health care is one them. This sector is greatly benefited with the advent of fog computing [14], [15]. However, transmission delay and processing delay have significant impact in data transmission between user equipments and fog servers [16]. Different optimization models are proposed in [8], [16] to overcome these constraints of processing and transmission delay. A hybrid task offloading scheme combining cloud and fog is proposed to reduce the energy consumption in communication network and computation considering delay as a constraint [17].

A fog computing environment comprises with devices of different characteristics. To maintain an uninterrupted connection among these devices, a proper integration and management scheme is necessary [25]. Fogernetes, a fog computing platform that enables management and deployment of fog applications, is presented in [26]. A simple and general model for fog computing infrastructures to continuously maintain the required QoS in multicomponent IoT applications is proposed in [27]. Specially, to connect large number of heterogeneous nodes, adoption of SDN and NFV techniques are effective [28]. In [29] authors critically reviewed the SDN and NFV for fog computing-based solutions to combat against the main challenges of IoT.

Hill Climbing Load Balancing Algorithm on fog computing is proposed in [5]. However, the scope of this work is limited for smart grid where load balancing is done only at VMs/servers level. Different approaches for server level load balancing to balance the load among fog servers in fog computing environment are proposed in [18], [19]. However, several fog computing applications like augmented reality, surveillance, and smart cities usually have a great extent of demand for both bandwidth and servers' resources. Thus, an optimization framework dealing with link level and server level load balancing jointly is required. Multilevel load balancing for fog computing is proposed in [20]. The problem of uneven load distribution for static load balancing is overcome here using historical resource selection. However, due to the transition of loads from one micro data-center to another micro data center and to cloud, link level load balancing needs to be analyzed here.

Cooperative fog computing system having the capability of offloading computational tasks in a fairness environment is introduced in [21]. In this work, authors proposed a joint optimization of QoE (average response time) and energy (average energy consumption) in an integrated fog computing platform. A fog computing architecture along with a framework to improve the QoS for IoT applications is proposed in [30]. Their proposed system is supposed to support cooperation among fog nodes in a given location to allow data processing in a shared mode. At the same time, it can satisfy QoS and serve largest number of service requests. However, there is lack of feasibility study and simulation evidence in this work.

A framework to optimize energy consumption by improving battery usage and reducing delay in a fog computing environment is shown in [22]. This framework is implemented by incorporating optimization libraries within the Robot Operating System (ROS). It is deployed on robotic sensors. A novel server level load balancing strategy for the effective usage of server resources and reduction of delay or latency using SDN in cooperative Internet of Vehicles (IoV) network is proposed in [6] and [7]. In [7], they used fog concept in cluster computing with local and global load balancing using local and global load balancer (SDN) respectively. A framework is developed in [23] to analyze the trade off between power consumption and transmission delay in a fog-cloud computing system. This work considered the workload distribution by addressing the overall power consumption as the objective function. Our work differs in two ways: we consider a three layer fog-cloud framework and we use a composite objective function that optimizes the tradeoff between bandwidth cost in the edge-to-fog and fog-to-cloud components as well as network-level and server-level load balancing..

In [24], they proposed a dynamic load balancing mechanism based on graph repartitioning. Load balancing for new node joining into the system and leaving from that system is also studied. In [31], a load balancing scheme to jointly balance the load in both network and server level is presented. An optimization model to minimize a composite objective function, consisting of bandwidth and energy cost, is formulated and evaluated in [32]. However, the works of [31] and [32] are done only for a cloud computing environment and did not consider a joint fog-cloud computing architecture.

All these works discussed thus far considered a number of issues related to the cloud and fog computing environment. However, to our knowledge, no work has considered the joint optimization of bandwidth cost and load balancing, both at the network and the server level together in a cooperative three layer fog-cloud computing environment. Secondly, we study the fog-cloud trade-off by varying priorities of different objectives. This is the key contribution of our work beyond the state of the art thus far; therefore, our work fills a significant gap in this area of research.

III. SYSTEM ASSUMPTION AND PROBLEM FORMULATION

In our framework, fog servers are positioned as intermediate compute nodes. The fog layer consisting of fog servers are independent without having dependency from specific devices. In case of emergency situations such as natural disaster, device failure, or due to the spatial and temporal diversity of demand, additional compute and network resources might be required. Thus, a cooperative fog computing scheme should be designed where these additional resources can be borrowed from nearby regions of the same fog layer. This is layer-1 in the fog topology as shown in Fig. 1 (if idle resources available). Also these resources can be taken from more powerful layer (layer-2), or even from the cloud. Therefore, we consider task offloading from fog to cloud or from edge to core on demand.

Now, a regional gateway is itself a fog device, cluster point, and OpenFlow protocol supported switch. Again, the LAN ports of the gateway may be connected with independent servers or with fog devices (other LAN switches). These fog devices are other candidates for becoming fog servers. Fog devices can send their overall conditions including current load, remaining capacity, and other required information to SDN controller over OpenFlow communication. Using this information, SDN controller can formulate data forwarding rules and flow tables of cluster points (CP) are populated, accordingly. Data flow and open flow communication among fog devices, cloud and SDN controller are shown in Fig. 1 for better understanding. We assume that the total servers' resource (computational capacity) demand can be satisfied primarily in its own region by each CP, then by the other regions of fog layer-1 server nodes, then by fog layer-2 server nodes, and finally any higher computational requirement would need to be satisfied by the cloud servers. Hence, a path with sufficient bandwidth needs to be established from a CP to a server, if that server is used to serve any portion of request generated from that CP.

In this work, we consider the data sensed or collected by the sensors, which are attached with the IoT devices need bandwidth and servers' resources to be processed. We denote the sensed data that require further processing to take automated decision from each sensor node as a request. Thus, each request consists of 2-tuple $\langle h, r \rangle$. Here, h is the bandwidth demand and r is the servers' resource demand. To formulate the mathematical model, we consider aggregated requests generated from all sensor nodes within a region. The CP is the central access point of a region which is connected with all the sensor nodes within this region. Thus, we consider CP as the source of this aggregated request. Now, this aggregated requests also consist of two demands, i.e., H_i as aggregated bandwidth demand and R_i as aggregated servers' resource demand within a region transferred through CP to be processed. We do not consider prioritizing the aggregated requests generated from any one CP within a region over the aggregated requests of other CPs of different regions in this work.

We formulate a MILP optimization model to mathematically represent the system assumption of this work. Then, we consider that an SDN controller is responsible to solve the optimization model and allocate resources according to the solution like [6]. Here the SDN controller is placed in fog layer-2 which has Open Flow communication with network devices and cloud. The aggregated demand profile is sent from each region of fog layer-1 to the SDN controller where it is solved using the model. Since, the size of demand profile is negligible compared to the size of data needs to be transferred, we ignore the cost of transferring demand profile in this work. Then, control signals are generated from the controller and sent using the control plane to allocate resources to the responsible devices based on the solution. The notations used to formulate the model is explained in Table 2.

A. CONSTRAINTS

In our formulation, H_i is the total amount of bandwidth demand generated from all sensor nodes within the region of cluster point i. Thus, H_i can be mathematically represented by:

$$H_i = \sum_{n \in N_i} h_{ni}, \quad i \in I \tag{1}$$

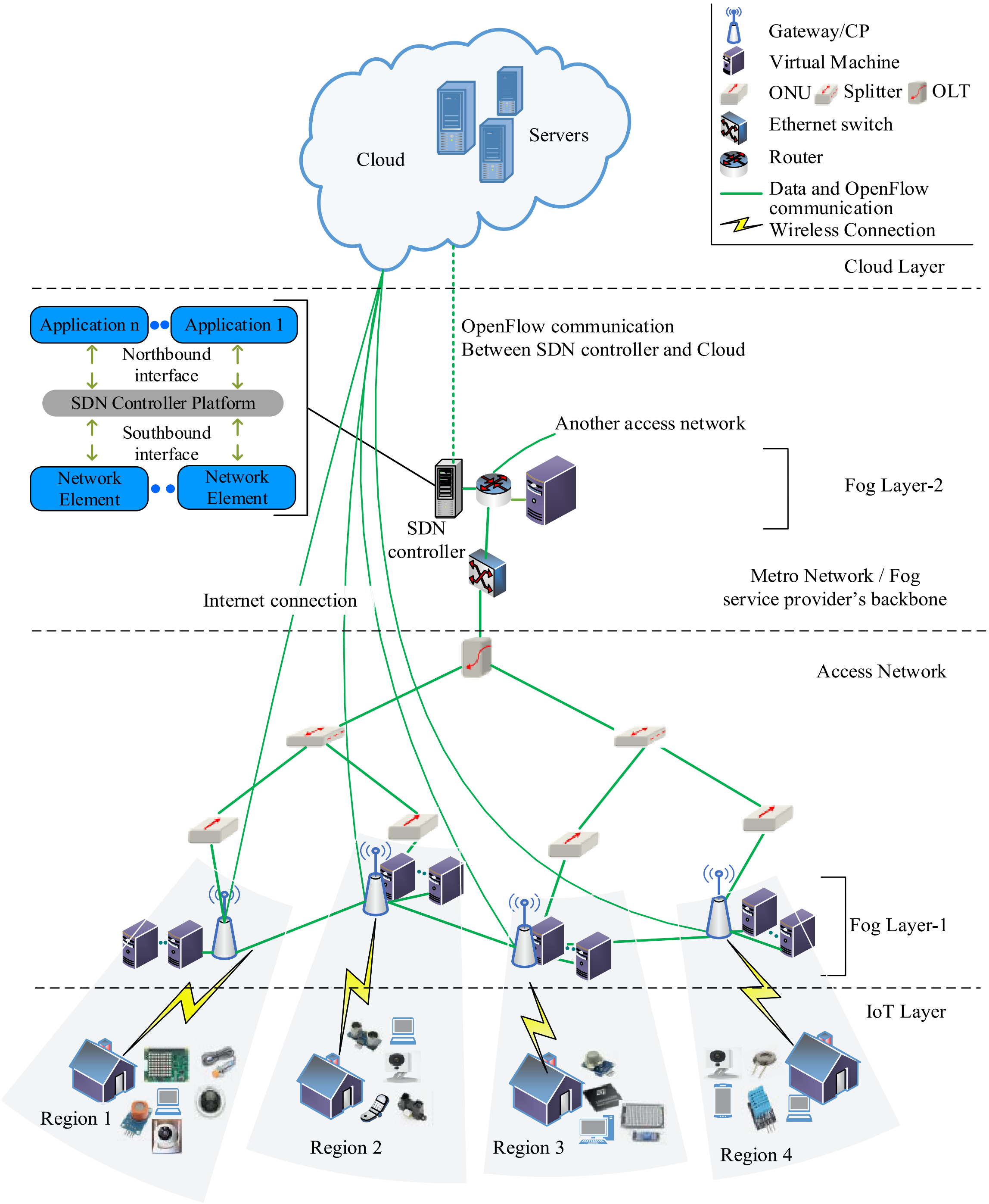

FIGURE 1. **Cooperative fog network architecture.**

Similarly, total amount of resource demand generated from all sensor nodes under cluster point i is represented by R_i:

$$R_i = \sum_{n \in N_i} r_{ni}, \quad i \in I \tag{2}$$

Initially, to transfer the data generated from the CPs to each server of fog layer-1 (f^1), fog layer-2 (f^2), or cloud (c) for necessary processing, sufficient bandwidth needs to be allocated from CP i to the responsible server or servers.

Now, according to our assumption of cooperative fog computing, the bandwidth demand generated from CP i can be satisfied by the available bandwidth of its own region, another region's layer-1 fog, or the layer-2 fog, or by the cloud. Therefore, the total amount of bandwidth allocation can be considered as sum of the allocated bandwidth to establish

TABLE 2. Notations used in formulation.

Constants:	
Symbol	**Definition**
f^1, F^1	Index, set of servers in fog layer-1
f^2, F^2	Index, set of servers in fog layer-2
c, C	Index, set of servers in cloud
s, S	Index, set of all available servers which is the combination of servers in fog layer-1, 2, and cloud
i, I	Index, set of cluster points (CP)
n, N_i	Index, set of sensor nodes under cluster point
P^s_{ip}	Set of paths from CP i to server s
h_{ni}	Bandwidth demand generated by sensor node n at CP i
r_{ni}	Resource demand generated by sensor node n from CP i
H_i	Aggregated bandwidth demand generated by all sensor nodes from CP i
R_i	Aggregated Resource demand generated by all sensor nodes from CP i
ϵ^s_l	Per unit cost of bandwidth consumed from link l for requests served by server s
ρ_l	Available capacity on link l
a_s	Capacity of server s
δ^s_{ipl}	Link-path indicator: 1 if path p which is set up from CP i to server s uses link l in order to satisfy demand of CP i by server s, 0 otherwise
α, β, γ	Weight parameters related to three optimization objectives
Variables:	
Symbol	**Definition**
$y_i^{f^1}$	Bandwidth allocation from CP i to fog layer-1 server f^1
$y_i^{f^2}$	Bandwidth allocation from CP i to fog layer-2 server f^2
y_i^c	Bandwidth allocation from CP i to cloud server c
y_i^s	Bandwidth allocation for traffic from CP i to server s
x^s_{ip}	Bandwidth allocation in path p, if traffic from CP i to server s uses path p
z^s_{il}	Total bandwidth demand from link l for requests generated from CP i and served by server s
u	Max. utilization of all links
$g_i^{f^1}$	Server's resource allocation from fog layer-1 server f^1 for CP i
$g_i^{f^2}$	Server's resource allocation from fog layer-2 server f^2 for CP i
g_i^c	Server's resource allocation from cloud server c for CP i
g_i^s	Server's resource allocation from server s for CP i
k	Max. utilization of all servers

paths from CPs to the servers of all these layers. These constraints are represented by (3) and (4):

$$\sum_{s\in S} y_i^s = \sum_{f^1\in F^1} y_i^{f^1} + \sum_{f^2\in F^2} y_i^{f^2} + \sum_{c\in C} y_i^c, \quad i\in I \tag{3}$$

$$\sum_{s\in S} y_i^s = H_i, \quad i\in I \tag{4}$$

The bandwidth that is allocated to a particular path from CP i to server s is given by using the path flow variables x^s_{ip}:

$$\sum_{p\in P^s_{ip}} x^s_{ip} = y^s_i, \quad i\in I, s\in S \tag{5}$$

If any bandwidth is allocated on particular path p to satisfy a portion of the request of bandwidth demand H_i from any CP i, then all the links associated with that path have to carry that portion of demand H_i. Therefore, we can determine the flow on each link l for all paths from i to s:

$$\sum_{p\in P^s_{ip}} \delta^s_{ipl} x^s_{ip} = z^s_{il}, \quad l\in L,\ i\in I,\ s\in S \tag{6}$$

The total amount of bandwidth required on link l must not exceed the capacity of that link times the maximum utilization variable of any link. This constraint is required to ensure link level load balancing.

$$\sum_{i\in I}\sum_{s\in S} z^s_{il} \le \rho_l u, \quad l\in L \tag{7}$$

Note that the maximum utilization of any link cannot be more than 1 at any point:

$$0 \le u \le 1. \tag{8}$$

Similar to (3), the resource demand generated from each region i can be satisfied by the available servers' resources (CPU processing capacity) of the servers within own region and other regions of layer-1 fog f^1, layer-2 fog f^2, or by cloud c. Therefore, total amount of servers' resources available is the sum of servers' resources of fog layer-1, fog layer-2, and cloud.

$$\sum_{s\in S} g_i^s = \sum_{f^1\in F^1} g_i^{f^1} + \sum_{f^2\in F^2} g_i^{f^2} + \sum_{c\in C} g_i^c, \quad i\in I \tag{9}$$

Now, the total amount of resource demand required can be split into all available servers:

$$\sum_{s\in S} g_i^s = R_i, \quad i\in I \tag{10}$$

Constraint (11) is used to ensure a proportional allocation between servers' resource and bandwidth. This indicates if more computational capacity is used from server s to satisfy computational demand generated from CP i, then, more bandwidth will be allocated from that CP i to that server s.

$$H_i g_i^s = R_i y_i^s, \quad i\in I,\ s\in S \tag{11}$$

The servers' resource requirement from all CPs $i \in I$ to server s must not exceed the available resources of that server times the maximum utilization of any server. This constraint is used to balance the load among different servers.

$$\sum_{i\in I} g_i^s \le a_s k, \quad s\in S \tag{12}$$

Since, the maximum utilization of server s cannot be more than 1, we have

$$0 \le k \le 1. \tag{13}$$

B. OBJECTIVE FUNCTION

There are three goals of this work: (i) to minimize bandwidth cost of routing, (ii) to minimize maximum link utilization of network links, and (iii) to minimize the maximum server resource utilization. These goals are presented with the composite objective function:

$$\min \alpha \sum_{s \in S} \sum_{i \in I} \sum_{l \in L} \epsilon_l^s z_{il}^s + \beta u + \gamma k \tag{14}$$

Here, by using three different weight factors α, β, γ and varying their values for each of the three parts in the objective function, we can change the priority associated with each part. Secondly, we can change the unit cost ϵ_l^s to properly reflect the bandwidth cost (and thus indirectly, the delay cost) depending on the location of the link and server in the three-layer architecture. In summary, the goal of the optimization problem is to minimize (14) subject to the constraints (3) to (13).

IV. SIMULATION STUDY SETUP AND RESULT ANALYSIS

To analyze this composite optimization problem, we used the fog computing architecture having layer-1 fog, layer-2 fog, and cloud nodes as shown in Fig. 1. We consider that a portion of cloud's computational resources is used to support this cooperative fog computing network. To emulate this setup, we assume that maximum number of servers used from cloud by the service provider of this cooperative fog computing network is fixed (due to monetary constraint). However, since the cloud services can be used on demand as utility, the number of servers used will vary based on the requirement to reduce cost, but it will not exceed the maximum number. For simplicity, we also consider the capacity of all available servers from each layer as the same. Thus, we vary the available computational capacity of each layer by varying the number of available servers.

The variation in the value of ε_l^s is used to emulate variable bandwidth cost to reach servers of different layers. The number of links/hops required can emulate the distance to reach servers of a layer. The combination of ε_l^s and number of links required to reach a server is used to design an environment. In this environment, the cost to reach a server of own region in layer-1 fog, other regions' layer-1 fog, layer-2 fog, and cloud servers will be in the increasing order. The value of the topology related parameters used in this study are presented in detail in Table 3. Layer-1 fog is the nearest layer from the edge or the IoT devices. Furthermore, layer-1 fog is divided into different regions and thus, resource distribution in this layer follows distributed nature. Each region is primarily responsible to mitigate the demand generated from its IoT devices. Therefore, to emulate a practical two-layer fog computing environment, it is reasonable to consider the lowest capacity within a region of layer-1 fog. Then, we increase the capacity in layer-2 fog as it is often considered as mini cloud (cloudlet) in the literature [11]. Finally, we consider highest capacity in the cloud layer as it can be purchased on demand by the fog service provider. In practice, compared to the fog layer, the capacity of cloud is very high.

TABLE 3. Topology related parameters.

# of CPs	4 (1 in each region)	
Max. # of available servers in each layer	Fog Layer-1	4 (each region)
	Fog Layer-2	8
	Cloud	40
# of links/hops required to establish path from each CP	Fog Layer-2	10
	Cloud	20
Per unit cost of bandwidth consumed from each link, ε_l^s	$\varepsilon_l^{f^1}$	1
	$\varepsilon_l^{f^2}$	2
	ε_l^c	5
Links' capacity (in Mbps)	Fog Layer-1	100
	Fog Layer-2	200
	Cloud	1000
Capacity of each server (in GHz)	2.5	

We used AMPL/CPLEX (v 12.6.0.0) as the tool to solve the MILP program that minimizes (14) subject to the constraints (3) to (13). For the cases we investigated, CPLEX required around 20 ms on an average to solve the MILP model each time. We also used shell scripting for result analysis. We conducted the experiment on HP ProBook 450 G4 Notebook PC with Linux OS, having Intel Core i5 2.5 GHz processor and 8 GB RAM.

We conducted several case studies. These help to investigate how the amount of network and resource support provided from the links and servers of different fog layers, and cloud vary with the change of priority in bandwidth cost minimization and load balancing. We also studied this problem considering that both homogeneous and heterogeneous type of demand are generated from the cluster points. All the cases studied in this paper are summarized in Table 4. In the homogeneous condition, the amount of demand generated from each CP is same while in the heterogeneous condition, the amount of demand generated from each CP can be different. The bandwidth and resource demand sets used for homogeneous and heterogeneous conditions are presented in Table 5 and 6, respectively.

A. HOMOGENEOUS DEMAND SETS

1) CASE-1 (CHANGES IN BANDWIDTH COST)

Fig. 2 shows the changes of bandwidth cost with α, i.e., priority to minimize the bandwidth cost. The figure is drawn in logarithmic scale considering the amount of demands generated from each CP as same (Homogeneous Demand). One of the key findings in this case is that when lower priority is given on bandwidth cost minimization (small values of α), the bandwidth demand is distributed among all the possible destinations, i.e., layer-1 fog (own or neighbor regions), layer-2 fog, and cloud nodes. This incurs the highest bandwidth cost. However, as α increases, a sequence of changes is

TABLE 4. List of cases investigated in this work.

Conditions	Cases
Homogeneous	Case-1: How bandwidth cost changes as the priority on bandwidth cost minimization weight factor (α) increases.
	Case-2: How maximum and average link utilization changes as the priority on link level load balancing (β) increases.
	Case-3: How maximum server resource utilization and average number of servers used changes as priority on server level load balancing (γ) increases.
	Case-4: Effect of bandwidth demand on bandwidth cost and average link utilization.
	Case-5: Effect of resource demand on average server resource utilization.
Heterogeneous	Case-1: How bandwidth cost changes as the priority on bandwidth cost minimization weight factor (α) increases.
	Case-2: How maximum and average link utilization changes as the priority on link level load balancing (β) increases.
	Case-3: How maximum server resource utilization and average number of servers used changes as priority on server level load balancing (γ) increases.

TABLE 5. Homogeneous case: Bandwidth and resource demand.

Bandwidth Demand (Mbps)				
Notation	CP1	CP2	CP3	CP4
h1	50	50	50	50
h2	100	100	100	100
h3	150	150	150	150
h4	200	200	200	200
Resource Demand (GHz)				
Notation	CP1	CP2	CP3	CP4
r1	5	5	5	5
r2	10	10	10	10
r3	15	15	15	15
r4	20	20	20	20

TABLE 6. Heterogeneous case: Bandwidth and resource demand.

Bandwidth Demand (Mbps)				
Notation	CP1	CP2	CP3	CP4
h1	1350	150	150	150
h2	1050	450	150	150
h3	850	650	150	150
h4	650	600	450	150
h5	450	450	450	450
Resource Demand (GHz)				
Notation	CP1	CP2	CP3	CP4
r1	60	8	8	8
r2	50	18	8	8
r3	40	28	8	8
r4	30	28	18	8
r5	21	21	21	21

observed in terms of service provisioning. At first, the cloud layer which is considered most costly due to its distance from the CPs, is discarded from the bandwidth demand mitigation list. Further increments of α removes layer-2 fog nodes. In the next sequence, the demand is satisfied only through the own region of demand generator and its neighbor regions which are resided in layer-1 fog. In the last sequence, the demand is mitigated solely from its own region, if bandwidth demand is less than or equal to the capacity of its own region.

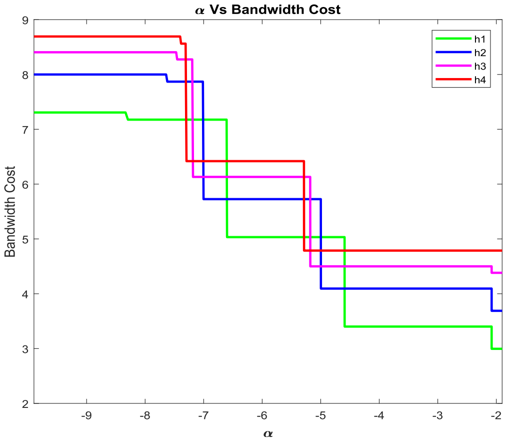

FIGURE 2. α versus bandwidth cost.

Besides these, it can be noticed that the lowest amount of bandwidth demand (50 unit here) results in the lowest bandwidth cost with a maximum number of transition points. These transitions are due to the distribution of load among different fog layers and cloud for different values of α. In contrary, the highest amount of bandwidth demand incurs the highest bandwidth cost with a minimum number of transition points. This indicates that the more closer the demand to the capacity, the less options are available to minimize the cost.

2) CASE-2 (MAXIMUM AND AVERAGE LINK UTILIZATION)

Fig. 3 shows how the maximum link utilization varies with β. When the value of β is small meaning that lowest priority is given on link level utilization, the highest value of maximum link utilization is observed. From our investigation, we can determine the mechanism behind it. With lower values of β, the demand is mitigated from the nearest providers of demand generator CPs, which results in the highest value of maximum link utilization. Thereafter, with continuous increments of β, bandwidth demands are distributed among neighbor regions of layer-1 fog, layer-2 fog, and cloud sequentially, and thus,

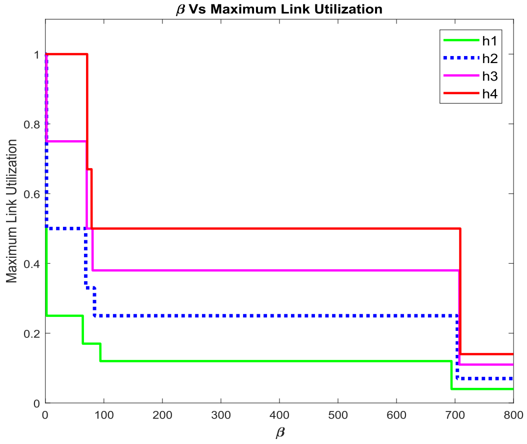

FIGURE 3. β **versus maximum link utilization.**

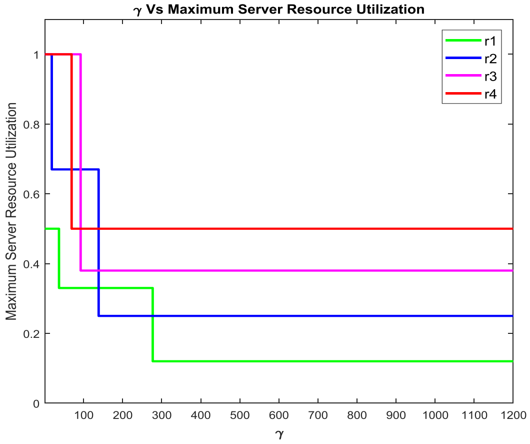

FIGURE 5. γ **versus maximum server resource utilization.**

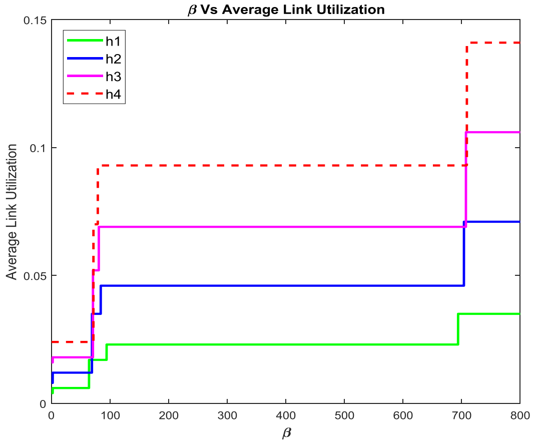

FIGURE 4. β **versus average link utilization.**

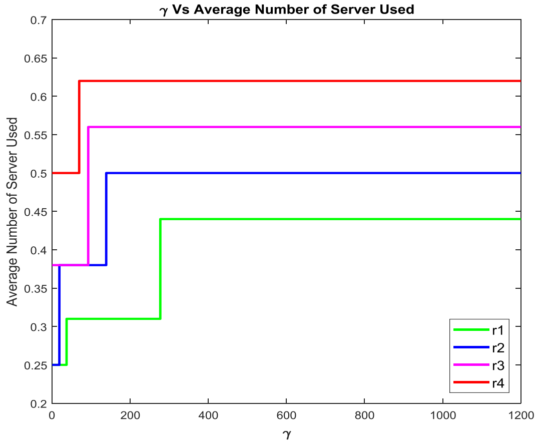

FIGURE 6. γ **versus average number of server used.**

the maximum link utilization decreases. This trend is just opposite of the trend of α vs. the bandwidth cost. Furthermore, for lower bandwidth demand, there are more options to distribute that demand among different layers and therefore, the number of transitions is higher. As the bandwidth demand increases, the links start to be highly utilized. However, by choosing an appropriate value of β, the maximum link utilization can be reduced while keeping the bandwidth cost within a tolerable limit.

Fig. 4 depicts that as the value of β is increased, the average link utilization also increases. The reason behind this is that as the priority on β increases, the loads are distributed across all the possible paths, and thus the average link utilization rises. Furthermore, when all the cluster points have lowest (50 units) bandwidth demand, the average link utilization curve has more number of transitions compared to the maximum bandwidth demand (200 unit) used. This happens as for lowest bandwidth demand, there are many options to reduce link utilization. These can be using cloud layer, or layer-2 fog, or layer-1 fog separately or using different combinations of those layers. In contrary, for higher bandwidth demand, there are less options available.

3) CASE-3 (MAXIMUM SERVER RESOURCE UTILIZATION AND AVERAGE NUMBER OF SERVERS USED)

Fig. 5 shows the variation of the maximum server resource utilization with γ. For lower values of γ, i.e., less priority on server level load balancing, the resource demands are mitigated through the servers closer to the sensor nodes. Since the maximum available servers' resources in the bottom layer (i.e., layer-1 fog) of fog-cloud computing architecture is low compared to the top layer (i.e., cloud layer), the servers of the bottom layer tend to be fully utilized. This results in a higher value of maximum server utilization. On the other hand, with the increase in γ the servers' resource demand starts to be distributed among all the available servers of fog layers and cloud. Therefore, the maximum server utilization continues to decrease until it reaches the minimum value.

Next, Fig. 6 depicts the change in the average number of servers used with γ. When γ, i.e., the priority on minimizing maximum server resource utilization, is increased, the loads are distributed among all the available servers. This results in the increase in the average number of servers used. Furthermore, the average number of servers used also increases with the servers' resource demand. This is explained by taking

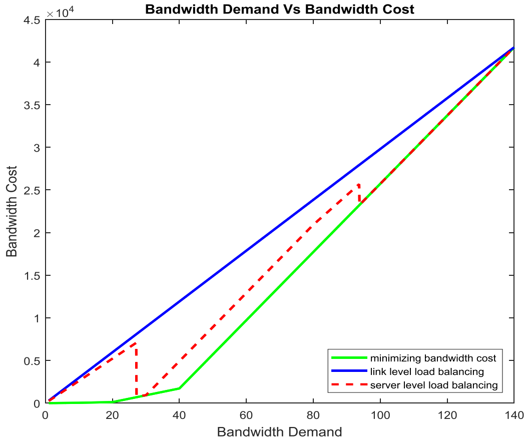

FIGURE 7. **Bandwidth demand versus bandwidth cost.**

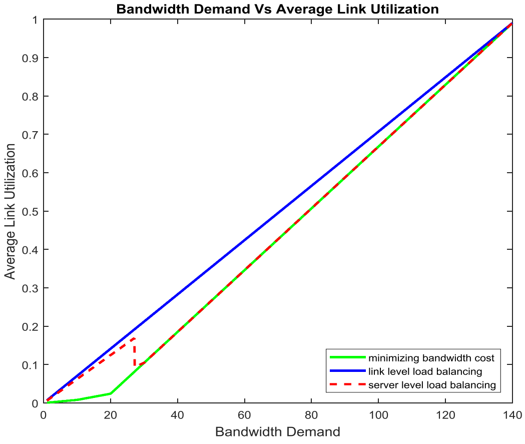

FIGURE 8. **Bandwidth demand versus average link utilization.**

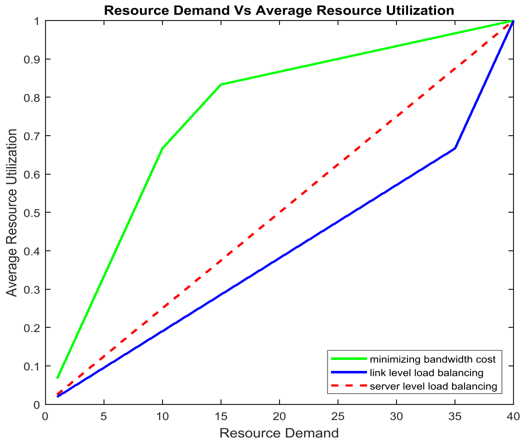

FIGURE 9. **Resource demand versus average server resource utilization.**

four sets of resources demand. For lowest value of demand set, r1, the average number of servers used is minimum and for the highest value of demand set, r4, the average number of servers used is maximum.

4) CASE-4 (EFFECT OF INCREMENT IN BANDWIDTH DEMAND)

Fig. 7 presents how the bandwidth cost changes with the increase in the bandwidth demand for setting different levels of priorities on different components of the objective function. These components are minimizing bandwidth cost, link level load balancing, and server level load balancing. This is a trade-off analysis, which helps the service providers to fine tune the value of weight factors α, β, and γ, based on the requirements. The bandwidth cost is minimum when priority is given on the bandwidth cost minimization and maximum when priority is given on the link level utilization or load balancing. This is due to the fact that when priority is given on link level load balancing, the allocation of bandwidth demand is done as evenly as possible among different links. Thus, the use of more alternate routes is increased, which results in a higher bandwidth cost. Since, server level load balancing does not have any direct impact on the bandwidth cost and thus, the curve due to this objective lies in the middle. It is also noted that the difference in bandwidth cost due to these three parts of the objective function is more visible for lower values of the bandwidth demand.

Fig. 8 shows the change in the average link utilization with the bandwidth demand. In general, as we increase the bandwidth demand, the average link utilization increases. However, we further investigated to understand how the nature of increase in the average link utilization varies due to setting up different levels of priority on different parts of the objective function. When the bandwidth cost minimization has priority, the average link utilization is less compared to other two components of the objective function. The observation here is that when priority is provided on minimizing the bandwidth cost, the bandwidth demand tends to be mitigated from lower level fog layers. Thus, the number of links associated in these case is small and therefore, the average link utilization becomes lower than other priorities. On the other hand, when priority is given on link level load balancing, the maximum average link utilization is observed. This is because the load tends to be distributed among all the possible links to minimize the load of any particular link, which increases the total number of links associated. In case of server level load balancing, the average link utilization remains in the middle similar to bandwidth cost curve shown in Fig. 7.

5) CASE-5 (VARIATION IN RESOURCE DEMAND)

Fig. 9 shows how the the average server resource utilization changes with increasing resource demand for different priorities on different objectives. When minimizing bandwidth cost is the prime focus, the average resource utilization is maximum. This is because in this condition, the load tends to be mitigated at first from the servers closer to sensors, and then destined to the distant servers of fog layer-2 and cloud. However, since the servers closer to sensors have lower resource capacity, they gets over utilized very soon and thus, average server resource utilization also increases. For link level load balancing, the loads are distributed among all possible destinations of the fog layers and cloud. Furthermore, since the resource capacity of upper level fog layer and cloud

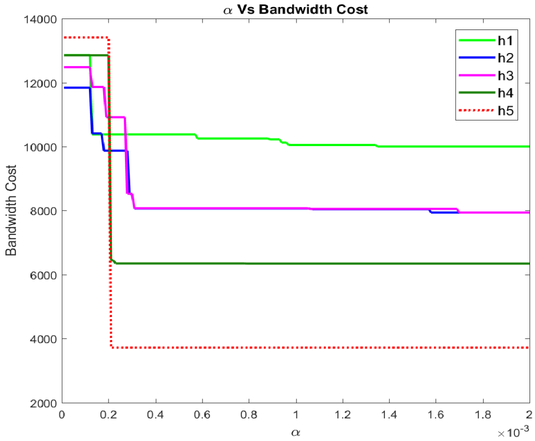

FIGURE 10. *α* **versus bandwidth cost for different demand sets.**

is higher, there is less possibility of the servers being over utilized. In this way, the average resource utilization remains less compared to the other two cases.

B. HETEROGENEOUS DEMAND SETS

We also studied the changes in bandwidth cost, average link utilization, and average server resource utilization for heterogeneous demands. We consider these demands require different amount of aggregated bandwidth and resource generated from different CPs. Here, we also vary the priorities on different parts of the composite objective.

1) CASE-1 (CHANGES IN BANDWIDTH COST)

Fig. 10 shows α (bandwidth cost minimization) versus changes in the bandwidth cost from different level of heterogeneity to homogeneity. To conduct a fair study for all demand sets, we keep the total amount of demand generated from all CPs as same. However, to create different levels of heterogeneity, we vary the amount of demand generated from each CP. This figure indicates that when the demand among four cluster points are same (homogeneous demand set), the bandwidth cost becomes lowest compared to other demand sets. At this condition, the demand is mostly mitigated by layer-1 fog. Since, layer-1 fog is the nearest layer from the CPs in the three layer fog-cloud architecture, thus, homogeneous demand set results in lowest bandwidth consumption. As the difference between the amount of bandwidth increases meaning that level of heterogeneity increases, the bandwidth cost also increases and reaches the maximum value when one of the CPs has the maximum bandwidth demand. This is due to the fact that this demand set uses the distant layers i.e., layer-2 fog and cloud more in comparison with other demand sets to mitigate the bandwidth demand.

2) CASE-2 (MAXIMUM AND AVERAGE LINK UTILIZATION)

Fig. 11 and Fig. 12 represent how the maximum link utilization and the average link utilization vary with the increase of β, respectively, for different levels of heterogeneity in bandwidth demand sets. Initially, from these analyses, it can be stated that the maximum link utilization

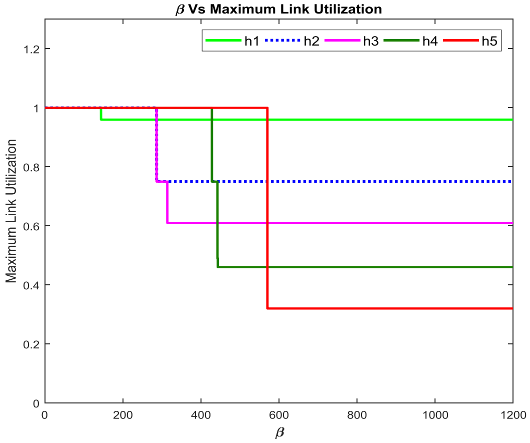

FIGURE 11. *β* **versus maximum link utilization for different demand sets.**

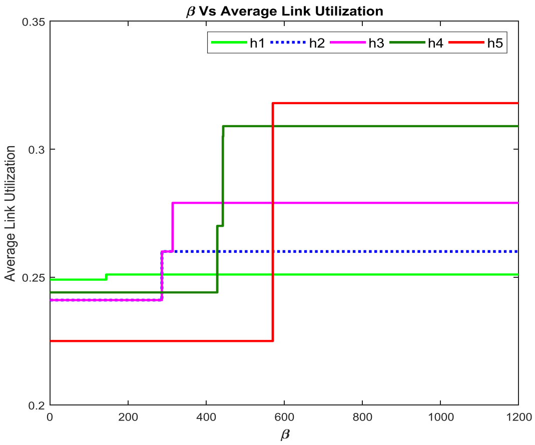

FIGURE 12. *β* **versus average link utilization for different demand sets.**

and the average link utilization decrease as we increase β, which is similar to our previous findings. However, from this study, we can also observe that as we continue to increase the level of heterogeneity from homogeneity of the demand set, the variation in the value of both the maximum and the average link utilization reduces. For the highest level of heterogeneity in demand set, this variation is the lowest.

For both maximum and average link utilization, $h1$ (highest level of heterogeneity) does not have room for large variations whereas for $h5$ (homogeneous demand set), we notice the highest level of transition. $h1$ needs to use upper layers even for the lower value of β. However, at the lower level of heterogeneity, the demand is initially satisfied by the own region of layer-1 fog for the lower value of β. Then, as we increase β, the load starts to be distributed among all fog layers and cloud. Therefore, the larger variation in the value of the maximum and the average link utilization is observed for lower levels of heterogeneity.

3) CASE-3 (MAXIMUM SERVER RESOURCE UTILIZATION AND AVERAGE NUMBER OF SERVERS USED)

Fig. 13 shows γ versus the maximum server resource utilization. As γ increases, the maximum server resource utilization

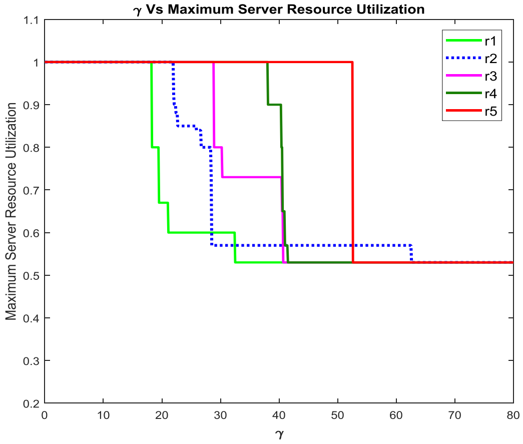

FIGURE 13. γ **versus maximum server resource utilization.**

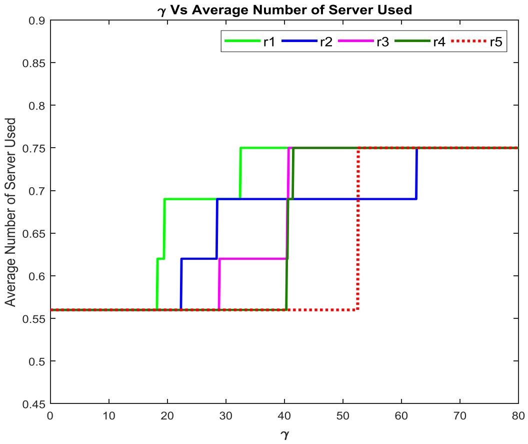

FIGURE 14. γ **versus average number of server used.**

decreases. When γ increases, i.e., priority is given on the minimization of maximum server utilization, this forces the loads to be distributed among different layers and thereafter the maximum server resource utilization decreases. For resource demand set $r1$ (when all the cluster points have same resource demand), the maximum resource utilization reaches the lowest value with the lower value of γ as compared to the resource demand $r5$ (when one cluster point have a large amount of resource demand than other cluster points).

Fig. 14 shows the changes of the average number of server used with γ. As γ is increased, the average number of server used increases as the load is distributed among different layers. For resource demand set $r1$, the average number of server used reaches the maximum value at the lower value of γ as compared to the resource demand $r5$. This can be considered as just the opposite of γ versus the maximum resource utilization case.

C. KEY OBSERVATIONS

- When the weight factor associated with bandwidth cost minimization (α) is the highest, the demand is satisfied by the demand's own region only (assuming sufficient capacity is available). Thus, the lowest bandwidth cost is ensured. As we continue to decrease α compared to the other weight factors, the demand starts to be distributed. Initially, it goes to the nearby region of layer-1 fog, then, layer-2 fog, and finally, to the cloud. Thus, with the lowest value of α, the demand is distributed to all possible destinations, which results in the highest bandwidth cost. This illustrates how the cooperative three layer fog-cloud computing system works.
- When the weight factor to prioritize link level load balancing (β) is relatively the highest, the load is distributed and with the lowest value of β, the demand is satisfied in its own region.
- As we increase the value of β, the maximum link utilization decreases while the average link utilization increases. This indicates, with the increase in β, the use of alternate routes increases.
- The maximum server resource utilization decreases and the average number of server used increases, with the increase in γ.
- For the heterogeneous resource demand: as the level of heterogeneity increases, the variation in range of minimum and maximum value of the bandwidth cost with α, the maximum link utilization and the average link utilization with β decreases.
- For the heterogeneous resource demand, as the level of heterogeneity increases, the maximum server resource utilization reaches the lowest value and the average number of servers used reaches the highest value for lower value of γ.

V. CONCLUSION

In this paper, minimization of bandwidth cost and efficient resource management in a cooperative three-layer fog-cloud computing environment is studied. We first presented a novel MILP optimization formulation in the three-layer fog-cloud computing environment. We then investigated several scenarios to evaluate how efficient utilization of network and server resources can be ensured in such an environment by leveraging SDN. Both homogeneous and heterogeneous network and server resource demands generation from CPs are considered. Then, the variation in performance is analyzed in terms of bandwidth cost, links' and servers' utilization, and the number of servers used. To the best of our knowledge, no other work has considered the bandwidth cost minimization and link and server level load balancing jointly in a cooperative three-layer fog-cloud computing environment. Furthermore, by tuning three weight factors associated in the composite objective function, i.e., bandwidth cost, link level utilization, and server resource utilization, the priority level can be controlled. This gives the service provider to choose which component of the objective function should get more priority based on the situation in the network. Our designed framework also provides the opportunity to change this priority level time to time due to the temporal and spatial variation in requirements. Thereafter, this work can help the fog service providers to allocate the limited resources effectively. Furthermore, our optimization model can be used as an important benchmark to compare the performance of any fast solution heuristic. Also, this model can be used to evaluate any fog computing topology.

In future, we plan to extend this work in a number of ways. We plan to study this framework in large scale using a heuristic that can be used in real time. We also want to incorporate server and network consolidation using virtualization techniques in a dynamic traffic environment to achieve further improvement in resource provisioning. A priority based request processing and resource allocation can also be studied. This can help to understand if the requests generated from one CP gets priority, how much improvement is achieved in service provisioning of prioritized CP and how other CPs are affected in this process. We also plan to investigate how the QoS can be ensured through traffic classification and therefore, a study can be conducted considering throughput guarantee as a requirement for delay sensitive services.

REFERENCES

[1] P. Middleton, P. Kjeldsen, and J. Tully, "Forecast: The Internet of Things, worldwide, 2013," Gartner Res., Stamford, CT, USA, Tech. Rep. G00259115, 2013, p. 57.

[2] *Internet of Things—Number of Connected Devices Worldwide 2015–2025*. Accessed: Jul. 1, 2020. [Online]. Available: https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/

[3] O. Osanaiye, S. Chen, Z. Yan, R. Lu, K.-K.-R. Choo, and M. Dlodlo, "From cloud to fog computing: A review and a conceptual live VM migration framework," *IEEE Access*, vol. 5, pp. 8284–8300, 2017.

[4] E. M. Tordera, X. Masip-Bruin, J. Garcia-Alminana, A. Jukan, G.-J. Ren, J. Zhu, and J. Farre, "What is a fog node a tutorial on current concepts towards a common definition," 2016, *arXiv:1611.09193*. [Online]. Available: http://arxiv.org/abs/1611.09193

[5] M. Zahid, N. Javaid, K. Ansar, K. Hassan, M. K. Khan, and M. Waqas, "Hill climbing load balancing algorithm on fog computing," in *Proc. Int. Conf. P2P, Parallel, Grid, Cloud Internet Comput.* Cham, Switzerland: Springer, 2018, pp. 238–251.

[6] X. He, Z. Ren, C. Shi, and J. Fang, "A novel load balancing strategy of software-defined cloud/fog networking in the Internet of vehicles," *China Commun.*, vol. 13, no. 2, pp. 140–149, 2016.

[7] A. J. Kadhim and S. A. Hosseini Seno, "Maximizing the utilization of fog computing in Internet of vehicle using SDN," *IEEE Commun. Lett.*, vol. 23, no. 1, pp. 140–143, Jan. 2019.

[8] R. Buyya and S. N. Srirama, *Fog and Edge Computing: Principles and Paradigms*. Hoboken, NJ, USA: Wiley, 2019.

[9] Y. Jiang, Z. Huang, and D. H. K. Tsang, "Challenges and solutions in fog computing orchestration," *IEEE Netw.*, vol. 32, no. 3, pp. 122–129, May 2018.

[10] R. Kumar Naha, S. Garg, and A. Chan, "Fog computing architecture: Survey and challenges," 2018, *arXiv:1811.09047*. [Online]. Available: http://arxiv.org/abs/1811.09047

[11] Y. Liu, J. E. Fieldsend, and G. Min, "A framework of fog computing: Architecture, challenges, and optimization," *IEEE Access*, vol. 5, pp. 25445–25454, 2017.

[12] S. Yi, C. Li, and Q. Li, "A survey of fog computing: Concepts, applications and issues," in *Proc. Workshop Mobile Big Data*, 2015, pp. 37–42.

[13] M. Mukherjee, R. Matam, L. Shu, L. Maglaras, M. A. Ferrag, N. Choudhury, and V. Kumar, "Security and privacy in fog computing: Challenges," *IEEE Access*, vol. 5, pp. 19293–19304, 2017.

[14] F. A. Kraemer, A. E. Braten, N. Tamkittikhun, and D. Palma, "Fog computing in healthcare–A review and discussion," *IEEE Access*, vol. 5, pp. 9206–9222, 2017.

[15] A. Kumari, S. Tanwar, S. Tyagi, and N. Kumar, "Fog computing for healthcare 4.0 environment: Opportunities and challenges," *Comput. Electr. Eng.*, vol. 72, pp. 1–13, Nov. 2018.

[16] Y. Chen, E. Sun, and Y. Zhang, "Joint optimization of transmission and processing delay in fog computing access networks," in *Proc. 9th Int. Conf. Adv. Infocomm Technol. (ICAIT)*, Nov. 2017, pp. 155–158.

[17] X. Meng, W. Wang, and Z. Zhang, "Delay-constrained hybrid computation offloading with cloud and fog computing," *IEEE Access*, vol. 5, pp. 21355–21367, 2017.

[18] H. A. Khattak, H. Arshad, S. U. Islam, G. Ahmed, S. Jabbar, A. M. Sharif, and S. Khalid, "Utilization and load balancing in fog servers for health applications," *EURASIP J. Wireless Commun. Netw.*, vol. 2019, no. 1, p. 91, Dec. 2019.

[19] X. Xu, S. Fu, Q. Cai, W. Tian, W. Liu, W. Dou, X. Sun, and A. X. Liu, "Dynamic resource allocation for load balancing in fog environment," *Wireless Commun. Mobile Comput.*, vol. 2018, pp. 1–15, Apr. 2018.

[20] N. Mostafa, "Cooperative fog communications using a multi-level load balancing," in *Proc. 4th Int. Conf. Fog Mobile Edge Comput. (FMEC)*, Jun. 2019, pp. 45–51.

[21] Y. Dong, C. Han, and S. Guo, "Joint optimization of energy and QoE with fairness in cooperative fog computing system," in *Proc. IEEE Int. Conf. Netw., Archit. Storage (NAS)*, Oct. 2018, pp. 1–4.

[22] V. Mushunuri, A. Kattepur, H. K. Rath, and A. Simha, "Resource optimization in fog enabled IoT deployments," in *Proc. 2nd Int. Conf. Fog Mobile Edge Comput. (FMEC)*, May 2017, pp. 6–13.

[23] R. Deng, R. Lu, C. Lai, T. H. Luan, and H. Liang, "Optimal workload allocation in fog-cloud computing toward balanced delay and power consumption," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 1171–1181, Dec. 2016.

[24] S. Ningning, G. Chao, A. Xingshuo, and Z. Qiang, "Fog computing dynamic load balancing mechanism based on graph repartitioning," *China Commun.*, vol. 13, no. 3, pp. 156–164, Mar. 2016.

[25] H. Wadhwa and R. Aron, "Fog computing with the integration of Internet of Things: Architecture, applications and future directions," in *Proc. IEEE Int. Conf. Parallel Distrib. Process. Appl., Ubiquitous Comput. Commun., Big Data Cloud Comput., Social Comput. Netw., Sustain. Comput. Commun. (ISPA/IUCC/BDCloud/SocialCom/SustainCom)*, Dec. 2018, pp. 987–994.

[26] C. Wobker, A. Seitz, H. Mueller, and B. Bruegge, "Fogernetes: Deployment and management of fog computing applications," in *Proc. IEEE/IFIP Netw. Operations Manage. Symp. (NOMS)*, Apr. 2018, pp. 1–7.

[27] A. Brogi and S. Forti, "QoS-aware deployment of IoT applications through the fog," *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1185–1192, Oct. 2017.

[28] S. Yi, Z. Hao, Z. Qin, and Q. Li, "Fog computing: Platform and applications," in *Proc. 3rd IEEE Workshop Hot Topics Web Syst. Technol. (HotWeb)*, Nov. 2015, pp. 73–78.

[29] O. Salman, I. Elhajj, A. Chehab, and A. Kayssi, "IoT survey: An SDN and fog computing perspective," *Comput. Netw.*, vol. 143, pp. 221–246, Oct. 2018.

[30] M. Al-khafajiy, T. Baker, H. Al-Libawy, A. Waraich, C. Chalmers, and O. Alfandi, "Fog computing framework for Internet of Things applications," in *Proc. 11th Int. Conf. Develop. eSystems Eng. (DeSE)*, Sep. 2018, pp. 71–77.

[31] M. M. S. Maswood and D. Medhi, "Optimal connectivity to cloud data centers," in *Proc. IEEE 6th Int. Conf. Cloud Netw. (CloudNet)*, Sep. 2017, pp. 1–6.

[32] M. M. S. Maswood, C. Develder, E. Madeira, and D. Medhi, "Dynamic virtual network traffic engineering with energy efficiency in multi-location data center networks," in *Proc. 28th Int. Teletraffic Congr. (ITC)*, vol. 1, Sep. 2016, pp. 10–17.

MIRZA MOHD SHAHRIAR MASWOOD (Member, IEEE) received the B.Sc. degree in electronics and communication engineering from the Khulna University of Engineering & Technology (KUET), Bangladesh, in 2010, and the M.Sc. degree in electrical engineering and the Ph.D. degree in telecommunications and computer networking from the University of Missouri–Kansas City (UMKC), USA, in 2015 and 2018, respectively. During his Ph.D. at UMKC, he worked as a Graduate Research Assistant. He is currently an Assistant Professor with the Department of Electronics and Communication Engineering, KUET. He has served as a Reviewer for the *Journal of Network and Systems Management*. He reviewed more than 40 peer-reviewed conference and journal articles. His research interests include data center optimization, traffic engineering, cloud computing, fog computing, and machine learning.

MD. RAHINUR RAHMAN (Member, IEEE) received the B.Sc. degree in electronics and communication engineering from the Khulna University of Engineering & Technology, Bangladesh, in 2016, where he is currently pursuing the M.Sc. degree with the Department of Electronics and Communication Engineering. He is also working as an Assistant Maintenance Engineer with the Department of ICT Infrastructure Maintenance and Management, Bangladesh Bank (The Central Bank of Bangladesh). His research interests include future networking, the IoT, and fog computing.

ABDULLAH G. ALHARBI (Member, IEEE) received the B.Sc. degree in electronics and communications engineering from Qassim University, Saudi Arabia, in 2010, and the master's and Ph.D. degrees in electrical engineering from the University of Missouri–Kansas City, USA, in 2014 and 2017, respectively. From 2010 to 2012, he was an Electrical Engineer with Saudi Aramco Company. He is currently an Assistant Professor with the Electrical Engineering Department, Al-Jouf University, Saudi Arabia. He has authored and coauthored over 45 journal and conference papers and one Springer book chapter. His research interests include digital IC design, memristor-based circuits, traffic engineering, cloud computing, and network routing. He is a member of the Gulf Engineering Union, the IEEE Circuits and Systems Society, the IEEE Young Professionals, the IEEE Signal Processing Society, the IEEE Instrumentation and Measurement Society Membership, and the IEEE Communications Society Membership, and a Professional Member of ACM. He was a recipient of several awards from Saudi Arabian Cultural Mission in USA, and the University of Missouri–Kansas City.

DEEP MEDHI (Fellow, IEEE) received the B.Sc. degree in mathematics from the Cotton College, Gauhati University, India, the M.Sc. degree in mathematics from the University of Delhi, India, and the Ph.D. degree in computer sciences from the University of Wisconsin–Madison, USA. He was a Member of the Technical Staff at the AT&T Bell Laboratories, in 1989. He has served as an invited Visiting Professor at the Technical University of Denmark, and a Visiting Research Fellow at the Lund Institute of Technology, Sweden, and the State University of Campinas, Brazil. As a Fulbright Senior Specialist, he was a Visitor at Bilkent University, Turkey, and Kurukshetra University, India. He is currently the Curators' Distinguished Professor with the Department of Computer Science Electrical Engineering, University of Missouri–Kansas City, USA, and an Honorary Professor with the Department of Computer Science & Engineering, Indian Institute of Technology–Guwahati, India. He has published over 150 articles, and is the coauthor of the books, *Routing, Flow, and Capacity Design in Communication and Computer Networks* (2004) and *Network Routing: Algorithms, Protocols, and Architectures* (1st edition in 2007, and 2nd edition in 2017), published by Morgan Kaufmann Publishers, an imprint of Elsevier Science. His research interests are multi-layer networking, network virtualization, data center optimization, network routing, design, and survivability, and video quality-of-experience. His research has been funded by NSF and DARPA. He is the Editor-in-Chief of Springer's *Journal of Network and Systems Management* and serves (or served) on the editorial board of the IEEE/ACM Transactions on Networking, the IEEE Transactions on Network and Service Management, the IEEE Communications Surveys and Tutorials, *Telecommunications Systems*, *Computer Networks*, and the IEEE Communications Magazine.