Scalable Design and Dimensioning of Fog-Computing Infrastructure to Support Latency-Sensitive IoT Applications

Ismael Martinez, Abdallah Jarray, and Abdelhakim Senhaji Hafid

***Abstract*—The fog-computing paradigm has appeared as a geo distributed response to a growing focus on latency-sensitive Internet-of-Things (IoT) applications and the long delay that may be provided by cloud data centers. Although many researchers have investigated how IoT can interact with a fog, very few have tackled the question of how to construct a fog infrastructure for the expected IoT traffic. This article addresses the design and dimensioning of a fog infrastructure via a mixed-integer linear program (MILP) to construct a physical fog network design by mapping IoT virtual networks to dimensioned fog nodes. Due to the exponential nature of this MILP formulation, we also propose a column generation model with near-optimal results at a significantly reduced design and dimensioning cost. The numerical results show the viability of the column generation method in its proximity to the optimal solution and in its reasonable solution time.**

***Index Terms*—Column generation, design and dimensioning, fog computing, Internet of Things (IoT), mixed-integer linear programming (MILP).**

I. Introduction

As Internet-of-Things (IoT) devices become more prominent in all facets of the industry, the number of IoT applications requiring real-time processing is also increasing. Heart monitors and vehicular traffic status sharing are examples of IoT applications that are increasing in number and use, and require near-immediate data transmission, processing, and response by the cloud. Processing through cloud data centers results in unacceptably high latency for IoT application responses due to the centralized and distant nature of these data centers. A report by Cisco emphasizes that existing cloud data centers are not designed to meet the current volume and variety of IoT data and requests. Meanwhile, the number of IoT devices is only expected to grow up to 50 billion by the year 2020 [1]; this means that the processing of IoT applications outside the Cloud is crucial.

Manuscript received October 1, 2019; revised January 9, 2020 and February 14, 2020; accepted March 2, 2020. Date of publication March 10, 2020; date of current version June 12, 2020. *(Corresponding author: Ismael Martinez.)*

Ismael Martinez and Abdelhakim Senhaji Hafid are with the Department of Computer Science and Operations Research, University of Montreal, Montreal, QC H3C 3J7, Canada (e-mail: ismael.martinez@umontreal.ca; ahafid@iro.umontreal.ca).

Abdallah Jarray is with the School of Electrical Engineering and Computer Science, University of Ottawa, Ottawa, ON K1N 6N5, Canada (e-mail: ajarray@uottawa.ca).

Digital Object Identifier 10.1109/JIOT.2020.2979705

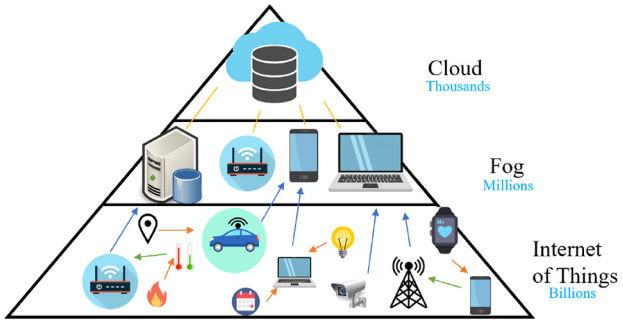

Fig. 1. Fog-DC infrastructure to process IoT requests.

To combat high round-trip latency and increasing data volume to cloud data centers, fog computing has been proposed as highly virtualized micro data centers on the network edge [2], allowing the fog network to analyze and process the most time-sensitive data. Although typically having fewer resources than the Cloud [3], fog nodes are decentralized and geographically distributed enabling IoT connectivity with minimal round-trip latency. Furthermore, the fog network can process many IoT applications that would otherwise be processed in the Cloud, reducing the volume of data reaching the Cloud. A wide-reaching fog network could, therefore, satisfy any IoT process regardless of location.

The increase in IoT devices is a direct result of an acknowledgment of value, proposal, and deployment of IoT applications across many domains, including healthcare [4] and smart cities [4]–[6]. The low latency, scalability, and geodistributed nature of fog have made it integral to the success of these IoT applications [7]. The future growth and adoption of 5G radio access networks further facilitate the viability and implementation of fog networks, and widen the scope of devices that can partake and aid in IoT-fog communication [8].

IoT applications proposed to take advantage of fog include energy management in residential domains [9], fire detection and fire fighting [10], video streaming [11], and video surveillance [12]. There have been proposals for healthcare applications using fog with existing wearable technologies [13], as well as medical-specific wearables to provide real-time-assisted living services in hospitals [14] or to help diagnose heart disease [15].

Most existing contributions [13], [16]–[22] are concerned with fog resource allocation to support the requirements of

IoT applications; however, all these contributions assume an existing fog-computing infrastructure. Other state-of-the-art research in fog computing focus on the conceptual fog infrastructure, and fog interaction with IoT and Cloud. There has been only one study [23] that investigated the design of a fog infrastructure; however, the proposed design was specific to Internet-of-Vehicle (IoV) applications. Indeed, there are no studies on the design and dimensioning of the fog network for the support of the general IoT paradigm. We intend to provide a scalable and efficient approach to the fog design and dimensioning that can be feasibly implemented and deployed.

We consider a geographical region where no fog infrastructure currently exists, and where there is an expected increase in IoT traffic. We define the *design* of the fog infrastructure as the selection of fog node locations from a set of candidate locations, and define the *dimensioning* of each fog node as determining the amount of CPU, memory, and storage resources to be installed. Designing and dimensioning will, therefore, yield a full blueprint necessary for the construction of a fog infrastructure.

To the best of our knowledge, this article is the first work to propose a design and dimensioning scheme of the general fog infrastructure for IoT applications. Our approach is based on the existing or predicted IoT traffic, with the goal to design a fog infrastructure to support the processing of upcoming IoT requests. Furthermore, our approach is extensible, allowing for new fog nodes to be optimally added to a current infrastructure in the event of an increase of IoT traffic.

We consider an infrastructure of the combined fog and Cloud paradigms known as fog-DC (Fig. 1), and optimize the design, dimensioning, and IoT-resource allocation to the fog-DC infrastructure. To optimize the cost of the fog-DC design and dimensioning, two approaches are presented: 1) an exact approach using a mixed-integer linear programming (MILP) method called fog-DC-MILP and 2) a heuristic solution using column generation called fog-DC-CG.

This article outlines the first study of the design and dimensioning of fog computing for IoT applications. Our main contributions are as follows.

1) We perform an analysis of model considerations such as transmission delay from IoT applications and network congestion toward fog-DC that impact the design and dimensioning of the fog infrastructure.
2) We propose an exact optimization approach for the fog-DC design and dimensioning via the one-shot MILP model (fog-DC-MILP) that minimizes the resource and network mapping cost between a fixed IoT set and the constructed fog infrastructure.
3) We propose a scalable heuristic column generation approach (fog-DC-CG) to overcome the computation complexity of fog-DC-MILP. The numerical results compare the efficiency of these two models with standard heuristics, such as a greedy algorithm and a matching-based model.

Devising a scalable method to constructing an optimal fog infrastructure paves the way for the implementation of IoT-fog-Cloud communication. Given the numerous studies into the interaction between IoT, fog, and Cloud reviewed in the next section, physical fog networks will allow for extended and practical research into the efficiency of fog computing.

The remainder of this article is organized as follows. Section II presents related work. Section III defines fog-DC. Section IV presents the impact of IoT traffic, network congestion, and routing delays on the fog-DC model. Section V defines the MILP model. Section VI presents the column generation model. Section VII defines two benchmark heuristic models to compare against our proposed model. Section VIII evaluates and compares the proposed solutions. Section IX concludes this article and presents future work.

II. Related Work

Conceptual research in the fog infrastructure has studied how fog could be set up to interact with IoT devices, and many proposals supporting IoT applications using an existing fog infrastructure have been brought forth. Across different studies, we use IoT *requests* and *applications* interchangeably, and use *tasks* and *modules* interchangeably as components of an IoT application.

Three IoV applications use clusters of slow moving or parked cars as the fog itself; Sookhak *et al.* [24] proposed incentives for participating, such as free parking, free Wi-Fi, or free shopping vouchers, Hou *et al.* [25] showed how non-smart cars may be upgraded with hardware and/or software in order to take part, and Zhou *et al.* [26] allocated vehicular fog resources to users in an information incomplete environment using preference matching algorithms. More generally, Chang *et al.* introduced the concept of the Consumer as a Provider (CaaP), a platform to allow user devices, such as phones and modems to act as fog nodes available for public use [27]. With a large enough user base, the CaaP fog network can cover vast continuous areas. Given an existing fog infrastructure, Souza *et al.* [17] proposed to virtually cluster fog nodes for seamless data sharing within clusters, and data transfers between clusters. Zhang *et al.* [28] proposed CFC-IoV as an IoV application whereby roadside fog nodes handover data between each other for moving vehicles.

Taneja and Davy [29] noted that each IoT request is in fact composed of multiple *modules*, each composed of a single sensor or an actuator. Using heuristic search methods, Xia *et al.* [30] assigned IoT application modules to fog devices. The module assignments reserve resources for the IoT application for all future requests until relinquished. Aazam and Huh [31] developed a pricing model for module assignments based on the probability that an IoT application will relinquish its hold of fog resources. By clustering IoT devices together that run the same service, and mapping those services to fog nodes, Yousefpour *et al.* [32] are able to extend the scope of the IoT devices which can use each assigned modules.

Frameworks over a combination of IoT, fog, and Cloud layers can facilitate resource allocation deployment. *FogBus* [33] is a framework where IoT requests first arrive to broker nodes to be assessed and forwarded to fog or cloud data centers. FogBus also separates fog nodes into computation and storage devices; these extra layers may decrease provisioning

efficiency and functionality. The *Foggy* [34] framework is built using only a single fog layer. Both frameworks require extra software installed in each fog node to achieve intercommunication of available resources and job lists.

Resource provisioning schemes aim to find an acceptable mapping between IoT resources and fog nodes. The objective of these schemes varies from minimizing round-trip latency [16], [30], maximizing fog utility which inherently minimizes Cloud utility [3], [19], or minimizing the resource provisioning cost [13], [35]. Uniquely, Ni *et al.* [36] focused on minimizing the fog credibility score, and Do *et al.* [11] proposed a model that jointly maximizes fog utility and minimizes the generated carbon footprint. Three separate models are proposed by Zhang *et al.* with independent objectives: 1) to minimize energy and delay costs of edge network resource allocation [37]; 2) to minimize IoT request rejection from fog nodes [28]; and 3) to minimize the stochastic delay cost associated with edge computing [38].

Naas *et al.* [16] proposed an integer programming formulation and a resource provisioning heuristic based on geographical zoning that was evaluated using the simulation toolkit iFogSim [39]. Donasollo *et al.* [35] solved an integer program using a divide and conquer approach while minimizing resource allocation cost. Intharawijitr *et al.* [21] proposed the use of 5G mobile networks with fog to minimize the number of rejected IoT requests; an MILP was formulated and solved via a greedy allocation policy. Souza *et al.* [17] also formulated an MILP model over a fog-Cloud infrastructure that was solved directly via Gurobi Optimizer for a limited instance size. Salaht *et al.* [40] used a constraint programming approach to satisfy all QoS requirements of IoT requests for a quicker solution; however, this can result in a large variety of provisioning costs to a user. Skarlat *et al.* [20] proposed a mixed nonlinear programming model that is solved using a genetic algorithm.

Most resource provisioning approaches use the fog alongside the Cloud to process IoT applications the fog cannot [16], [17], [20], while [21] establishes that IoT requests not processed by the fog are not processed at all. The rejection of IoT requests occurs when all fog nodes that could satisfy the IoT latency threshold do not have enough available resources. To avoid this rejection, either a larger fog infrastructure should be implemented, or a *network sink* such as the Cloud should be used to receive and process all IoT applications unable to be processed by the fog infrastructure.

The core limitation of the reviewed research is the assumption that a fog infrastructure exists. To the best of our knowledge, there is only one work that designs a fog infrastructure to support IoV applications [23].

Yu *et al.* [23] proposed a fog deployment and dimensioning scheme for IoV applications, whereby vehicles connect to roadside units (RSUs), which connect to fog nodes in order to receive and provide real-time traffic conditions on all covered areas. Additionally, the optimal location of gateways to access the Cloud is also found. The locations and dimensioning of RSUs, fog and gateway devices are optimally found from a set of candidate locations and dimensioning configurations in an arrangement that minimizes infrastructure cost. Regarding vehicular traffic, the model assumes a known static set of vehicle resources accessing the network across different regions, which may not be true in practice. Although having a set of candidate locations for fog nodes is practical, the model also assumes a finite candidate set of dimensioning configurations; the solution to this model is thus dependent on the completeness of such a set. Finding the optimal placement of RSUs and gateways increase the complexity of the model, while the inclusion of RSUs also restricts the application of this model to IoV.

In this article, we propose a fog design and dimensioning scheme to support a general heterogeneous IoT population for a predicted set of IoT traffic. In place of using an exhaustive finite set of resource dimensions from which we allocate to each fog node, we define the dimensions of each fog node as continuous below its maximum resource capacity. Furthermore, by assuming any fog has network access, we acknowledge that any fog node could access the Cloud, thus removing the need to place gateways.

III. FOG-DC

A. IoT Traffic

We consider the total amount of expected IoT device tasks[1] for a given area made up of sensors and actuators, where each IoT request is composed of one or more of these devices. We use these expected numbers of tasks as well as the frequency of requests to formulate our predicted IoT traffic.

We define a set K of unique request classes,[2] and use this set to build our expected IoT request set N. Each request class $k \in K$ defines a set of IoT applications with identical data transmission and response format, and the number of request tasks q_k; furthermore, we assume that data uploads from requests with class k follow a Poisson process with arrival rate λ_k [22].

Suppose we want our fog-DC design to be built to handle the incoming traffic the majority of the time; we define $\rho \in [0.5, 1)$ to be the percentile of the IoT traffic we wish to cover. For each IoT request class $k \in K$ with arrival rate λ_k, we define the ρ-percentile as

$$x_k^\rho = \min\left\{x_k \in \mathbb{Z}^+ : \mathcal{P}(x_k; \lambda_k) \geq \rho\right\}, \quad k \in K \tag{1}$$

where $\mathcal{P}(x; \lambda)$ is the cumulative distribution function of the Poisson distribution with parameter λ. Let k_m represent the mth request of class k within a larger set. We define the ρ-percentile set of IoT requests as

$$N_\rho = \bigcup_{k \in K} \bigcup_{m=1}^{x_k^\rho} k_m \tag{2}$$

where $\bigcup$ defines the union of sets. By the construction of N_ρ in (2), we observe a mapping between N_ρ and K such that for each request $n \in N_\rho$, there exists a class $k \in K$ associated with n. We define this mapping as $M_K : N_\rho \mapsto K$. If $M_K(n) = k$,

[1] These estimates may come from estimations of the planned manufacturing and business expansion of the given area.

[2] Multiple requests executed from the same set of devices, with the same transmitted data format and expected response format at different times are considered to follow the same unique request class.

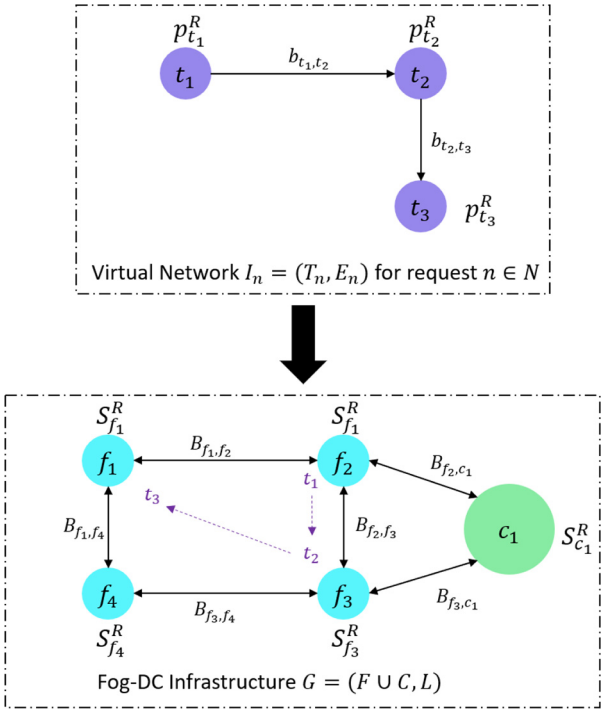

Fig. 2. Mapping of an IoT request virtual network $I_n = (T_n, E_n)$ to fog-DC infrastructure $G = (F, L)$.

then the number of tasks in request n is q_k. Moving forwards, for ρ-percentile of expected IoT traffic, we define $N = N_\rho$ to be the set of IoT requests. Since $\mathcal{P}(x; \lambda)$ is monotonically increasing for fixed λ, larger values of ρ yield a larger set N.

B. IoT Virtual Network

Each of the IoT requests is divisible into a set of interdependent atomic tasks modeled as a weighted directed virtual network graph, represented by a task dependency graph (TDG) $I_n = (T_n, E_n)$ (Fig. 2), for request $n \in N$; T_n denotes the set of tasks in request n, and E_n denotes the set of directed virtual networking links between tasks.

For a request $n \in N$, each task $t \in T_n$ has a set of VM Cloud computing requirements: 1) CPU capacity p_t^{CPU}; 2) memory processing requirement p_t^{MEM}; 3) storage requirement p_t^{STR}; and 4) processing order o_t with respect to any other $t' \in T_n$. For simplicitly, we define the set $R = \{\mathrm{CPU}, \mathrm{MEM}, \mathrm{STR}\}$ to be the set of VM resource requirements. This processing order o_t is used for the construction of the IoT TDG I_n.

C. Fog-DC Physical Infrastructure

The fog-DC physical infrastructure is represented by a directed graph $G = (W, L)$ (Fig. 2), where $W = F \cup C$ is the set of potential fog node locations F and available Cloud Data Centre sites C in the vicinity of the designed infrastructure, and L is the set of wired/wireless bidirectional links connecting fog-DC nodes.

Each fog-DC node $w \in W$ has a maximum available resource limit of S^r_w, $r \in R$. Between any two fog-DC nodes $w, w' \in W$, the link $l_{(w,w')} \in L$ has a maximum bandwidth capacity of $B_{(w,w')} > 0$.

The set of fog-DC nodes $W = F \cup C$ contains both fog node candidates and fixed Cloud nodes. This leverages the low-latency benefits of fog nodes on the network edge, and the large memory and storage capabilities of the Cloud. The result is a designed and dimensioned architecture that can satisfy low-latency responses from IoT with fog, and pass long-term storage and latency insensitive application processing to the Cloud.

Altogether, we have $|N|$ virtual networks—one for each IoT request and one physical network of fog nodes; all descriptions of parameters associated with these networks are summarized in Tables I and II. For each IoT virtual network I_n, $n \in N$, we want to establish a network mapping to the fog-DC infrastructure G. A feasible mapping of all IoT tasks to fog-DC nodes must adhere to the resource and bandwidth capacities of the fog-DC infrastructure.

The mapping of each IoT request I_n (Fig. 2) can be divided into hosting and network mapping. Each IoT hosting node $t \in T_n$ belonging to an IoT request $n \in N$ is mapped to a distinguished fog-DC node $w \in W$ by mapping $M_W : T_n \mapsto W$.

We define the set of all possible paths between any two nodes in W as Π, and the set of all possible paths between two specific nodes $w, w' \in W$ as $\Pi_{(w,w')}$. Each virtual IoT link $e = (t, t')$, $e \in E_n$ belonging to an IoT request $n \in N$ is mapped to a fog-DC path $\pi \in \Pi_{(w,w')} \subseteq \Pi$ via $M_E : E_n \mapsto \Pi_{(w,w')}$, where $M_W(t) = w$ and $M_W(t') = w'$. Note that if $w = w'$, then $\Pi_{(w,w')} = \varnothing$ since no path is needed.

IV. Model Considerations

A. Reachable Servers

The large selling point of a fog infrastructure closer to an IoT device is reduced latency which benefits applications needing real time or near real-time responses. Consider the IoT device from which a task $t \in T_n$, $n \in N$ is executed; for simplicity, we refer to this device as IoT device t. Suppose IoT device t has a range[3] γ_t, and a distance $d_{t,w}$ between an IoT device t and a fog node $f \in F$. If $d_{t,f} \leq \gamma_t$, we consider this fog node *reachable* from t, and if $d_{t,f} > \gamma_t$, then f can be accessed by t by one of many available access points a within the task's radius of communication. Similarly, either a fog node f or an access point a can act as a gateway to reach any cloud data centers. This communication is visually represented in Fig. 3.

B. Network Congestion

In order to correctly model the latency given to IoT-fog mapping, we must consider the effect IoT traffic has on physical link congestion. For a request $n \in N$ such that $M_K(n) = k$, let $\{X_n(\tau), \tau \geq 0\}$ be the volume of transmitted IoT data by time τ following a Poisson process with rate λ_k per unit time, i.e., $\mathbb{E}[X_n] = \lambda_k \ \forall n \in N, M_K(n) = k$. The total rate of data entering fog-DC is the sum of these individual request amounts; therefore, the overall arrival rate is

$$\sum_{n \in N} \lambda_{M_K(n)} = \lambda_\rho. \tag{3}$$

[3]A survey on IoT found the range of most devices were from 10 to 100 m, depending on the communication standard and technology used [41].

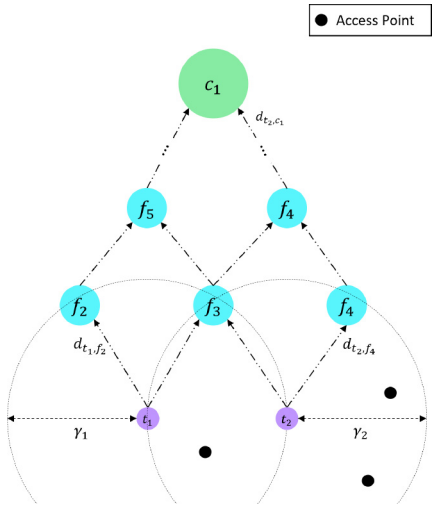

Fig. 3. Each task can directly communicate with any fog node within its range of communication γ. Fogs can also be accessed via an Internet access point at a greater latency cost. Every fog node and access point can act as a gateway to Cloud devices.

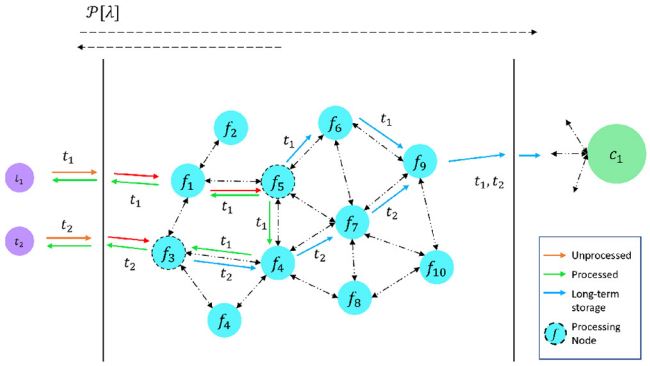

Fig. 4. Fog-DC infrastructure divided into IoT, fog, and Cloud to illustrate how congestion may occur. Shown is the lifecycle of a single request with tasks t_1 and t_2; a response to t_1 is required from fog f_5, and a response to t_2 is required from t_3. Once processed, data are also propagated to the Cloud for long-term storage.

By the construction of N in (2), an increase in ρ to ρ' will yield an increased arrival rate $\lambda'_{M_K(n)}$ $\forall n \in N$, hence an increased overall arrival rate $\lambda_{\rho'}$.

Within the fog, we define the *outermost* nodes to be the nodes that fall within the radius of communication γ_t of at least one IoT device t; these are also called the *edge* nodes. We define the *innermost* nodes to be the fog nodes that link directly to a Cloud Data centre. All nodes in between are referred to as *core* fog nodes.

Regardless of where data from task t is processed, we assume all data reach the Cloud at some point either for processing or long-term storage. In Fig. 4, we have divided our system into three layers—IoT, fog, and Cloud. The amount of data that enters the fog follows a Poisson distribution with rate λ_ρ and expects a response for each data point; we assume the amount of data that eventually enters the Cloud is equivalent. In other words, for every data point transmitted by IoT device t, the data's lifecycle includes being processed by a node either in the Cloud or the fog network with a response being returned, and long-term storage being propagated to the Cloud.

Since we assume all data must pass through the Cloud, we recognize substantial link congestion entering the Cloud; however, having some data processing occur in the fog results in reduced link congestion returning from the Cloud. Similarly, there may be higher link congestion as we move toward the innermost nodes, but the amount of link congestion from processing responses is reduced as we process more tasks in the outermost nodes.

Congestion on a single link can be modelled as an $M/M/1$ queue, where requests are processed and released at an exponentially distributed rate μ_w by $w \in W$. We define p_w to be the probability that a fog-DC node $w \in W$ receives the executed IoT request such that $\sum_{w \in W} p_w = 1$; this probability is based on available resources of w and distance from the IoT devices. We also define the expected number of IoT requests received by w as $p_w \lambda_\rho$. Using the IoT data arrival and service rates, as well as Little's law [42], we determine the average wait time to be processed by $w \in W$, denoted $\bar{\eta}_w$ to be

$$\bar{\eta}_{w;\rho} = \frac{p_w \lambda_\rho}{\mu_w \left(\mu_w - p_w \lambda_\rho\right)}. \tag{4}$$

For a fixed u_w and p_w, $\bar{\eta}_w$ is monotonically increasing with $\lambda_\rho > 0$. By (2) and (3), λ_ρ is monotonically increasing with the percentile ρ; therefore, an increased percentile ρ leads to a larger average wait time.

Let $L_w = \{l \in L \mid l = (w_0, w) \in L\} \cup \{(t, w) \mid M_W(t) = w\}$ be the set of links feeding into w, and let $H_{w;\rho}$ denote the random variable of waiting time before being processed by node $w \in W$ for a network accepting the ρ-percentile of IoT traffic. We know $\mathbb{E}[H_{w;\rho}] = \bar{\eta}_{w;\rho}$ by definition. By [42], the cumulative distribution $G_{w;\rho}(x)$ of $H_{w;\rho}$ is

$$G_{w;\rho}(x) = \mathbb{P}\left[H_{w;\rho} \leq x\right] = 1 - \frac{p_w \lambda_\rho}{\mu_w} e^{-(\mu_w - p_w \lambda_\rho)t}. \tag{5}$$

We denote this wait time $H_{w;\rho}$ to represent our *congestion factor* for a link (w_0, w), $w \in L_w$, and ρ-percentile IoT traffic. For any response from w to t, we define the service rate $\mu_t = 0$; therefore, we define $H_{t;\rho} = 0$, $t \in T_n, n \in N$.

Let π_W be the set of all nodes $w \in W \cup T_n, n \in N$ in a path $\pi \setminus \{w_0\}$ $\pi \in \Pi_{(w_0,w)}$. To estimate congestion over a path from w_0 to w', Algorithm 1 simulates the maximum waiting time over all paths $\pi \in \Pi_{(w_0,w)}$ which gives a worst case congestion per simulation, and calculate the mean and variance. By central limit theorem, [43], we can approximate the congestion factor over any path by a normal distribution for a large number of m simulations.

Note that, a path $\Pi_{(w_0,w)}$ has at most two elements in $T_n, n \in N$ at the beginning and/or end of the sequence. Let $G_{(w_0,w);\rho}(x)$ represent the cumulative distribution function of $H_{(w_0,w);\rho}$.

Once an IoT request is processed by a fog node and a response is returned to the IoT device, the request data are also sent to the Cloud for long-term storage; however, since no further processing of the request is needed, the link congestion associated with this data instance is negligible. From this, we infer that congestion toward the Cloud is alleviated

Algorithm 1: Simulate Congestion Over a Path w_0 to w

Result: Estimated congestion distribution $H_{(w_0,w);\rho}$ for ρ-percentile IoT traffic.
Initialize $\rho \in [0.5, 1)$;
for i = *1 to* m **do**
  for $w \in W \cup T_n, n \in N$ **do**
    $\eta_w \leftarrow H_{w;\rho}$ // Sample
  end
  $\tilde{\eta}_{(w_0,w),i} \leftarrow \max\{\sum_{w \in \pi_W} \eta_{w;\rho} : \pi \in \Pi_{(w_0,w)}\}$
end
$\bar{\eta}_{(w_0,w)} \leftarrow \frac{1}{n}\sum_{i=1}^{n} \tilde{\eta}_{(w_0,w),i}$
$\hat{\sigma}^2_{w_0,w} \leftarrow \frac{1}{n-1}\sum_{i=1}^{n}(\tilde{\eta}_{(w_0,w),i} - \bar{\eta}_{(w_0,w)})^2$
$H_{(w_0,w);\rho} \sim \mathcal{N}(\bar{\eta}_{(w_0,w)}, \hat{\sigma}^2_{w_0,w})$.

as the probability that a request is processed within the fog network instead of the Cloud increases.

We make the simplistic assumption that a fog-DC node w can only process one IoT request at a time, which is not true in practice. Furthermore, our current analysis of network congestion is based on placing a fog node f in every candidate location in F; in reality, only a handful of these locations will be used, raising the congestion within the fog network. To determine a more accurate estimate of how much congestion is present surrounding different nodes in the network, a simulation is necessary provided a network architecture and expected IoT traffic; a simulation framework, such as iFogSim [39], could be used, which we propose to do in future work.

C. Transmission Delay

Each task t has a transmission file size s_t^+ and a bandwidth requirement[4] b_t that remain consistent throughout transmission from t to t' where $o_t = o_{t'} - 1$, $t, t' \in T_n$, $n \in N$. Wang *et al.* [45] found a positive correlation between the number of hops and the latency of Internet communication between two nodes under 1000 km in distance; therefore, we model transmission delay using hops.

For fog node $f \in F$, let $G^{-1}_{f;\rho}(\cdot)$ denote the inverse cumulative distribution function of $H_{f;\rho}$, and $G^{-1}_{(t,w);\rho}(\cdot)$ denote the inverse cumulative distribution function of $H_{(t,w);\rho}$. For $U \sim$ Uniform(0, 1), we can represent a random sample of congestion over a path as $H_{f;\rho} = G^{-1}_{f;\rho}(U)$ and $H_{(t,w);\rho} = G^{-1}_{(t,w);\rho}(U)$. This congestion can also be interpreted as queuing delay or wait time of a package over the specified link. Let $\omega \in (0, 1)$ represents the percentile of estimated congestion for a random path; higher values of ω yield a worst case estimate of transmission delay.

For a fog-DC node $w \in W$, the transmission delay is dependent on the transmission bandwidth, the data file size, an upper bound on the number of hops from t to w denoted $h_{t,w}$, and any network congestion found along the way. Based on these factors, we devise an approximation to the communication delay between a task t and a fog-DC node w denotes $\nu^{\rho,\omega}_{t,w}$.

For a fog node $f \in F$ such that $M_W(t) = f$ and $d_{t,f} \le \gamma_t$, and ρ-percentile IoT traffic, we get the transmission latency as

$$\nu^{\rho,\omega}_{t,f} = \frac{s_t^+}{b_t} + G^{-1}_{f;\rho}(\omega). \tag{6}$$

If a fog-DC node $w \in W$ is not within the communication range γ_t of a task t, the expected transmission delay becomes

$$\nu^{\rho,\omega}_{t,w} = h_{t,w} \cdot \frac{s_t^+}{b_t} + G^{-1}_{(t,w);\rho}(\omega) \tag{7}$$

where we only consider one hop of congestion entering the fog-DC infrastructure.

We assume the bandwidth of a task remains constant throughout its transmission and response, a response file size s_t^- after processing, and a ω-percentile of congestion on any path. Then, for a task mapping $M_W(t) = w$, we have

$$\begin{aligned} \nu^{\rho,\omega}_{t,w} &= h_{t,w} \cdot \frac{s_t^+}{b_t} + G^{-1}_{(t,w);\rho}(\omega) \\ \nu^{\rho,\omega}_{w,t} &= h_{w,t} \cdot \frac{s_t^-}{b_t} + G^{-1}_{(w,t);\rho}(\omega) \end{aligned} \tag{8}$$

for $h_{t,w} = h_{w,t}$.

D. Physical Network Delay

Data transmitted by t to w are a package of size s_t^+ with response package size s_t^-. For a pair of ordered and successive tasks t, t' and task mappings $M_W(t) = w$ and $M_W(t') = w'$, the network delay is dependent on the task bandwidth b_t, the response file size s_t^-, the ρ-percentile of IoT traffic, and the ω-percentile of resulting congestion. We define the network delay from a task t as

$$\phi^{\rho,\omega}_t(w, w') = h_{w,w'} \cdot \left(\frac{s_t^-}{b_t} + G^{-1}_{(w,w');\rho}(\omega)\right) \tag{9}$$

where $h_{w,w'}$ is an upper bound on the number of hops between w and w', and $\omega \in (0, 1)$ is the percentile of congestion expected on a random path $\pi \in \Pi_{(w,w')}$.

For a single request $\{n \in N \mid M_K(n) = k\}$, we order the request tasks $t_1, \ldots, t_{q_k}$, each with differing file sizes and bandwidth requirements to the next task. For $M_W(t_i) = w_i$, the total physical network delay for request $n \in N$ is

$$\phi^{\rho,\omega}_n(w_1, w_{q_k}) = \sum_{i=1}^{q_k - 1} \phi^{\rho,\omega}_{t_i}(w_i, w_{i+1}). \tag{10}$$

When modeling a large network, these approximate calculations of physical network delay between each task may become too large. We invoke a worst case estimate of physical network delay to our model to provide model scalability.

Consider a pair of fog-DC nodes $w, w' \in W$, and let $\pi \in \Pi_{(w,w')}$ be the set of all possible paths between these two nodes. If we assume data can take any path between these nodes, we define π^* to be the critical path (longest path), and h_{π^*} to be the total number of hops taken to traverse this path.

[4]The most common IoT transmission standard is IEEE 802.15.4 [41], which for a common frequency of 2.4 GHz transmits at 250 kb/s, with this rate decreasing with lower transmit frequencies [44], giving us an idea of the range of transmission bandwidth b_t.

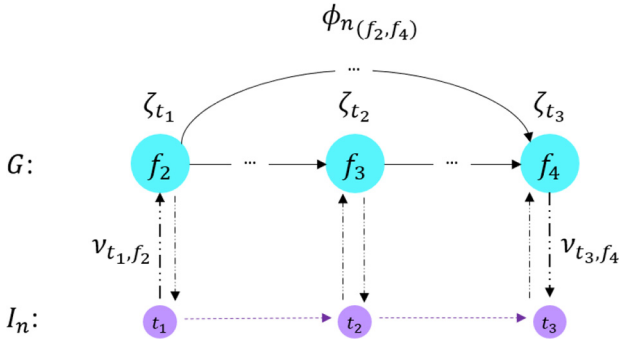

Fig. 5. End-to-end delay for a sequence of tasks is the sum of transmission times, processing times, and routing time.

When we are considering the routing time from the first task to the last task, we assume the worst case scenario regardless of whether there exists shorter routing through the other tasks. We also define

$$r_n = \max_{i=1,\ldots,q_k} \left\{ \frac{s^-_{t_i}}{b_{t_i}} \right\} \tag{11}$$

as the worst case transmission rate. We then define

$$\tilde{\phi}^{\rho,\omega}_n(w_1, w_{q_k}) = h_{\pi^*}\left(r_n + G^{-1}_{(w_1,w_{q_k});\rho}(\omega) \right) \tag{12}$$

as the worst case transmission rate with congestion percentile $\omega \in (0, 1)$.

E. Total Delay

For a task t and mapping $M_W(t) = w$, we incur the transmission delays $\nu^{\rho,\omega}_{t,w}$ to transmit and $\nu^{\rho,\omega}_{w,t}$ to receive a response (8).

Once a task is transmitted to a fog node, it must be processed before moving on. Each task t has a set computing time ζ_t which is specific to the set of resources needed for that task $\{p^r_t \mid r \in R\}$; as a result, it does not depend on the fog node on which it is executed.

For a mapping $M_K(n) = k$, let the tasks of the request n be ordered $t_1, t_2, \ldots, t_{q_k}$ such that $t < t'$ if $o_t < o_{t'}$. For a task mapping $M_W(t_i) = w_i$ for each $t_i \in T_n$, the end-to-end delay with ρ-percentile IoT traffic and ω-percentile congestion can be defined as

$$\nu^{\rho,\omega}_{t_1,w_1} + \tilde{\phi}^{\rho,\omega}_n(w_1, w_{q_k}) + \nu^{\rho,\omega}_{w_{q_k},t_{q_k}} + \sum_{i=1}^{q_k} \zeta_{t_i}. \tag{13}$$

Fig. 5 shows that each task is uploading data to their paired fog-DC node, and may receive a response. In order for a fog-DC node w to process its paired task t, it must both receive the task data from t as well as the processed data from the preceding fog node. We assume all tasks $t \in T_n$ for a request $n \in N$ are transmitted synchronously, so there exists no delay between the start of each task. To this end, we make the assumption that for a task mapping pair $M_W(t_i) = w_i$ and $M_W(t_j) = w_j$, $o_{t_i} < o_{t_j}$ with ρ-percentile IoT traffic and ω-percentile congestion, we have

$$\nu^{\rho,\omega}_{t_i,w_i} + \zeta_{t_i} + \tilde{\phi}^{\rho,\omega}_n(w_i, w_j) \geq \nu^{\rho,\omega}_{t_j,w_j}. \tag{14}$$

This ensures that w_j receives the task information from t_j prior to any data required from w_i, allowing w_j to begin on any processing required from the previous tasks immediately upon arrival. All descriptions of variables associated with model considerations are summarized in Table III.

Though here we assume a normal distribution of congestion and a worst case routing delay, in practice, we would use historical data on incoming IoT data to properly model a probability distribution describing routing delay patterns. In the event of no current IoT devices set up in the fog designed area, we elect to use historical data of similar IoT devices from surrounding districts. This is left for future work.

V. MILP Model

Design and Dimensioning Approach

Design and dimensioning are one of the most important aspects of fog-DC management, since they are directly related to the cost and the QoS requirements of IoT computing services. Efficient design and dimensioning will have a positive impact on fog-DC service provider's profitability. Resource allocation to the incoming IoT requests could be performed in a batchwise fashion. The size of the batch can be modified depending on the considered topology and IoT traffic so as to ensure real-time response to the requests. The fog-DC design and dimensioning problem is in: 1) selecting the optimal location and dimensioning of fog-DC sites and 2) minimizing the cost of resources (e.g., computing and communication) while satisfying QoS requirements of IoT requests. These requirements include: 1) *Bandwidth:* transmission capacity between tasks of the IoT request; 2) *Latency:* the time it takes to process the request and receive the response; 3) *Computing:* computing capacity to process the tasks of the IoT request; 4) storage and memory capacity requirements of each IoT task; and 5) processing order of the tasks that composes the IoT request.

To evaluate the merits of the proposed fog-DC design, dimensioning and resource allocation approach, we propose the following MILP-based mathematical formulation we call fog-DC-MILP. A virtual link $e \in E_n$ is defined by the pair $e = (t, t')$ such that $t, t' \in T_n$, $n \in N$, with $o_t = o_{t'} - 1$ meaning t' immediately follows t in the order of tasks. Each link $e \in E_n$ has a data transfer capacity requirement b_t between a pair of tasks t and t'. Furthermore, we invoke a partial latency requirement τ_t that the completion of all tasks up to and including t must satisfy.

Recall, for a request $I_n = (T_n, E_n)$, we define the task mapping as $M_W : T_n \mapsto W$ and the network mapping as $M_E : E_n \mapsto \Pi$. When an IoT request arrives, the fog-DC provider has to determine whether it is to accept or reject. This decision is largely based on the QoS requirements of the IoT request, the availability of fog-DC resources, and the economic cost of accepting the request. The total cost of a request I_n is then represented as the combined cost of the task mapping, and the virtual link mapping as follows:

$$\text{COST}[I_n] = \text{COST}[M_W(T_n), M_E(E_n)]. \tag{15}$$

TABLE I
FOG INFRASTRUCTURE NETWORK $G = (F, L)$

Parameters	Description
F	Set of possible Fog locations.
C	Set of existing Cloud Data Centres.
W	Set of Fog-DC nodes; $W = F \cup C$.
L	Set of Fog-DC bidirectional links.
R	Type of resource that can be CPU, Memory or Storage. $R = \{\text{CPU}, \text{MEM}, \text{STR}\}$.
S_w^r	Maximum capacity of resource $r \in R$ in node $w \in W$.
B_l	Bandwidth capacity of physical link $l \in L$.
Y_{MAX}^r	Maximum number of Fog-DC locations that can receive resource $r \in R$.
c_l	Bandwidth unit cost of using substrate link $l \in L$.
c_w^r	Unit cost of using resource $r \in R$ in node $w \in W$.
u_f^r	Unit cost of setting up resource $r \in R$ in node $f \in F$.
D_f^r	Capital cost of setting up resource $r \in R$ in node $f \in F$.
Π	Set of paths in Fog-DC network.
$\Pi_{(w,w')}$	Set of paths between Fog-DC nodes w and w'.

TABLE II
IOT VIRTUAL NETWORK $I_n = (T_n, E_n)$

Parameters	Description
N	Set of IoT requests.
I_n	Virtual network representing IoT request $n \in N$.
T_n	Set of tasks in an IoT request I_n.
E_n	Set of virtual directional links between tasks in an IoT request I_n.
R	Type of resource that can be CPU, Memory or Storage. $R = \{\text{CPU}, \text{MEM}, \text{STR}\}$.
p_t^r	Required capacity of resource of type $r \in R$ for task t.
o_t	Order number of task t in request T_n.
b_t	Transfer bandwidth between a task t up to the next ordered task.
s_t^+	File size of package uploaded by task t.
s_t^-	File size of package after processed by $w \in W$, either as a response to t or outgoing to the next task t'.
τ_t	The end-to-end delay threshold for the completion of the first task up to and including t.

A. Decision Variables

1) Design and Dimensioning Variables: We recall that r is the type of fog-computing resource that takes values in the set $R = \{\text{CPU}, \text{MEM}, \text{STR}\}$. We define the float decision variable z_f^r to measure the amount of computing resources of type r required to be set up in fog-DC node $f \in F$. An upper bound S_f^r will be set up to limit the available fog-computing resources of type r that can be set up in fog-DC node f.

We define a binary decision variable that denotes whether fog node $f \in F$ requires resources $r \in R$ to be set up; this is precisely whether z_f^r is greater than zero

$$y_f^r = \begin{cases} 1, & \text{if } z_f^r > 0 \\ 0, & \text{otherwise.} \end{cases} \tag{16}$$

The variables z_c^r and y_c^r, $c \in C$, respectively, represent the amount of resource r available in c, and whether this amount is strictly positive; these values are fixed as we cannot alter the resources provided in the Cloud.

2) IoT Request Mapping Variables: To decide on the acceptance of an IoT request I_n, we need to define the following decision variables:

$$a_n = \begin{cases} 1, & \text{if an IoT request } n \in N \text{ is accepted to be processed by fog-DC} \\ 0, & \text{otherwise} \end{cases} \tag{17}$$

$$x_w^t = \begin{cases} 1, & \text{if virtual node } t \in T_n,\ n \in N \text{ is assigned to fog-DC node } w;\ M_W(t) = w \\ 0, & \text{otherwise} \end{cases} \tag{18}$$

$$x_\pi^e = \begin{cases} 1, & \text{if virtual link } e \in E_n,\ n \in N \text{ is assigned to physical path } \pi \in \Pi;\ M_E(e) = \pi \\ 0, & \text{otherwise.} \end{cases} \tag{19}$$

The description and domain of fog-DC-MILP decision variables are summarized in Table IV.

B. Objective Function

The objective of fog-DC-MILP is to minimize the total cost of fog design and dimensioning as defined by the IoT task and link mappings [15] as follows:

$$\begin{aligned} Z_{\text{MILP}} = \min & \sum_{f \in F} \sum_{r \in R} \left(D_f^r \cdot y_f^r + u_f^r \cdot z_f^r \right) \\ & + \sum_{n \in N} \sum_{t \in T_n} \sum_{w \in W} \sum_{r \in R} c_w^r \, p_t^r \, x_w^t \\ & + \sum_{n \in N} \sum_{e \in E_n} \sum_{(w,w') \in W^2} \sum_{\pi \in \Pi_{(w,w')}} \sum_{l \in \pi} x_\pi^e \, c_l \, b_t \end{aligned} \tag{20}$$

TABLE III
MODEL CONSIDERATIONS

Variable	Description
K	Set of unique request classes.
$\mathcal{P}(x;\lambda)$	Poisson cumulative function with IoT traffic arrival rate λ.
ρ	Percentile of estimated IoT traffic volume entering network.
ω	Percentile of estimated congestion over any network path.
λ_k	Arrival rate of all requests with class $k \in K$.
λ_ρ	Sum of arrival rates over all classes using ρ-percentile IoT traffic.
p_w	Probability of a task t accessing Fog-DC node $w \in W$.
μ_w	Service rate of Fog-DC node $w \in W$.
$\bar{\eta}_{w;\rho}$	Mean waiting time/congestion of any link entering $w \in W$ with ρ-percentile IoT traffic.
$H_{w;\rho}$	Random variable representing congestion entering node $w \in W$.
$G_{w;\rho}(t)$	Cumulative Distribution Function of $H_{w;\rho}$.
$H_{(w,w');\rho}$	Random variable representing congestion over any path $\pi \in \Pi_{(w,w')}$ with ρ-percentile IoT traffic.
$G_{(w,w');\rho}(t)$	Cumulative Distribution Function of $H_{(w,w');\rho}$.
$\nu^{\rho,\omega}_{t,w}$	Upload transmission delay with ρ-percentile IoT traffic and ω-percentile congestion.
$\nu^{\rho,\omega}_{w,t}$	Download transmission delay with ρ-percentile IoT traffic and ω-percentile congestion.
$\phi^{\rho,\omega}_n(w_1, w_{q_k})$	Routing delay through intermediate nodes $w_1, ..., w_{q_k} \in W$ with ρ-percentile IoT traffic and ω-percentile congestion.
$\tilde{\phi}^{\rho,\omega}_n(w_1, w_{q_k})$	Worst-case routing delay with ρ-percentile IoT traffic and ω-percentile congestion.
ζ_t	Fixed processing time for task t.

where D^r_f and u^r_f are the capital cost (Capex) and the unit cost, respectively, to set up computing resource of type $r \in R$ in fog-DC node $f \in F$, and W^2 is the set of pairs $W \times W$.

We have divided the objective function (20) into three terms to express the optimization of different factors in fog-DC-MILP: 1) the fixed and unit cost of resources placed in each fog node $f \in F$ which represents the design and dimensioning cost; 2) the cost of allocating resource $r \in R$ for task $t \in T_n$ to fog-DC nodes $w \in W$; and 3) the networking cost of assigning path $\pi \in \Pi_{(w,w')}$ to a virtual link $e = (t, t') \in E_n$, $n \in N$.

1) IoT Resource Allocation Constraints: For the following constraints, we define Q_t as the set of possible fog-DC sites $w \in W$ that can serve a task of an IoT request $t \in T_n$, and Q_e as the set of possible paths $\pi \in \Pi$ that can be assigned to $e \in E_n$. For a link $l \in L$, we define Π^l to be the set of paths that contain the link l, and $\Pi^l_{(w,w')}$ to be the set of paths from w to w' that contain link l

$$\sum_{n\in N} a_n \geq |N| \cdot A \tag{21}$$

$$\sum_{w\in W} x^t_w = a_n, \;\; t \in T_n; \;\; n \in N \tag{22}$$

$$x^t_w = 0; \;\; \forall w \in W \setminus Q_t \quad \forall t \in T_n, \;\; n \in N \tag{23}$$

$$x^e_\pi = 0; \;\; \forall \pi \in \Pi \setminus Q_e \quad \forall e \in E_n, \;\; n \in N \tag{24}$$

$$x^t_w \cdot x^{t'}_{w'} = \sum_{\pi\in\Pi_{(w,w')}} x^e_\pi; \;\; (w, w') \in W^2 \quad e = (t, t') \in E_n, \;\; n \in N \tag{25}$$

$$\Delta^{t_i,\rho,\omega}_{(w_1,w_i)} \leq \tau_{t_i}; \;\; i \in \{1, \ldots, |T_n|\}, \;\; t_i \in T_n \quad n \in N, \;\; \omega \in (0, 1) \tag{26}$$

$$\sum_{n\in N} \sum_{\substack{e\in E_n \\ e=(t,t')}} \sum_{(w,w')\in W^2} \sum_{\pi\in\Pi^l_{(w,w')}} b_t \cdot x^e_\pi \leq B_l, \;\; l \in L \tag{27}$$

$$\sum_{n\in N} \sum_{t\in T_n} x^t_w \cdot p^r_t \leq z^r_w, \;\; r \in R; \; w \in W. \tag{28}$$

Equation (21) expresses the acceptance ratio of the proposed fog-DC design where parameter $A \in [0, 1]$. Equations (22) and (23) express the selection of fog-DC nodes to process IoT request tasks. Equations (24) and (25) ensure that only one valid embedding path is assigned for each virtual link. In (26), we define the partial delay Δ between the first task t_1 and the ith task t_i with transmission delay and congestion percentile $\omega \in (0, 1)$ to be

$$\Delta^{t_i,\rho,\omega}_{(w_1,w_i)} = \nu^{\rho,\omega}_{t_1,w_1} \cdot x^{t_1}_{w_1} + \tilde{\phi}^{\rho,\omega}_{(w_1,w_i)} \cdot x^{t_1}_{w_1} \cdot x^{t_i}_{w_i} + \nu^{\rho,\omega}_{w_i,t_i} \cdot x^{t_i}_{w_i} + \sum_{m=1}^{i} \zeta_{t_m}, \;\; t \in T_n; \;\; n \in N; \; \omega \in (0, 1) \tag{29}$$

which expresses the QoS requirements for the mapping of virtual links to meet the latency threshold of tasks. Recall, ρ is the percentile of IoT traffic that defines the set N and the congestion distribution, therefore it is set prior to modelization. We note that the term $\phi^{\rho,\omega}_{(w_1,w_i)} \cdot x^{t_1}_{w_1} \cdot x^{t_i}_{w_i}$ in (29) is quadratic, and we use the same linearization technique as in Section VI-C4. Equations (27) and (28) express, respectively, the bandwidth and resource (CPU, memory, and storage) capacity of links and nodes.

2) Fog-DC Design and Dimensioning Constraints:

$$z^r_f \leq S^r_f \cdot y^r_f, \;\; f \in F; \;\; r \in R \tag{30}$$

$$\sum_{f\in F} y^r_f \leq Y^r_{\text{MAX}}, \;\; r \in R. \tag{31}$$

Equation (30) expresses that no resources r can be set up in a given fog node f if it is not selected as an optimal location to

TABLE IV
FOG-DC-MILP DECISION VARIABLES

Decision Variables	Domain	Description
z_f^r	$\mathbb{R}, \geq 0$	Dimension of resources $r \in R$ allocated to Fog node $f \in F$.
z_c^r	$\mathbb{R}, \geq 0$	Dimension of resources $r \in R$ allocated to Cloud node $c \in C$; FIXED.
y_f^r	$\{0,1\}$	Whether a resource $r \in R$ is present in Fog node $f \in F$.
y_c^r	$\{0,1\}$	Whether a resource $r \in R$ is present in Cloud node $c \in C$; FIXED.
a_n	$\{0,1\}$	Whether a request I_n is accepted by Fog-DC.
x_w^t	$\{0,1\}$	Whether task $t \in T_n$ is mapped to Fog-DC node $w \in W$.
x_π^e	$\{0,1\}$	Whether virtual link $e = (t, t') \in E_n$ is mapped to physical path $\pi \in \Pi$.

set up a fog node, and must not exceed the maximum resource amount for that fog location denoted by S_f^r. Equation (31) expresses the maximum number of fog-DC locations Y_{MAX}^r that can receive a resource r.

C. Extensibility of fog-DC Formulation

Suppose a fog infrastructure already exists and we wish to extend it to support higher IoT traffic. We define the current set of dimensioned fog nodes as F'. Into the current formulation, we integrate $y_f^r = 1$ fixed and $z_f^r \in [\bar{z}_f^r, S_f^r]$ for $f \in F'$, where $\bar{z}_f^r$ is the current dimensioning and S_f^r is the maximum resource capacity of z_f^r. Proceeding normally beyond integration will yield the appropriate *extended* formulation.

D. Drawbacks of fog-DC-MILP Formulation

This formulation allows for both node and link mappings to be performed in one shot; however, since the fog-DC-MILP formulation is based on an integer linear programming model, it suffers from scalability issues. With a large number of IoT requests, the mathematical model takes on a large number of variables and constraints. This is potentially a significant drawback to solve the MILP model optimally in a reasonable computation time.

1) For a given IoT request $I_n = (T_n, E_n)$, a virtual link $e \in E_n$ can be assigned to up to
$$|W| \times |W| \times |\Pi| \tag{32}$$
possible embedding path solutions.
2) Thus, for $|E| = \max_{n \in N} |E_n|$ being the maximal size of virtual links, $|E|$ virtual links over N IoT requests can be assigned to up to
$$(|E| \times |W| \times |W| \times |\Pi|)^{|N|} \tag{33}$$
possible mapping solutions, which can be approximated by the exponential number $O(g^{|N|})$.

Node and link embedding is known to be an NP-hard problem, equivalent to a multiway separator problem [46]. To address this complexity, we propose a decomposition approach based on the column generation technique [47]. This implies a pricing of nonbasic variables to generate new columns or to prove LP optimality at a node of the branch-and-bound tree.

VI. COLUMN GENERATION FORMULATION FOR AN IOT SERVICE RESOURCE ALLOCATION (FOG-DC-CG)

To avoid the scalability issue identified in the MILP formulation, we propose to use the column generation formulation (Fog-DC-CG) to allocate fog-DC resources to service IoT requests. We reformulate the resource allocation problem in terms of independent fog-DC configurations (IFCs). Each IFC solves the resource allocation problem of a single IoT request. We denote by Θ the set of all possible IFCs. Accordingly, the resource allocation problem can then be formulated with respect to the decision variables λ_θ such that

$$\lambda_\theta = \begin{cases} 1, & \text{if IFC } \theta \in \Theta \text{ is used in the Fog mapping solution} \\ 0, & \text{otherwise.} \end{cases} \tag{34}$$

In this new formulation, the mapping problem is to choose a maximum of $|N|$ IFCs, as each IFC is serving one IoT request. The resulting configuration corresponds to what is known as the master problem in a column generation approach, while each configuration IFC corresponds to what is known as the pricing problem. Here, we are making the assumption that parameter $A = 1$, i.e., the fog-DC design should accept all IoT requests $n \in N$.

An IFC configuration $\theta \in \Theta$ is defined by the vector $(a_n^\theta)_{n \in N}$ such that

$$a_n^\theta = \begin{cases} 1, & \text{if IFC } \theta \text{ services the IoT request } I_n \\ 0, & \text{otherwise} \end{cases} \tag{35}$$

$$\sum_{n \in N} a_\theta^n = 1, \quad \theta \in \Theta. \tag{36}$$

We denote by COST_θ the cost of configuration θ, which corresponds to the sum of costs of the resources used (hosting and networking) for the IoT request granted by IFC θ.

The use of column generation formulation divides the original problem into a master problem and a pricing problem with two separate objectives.

1) *Master Problem:* The problem of finding the best subset among the already generated IFCs that minimize the dimensioning costs.
2) *Pricing Problem:* The problem of generating an additional column (IFC) to the constraint matrix of the master problem.

A. Master Problem

The master problem, denoted by fog-CG-M, is defined as follows.

1) Objective Function:

$$\min \sum_{\theta\in\Theta} \text{COST}_\theta\, \lambda_\theta + \sum_{f\in F}\sum_{r\in R} D_f^r \cdot y_f^r + u_f^r \cdot z_f^r \tag{37}$$

where

$$\text{COST}_\theta = \sum_{l\in L} B_\theta^{\text{B}}(l)\cdot c_l + \sum_{w\in W} P_\theta^r(w)\cdot c_w^r. \tag{38}$$

Here, c_w^r and c_l are the same unit resource costs as in Section V-B. $B_\theta^{\text{B}}(l)$ is the bandwidth used on a networking link l by IFC θ and B_l is the maximum available bandwidth on networking link l. We also denote $P_\theta^r(w)$ to be the amount of resource r in fog-DC location w used by IFC θ.

2) IoT Resource Allocation Constraints:

$$\sum_{\theta\in\Theta} \lambda_\theta \cdot P_\theta^r(w) \le z_w^r, \quad w\in W;\ r\in R \quad \left(\alpha_w^r\right) \tag{39}$$

$$\sum_{\theta\in\Theta} \lambda_\theta \cdot B_\theta^{\text{B}}(l) \le B_l, \quad l\in L \quad (\beta_l) \tag{40}$$

$$\sum_{\theta\in\Theta} \lambda_\theta \Delta_\theta^{\text{L},\rho,\omega}(t) \le \tau_t, \quad t\in T_n;\ n\in N \quad (\gamma_t) \tag{41}$$

$$\sum_{\theta\in\Theta} \lambda_\theta \le |N|, \quad (\mu_0) \tag{42}$$

$$\sum_{\theta\in\Theta} \lambda_\theta \cdot a_\theta^n \ge 1, \quad n\in N \quad (\psi_n) \tag{43}$$

Equation (39) expresses the available capacity of resource r in fog-DC node w. Equation (40) expresses the bandwidth capacity of networking link l. Equation (41) expresses the latency threshold that must be satisfied for each task t for some transmission delay, ρ-percentile of IoT traffic, and ω-percentile of congestion. Equation (42) guarantees the convexity of the ILP model. Equation (43) grants the satisfaction of the maximum number of IoT requests.

3) Fog-DC Design and Dimensioning Constraints:

$$z_f^r \le S_f^r \cdot y_f^r, \quad f\in F;\ r\in R \tag{44}$$

$$\sum_{f\in F} y_f^r \le Y_{\text{MAX}}^r, \quad r\in R. \tag{45}$$

Equations (44) and (45) are the same constraints as in the Fog-DC-MILP formulation.

4) Linear Relaxation of fog-CG-M: In order to obtain the dual variables associated with (39)–(43), we formulate a linear relaxation of fog-CG-M. This linear program formulation, denoted Fog-CG-M-LP only differs from Fog-CG-M in the removal of variables z and y. For $c\in C$, let S_c^r be the total amount of resource r in c

$$\begin{aligned} \min\ \text{Fog-CG-M-LP} &= \sum_{\theta\in\Theta} \text{COST}_\theta \lambda_\theta \\ \text{Subject to } &\sum_{\theta\in\Theta} \lambda_\theta \cdot P_\theta^r(w) \le S_w^r; \quad w\in W; \quad r\in R \\ &(40)-(43) \\ &\lambda_\theta \in [0,1]. \end{aligned} \tag{46}$$

B. Pricing Problem

As mentioned previously, the pricing problem corresponds to the generation of an additional configuration (IFC), i.e., an additional column for the constraint matrix of the current master problem. Let α_w^r, β_l, γ_t, μ_0, and ψ_n be the dual variables associated with constraints (39)–(43), respectively, and obtained from solving the Fog-CG-M-Dual. Then, the reduced cost of variable λ_θ for an IFC θ can be written

$$\begin{aligned} \overline{\text{COST}}_\theta = \text{COST}_\theta &- \sum_{n\in N} \psi_n \cdot a_\theta^n + \sum_{r\in R}\sum_{w\in W} \alpha_w^r \cdot P_\theta^r(w) \\ &+ \sum_{l\in L} \beta_l \cdot B_\theta^{\text{B}}(l) + \sum_{t\in T_n}\sum_{n\in N} \gamma_t \cdot \Delta_\theta^{\text{L},\rho,\omega}(t) + \mu_0 \end{aligned} \tag{47}$$

where COST_θ is defined by (38).

We now express (47) in terms of the decision variables of the pricing problem; in order to alleviate notation, we omit θ from the index of the decision variables as follows. Those variables are implicitly defined within the context of θ as follows:

$$a_n = \begin{cases} 1, & \text{if an IoT request } I_n\ n\in N \text{ is serviced} \\ 0, & \text{otherwise} \end{cases} \tag{48}$$

$$x_w^t = \begin{cases} 1, & \text{if virtual node } t\in T_n,\ n\in N \text{ is assigned} \\ & \text{to fog-DC node } w;\ M_W(t) = w \\ 0, & \text{otherwise} \end{cases} \tag{49}$$

$$x_\pi^e = \begin{cases} 1, & \text{if virtual link } e\in E_n,\ n\in N \text{ is assigned} \\ & \text{to physical path } \pi;\ M_E(e) = \pi \\ 0, & \text{otherwise.} \end{cases} \tag{50}$$

Next, we derive the relations between the pricing variables and the coefficients of the master problem for each configuration $\theta\in\Theta$. For each $n\in N$, $a_\theta^n = a_n$. For each fog-DC node $w\in W$ and resource $r\in R$, we have

$$P_\theta^r(w) = \sum_{n\in N}\sum_{t\in T_n} p_t^r \cdot x_w^t. \tag{51}$$

For each link $l\in L$, we have

$$B_\theta^{\text{B}}(l) = \sum_{e=(t,t')\in E_n}\sum_{n\in N}\sum_{(w,w')\in W^2}\sum_{\pi\in\Pi_{(w,w')}^l} b_t \cdot x_e^\pi. \tag{52}$$

For each task $t\in T_n,\ n\in N$, we have

$$\begin{aligned} \Delta_\theta^{\text{L},\rho,\omega}(t) = &\sum_{t\in T_n}\sum_{n\in N}\sum_{(w_0,w)\in W^2} \left(\nu_{t_1,w_1}^{\rho,\omega}\cdot x_{w_1}^{t_1} + \nu_{t,w}^{\rho,\omega}\cdot x_w^t\right) \\ &+ \sum_{t\in T_n}\sum_{n\in N}\sum_{(w_0,w)\in W^2} x_w^t x_{w'}^{t'} \cdot \tilde{\phi}_{(w_1,w)}^{\rho,\omega} \\ &+ \sum_{t\in T_n}\sum_{n\in N}\sum_{i=t_1}^{t} \zeta_i \end{aligned} \tag{53}$$

where $\nu^{\rho,\omega}$, $\tilde{\phi}^{\rho,\omega}$, and ζ are, respectively, the transmission, routing, and processing costs from (13), and t_1 represents the first task of a request.

The substitution of the pricing decision variables into the reduced cost in (47) gives us the pricing problem denoted as Fog-CG-P

$$\begin{aligned} \min &\sum_{e=(t,t')\in E_n}\sum_{n\in N}\sum_{(w,w')\in W^2}\sum_{\pi\in\Pi_{(w,w')}^l} b_t\cdot x_e^\pi\cdot(c_l+\beta_l) \\ &+ \sum_{n\in N}\sum_{t\in T_n} p_t^r\cdot x_w^t\cdot\left(c_w^r+\alpha_w^r\right) \end{aligned}$$

$$+\sum_{t\in T_n}\sum_{n\in N}\sum_{(w_1,w)\in W^2}\left(\nu^{\rho,\omega}_{t_1,w_1}\cdot x^{t_1}_{w_1}+\nu^{\rho,\omega}_{t,w}\cdot x^{t}_{w}\right)$$
$$+\sum_{t\in T_n}\sum_{n\in N}\sum_{(w_0,w)\in W^2} x^{t}_{w}x^{t'}_{w'}\cdot\phi^{\rho,\omega}_{(w_0,w)}$$
$$+\sum_{t\in T_n}\sum_{n\in N}\sum_{i=t_0}^{t}\zeta_i+\mu_0-\sum_{n\in N}\psi_n\cdot a_n. \tag{54}$$

The optimal solution of Fog-CG-P defines the additional configuration to be added via (48)–(50).

C. Pricing Constraints

1) Mapping of IoT Service Tasks:

1) Mapping is done for all tasks of an accepted IoT request I_n

$$a_n \le \sum_{(w,w')\in W^2} x^{t}_{w}x^{t'}_{w'},\ e=(t,t')\in E_n;\ n\in N. \tag{55}$$

2) A task t of an accepted request I_n is assigned to only one fog-DC location node w

$$\sum_{w\in W} x^{t}_{w}\le a_n,\quad t\in T_n;\quad n\in N. \tag{56}$$

2) Mapping of the IoT Request Link:

$$\sum_{(w,w')\in W^2}\ \sum_{\pi\in\Pi^{e}_{(w,w')}} x^{e}_{\pi}\le a_n,\quad e\in E_n;\ n\in N. \tag{57}$$
$$x^{t}_{w}x^{t'}_{w'}\le\sum_{\pi\in\Pi_{(w,w')}} x^{e}_{\pi},\quad (w,w')\in W^2$$
$$e=(t,t')\in E_n,\ n\in N. \tag{58}$$

Equation (57) expresses that if request I_n is accepted then at least one networking path π is assigned to grant data transfer over virtual link e, and likewise (58) for a path assignment to fog-DC site locations w and w'.

3) Latency Relaxation: To push the column generation formulation toward generating viable columns, we add a relaxed latency constraint. We define a variable $\tilde{\Delta}^{\mathrm{L},\rho,\omega}_{w}(t)$ such that

$$\tilde{\Delta}^{\mathrm{L},\rho,\omega}_{w}(t)=\nu^{\rho,\omega}_{t,w}x^{t}_{w}+\sum_{i=t_1}^{t}\zeta_i. \tag{59}$$

From (53), we infer that $\tilde{\Delta}^{\mathrm{L},\rho,\omega}_{w}(t)\le\Delta^{\mathrm{L},\rho,\omega}_{w}(t)$; therefore, we add the constraint

$$\sum_{w\in W}\tilde{\Delta}^{\mathrm{L},\rho,\omega}_{w}(t)\le\tau_t,\quad t\in T_n;\ n\in N. \tag{60}$$

4) Linearization of Quadratic Terms: We note that objective term (53) and constraints (29), (55), and (58) include the quadratic terms $x^{t}_{w}x^{t'}_{w'}$. Since this quadratic term is the product of two binary variables, it can be linearized easily by replacing the quadratic term by a new binary variable $y^{t,t'}_{w,w'}$, where $y^{t,t'}_{w,w'}=x^{t}_{w}x^{t'}_{w'}$ and by adding the constraints

$$y^{t,t'}_{w,w'}\ge x^{t}_{w}$$
$$y^{t,t'}_{w,w'}\ge x^{t'}_{w'} \tag{61}$$

which ensure that $y^{t,t'}_{w,w'}$ will be zero if either x^{t}_{w} or $x^{t'}_{w'}$ are zero. Adding the inequality

$$y^{t,t'}_{w,w'}\ge x^{t'}_{w'}+x^{t}_{w}-1 \tag{62}$$

makes sure that $y^{t,t'}_{w,w'}$ will take value 1 if both binary variables x^{t}_{w} or $x^{t'}_{w'}$ are set to 1. We note that such a linearization technique is done implicitly in our simulation by the used linear solver CPLEX.

D. Solving the Fog-DC-CG Model

The steps involved in solving the Fog-DC-CG model formulated in Section V are as follows.

1) Initialize Fog-CG-M-LP by a subset of dummy configuration that is, a set of artificial IFCs with a large cost.
2) Solve the dual of Fog-CG-M-LP formulation to optimality using the CPLEX solver to obtain dual variables α^{r}_{w}, β_l, γ_t, ψ_n, and μ_0, the variables associated with the Fog-CG-M-LP constraints.
3) Solve the pricing problem Fog-CG-P to optimality using the CPLEX solver. This may generate several possible columns (IFCs).
4) For each column generated, calculate the reduced cost. If a column with a negative reduced cost has been found, add this column to the current master problem and repeat steps 2 and 3. Otherwise, Fog-CG-M-LP is optimally solved.

The optimal solution of fog-CG-M-LP only provides a lower bound on the optimal integer solution of fog-CG-M. We solve the fog-CG-M integer programming formulation to optimality using the branch-and-bound CPLEX solver.

VII. Benchmarks

To make an appropriate comparison between our two proposed models, we define two benchmarks inspired by literature: 1) a matching-based model [26] and 2) a greedy model [48] known as fog-DC-Match and fog-DC-Greedy, respectively.

A. Fog-DC-Match

For a request $t\in T_n, n\in N$, let $r^{t}_{W}:\mathbb{Z}^{+}\mapsto W$ be a function such that $r^{t}_{W}(i)$ returns the fog-DC node with the ith lowest latency to t. We formulate a relaxed MILP to find the fog-DC nodes that minimize the ranking as the first phase of our heuristic

$$\tilde{Z}_{P1}=\min\ \sum_{n\in N}\sum_{t\in T_n}\sum_{i=1}^{|W|} i\cdot x^{t}_{w_i},\qquad w_i=r^{t}_{W}(i) \tag{63}$$
$$\text{Subject to}\ \sum_{n\in N}\sum_{t\in T_n} x^{t}_{w_i}\cdot p^{r}_{t}\le S^{r}_{w_i}$$
$$r\in R,\ w_i=r^{t}_{W}(i)\in W \tag{64}$$
$$x^{t}_{f}\cdot p^{r}_{t}\le S^{r}_{f}\cdot y^{r}_{f}$$
$$f=r^{t}_{W}(i)\in F,\quad i\in\{1,\ldots,|W|\};\ r\in R$$
$$(21)-(23). \tag{65}$$

For a virtual link $e \in E_n, n \in N$, let $r_E^e : \mathbb{Z}^+ \mapsto \Pi$ be a function $r_E^e(i)$ returns the fog-DC path with the ith lowest cost, given the set optimal fog-DC node mappings from the first phase denoted $M_{W,P1} : T_n \mapsto W$, we rank the paths to formulate our second phase of our heuristic

$$\tilde{Z}_{P2} = \min \sum_{n \in N} \sum_{e \in E_n} \sum_{i=1}^{|\Pi|} i \cdot x_\pi^e, \quad \pi = r_E^e(i) \tag{66}$$

$$\text{Subject to (24), (27)}$$

$$x_w^t \cdot x_{w'}^{t'} = \sum_{\pi \in \Pi_{(w,w')}} x_\pi^e$$

$$e = (t, t') \in E_n, \quad n \in N$$

$$M_{W,P1}(t) = w, \quad M_{W,P1}(t') = w'. \tag{67}$$

For design and dimensioning solutions from $\tilde{Z}_{P1}$ and $\tilde{Z}_{P2}$, we know $z_f^r = \sum_{t \in T_n} \sum_{n \in N} x_f^t \cdot p_t^r$, $f \in F$ so we can obtain the objective function Z_{MILP} (20) for comparison with other models.

B. Fog-DC-Greedy

For a set of ordered tasks $\{t_1, \ldots, t_{q_k}\}$ from a request $n \in N$, we wish to allocate task resources to fog-DC nodes by greedily choosing the node w_j with available resources and with minimal latency to t_j . Once a servicing fog-DC node is chosen, we find a path with the minimal routing cost between w_{j-1} and w_j that satisfies the bandwidth constraints per link; this procedure is formulated as Algorithm 2. Aside from reducing design and dimensioning cost, our main objective is to satisfy the latency requirements of tasks for a high fog acceptance rate. For this reason, our greedy metric is first and foremost on latency.

Algorithm 2: Fog-DC-Greedy; Greedy Algorithm by fog-DC Node Latency

Result: Greedy fog-DC Design and Dimensioning.
Enumerate requests such that $N = \{1, 2, ..., |N|\}$.
For request n, enumerate tasks $t_1, ..., t_{q_k}$ where $o_{t_i} < o_{t_j}$, $1 \le i < j \le q_k$, $M_K(n) = k$.
Initialize $\omega \in (0, 1)$, $S^r_{w,\text{USED}} = 0, B_{l,\text{USED}} = 0$.
for *n = 1 to* $|N|$ **do**
  $k \leftarrow M_K(n)$
  for $j = 1$ *to* q_k **do**
    $W' \leftarrow W$;
    // For task t_j, select a node w_j with minimal latency.
    while *True* **do**
      if $j = 1$ **then**
        $\Delta_{t_1} \leftarrow \min\{\nu^{\rho,\omega}_{t_1,w} \mid w \in W'\}$;
        $w_1 \leftarrow \arg\min\{\nu^{\rho,\omega}_{t_1,w} \mid w \in W'\}$;
      end
      else
        $\Delta_{t_j} \leftarrow \min\{\Delta^{t_j,\rho,\omega}_{(w_1,w)} \mid w \in W'\}$;
        $w_j \leftarrow \arg\min\{\Delta^{t_j,\rho,\omega}_{(w_1,w)} \mid w \in W'\}$;
      end
      if $p^r_{t_j} \le S^r_{w_j} - S^r_{w_j,\text{USED}}$ $\forall r \in R$ **then**
        $S^r_{w_j,\text{USED}} \leftarrow p^r_t + S^r_{w_j,\text{USED}}$;
        // Greedily route backwards to w_{j-1}.
        while $j > 1$ **do**
          $c_\pi \leftarrow \min_{\pi \in \Pi_{(w_{j-1},w_j)}} \sum_{l \in \pi} c_l b_l$;
          $\pi \leftarrow \arg\min_{\pi \in \Pi_{(w_{j-1},w_j)}} \sum_{l \in \pi} c_l b_l$;
          if $b_l \le (B_l - B_{l,\text{USED})})$ $\forall l \in \pi$ **then**
            $B_{l,\text{USED}} \leftarrow b_l + B_{l,\text{USED}}$ $\forall l \in \pi$;
            break;
          end
          else
            $\Pi_{(w_{j-1},w_j)} \leftarrow \Pi_{(w_{j-1},w_j)} \setminus \pi$;
          end
        end
      end
      else
        $W' \leftarrow W' \setminus w_j$;
      end
    end
  end
end

VIII. Results

A. Simulation Setup

In this section, we conduct a time and cost comparison between the Fog-DC-MILP model and the heuristic Fog-DC-CG model. Both models are solved using the IBM CPLEX Solver on a machine with an i7 Dual-Core 2.5-GHz CPU and 12 GB of RAM. In addition, we compare the results with the two heuristics Fog-DC-Match and fog-DC-Greedy using the same configurations.

Given the intractability of Fog-DC-MILP, we selected the following configuration settings to produce results in a reasonable amount of time. We chose to use five IoT requests with the varying number of tasks from 1 to 4, totaling ten tasks in all. For simplicity, we set the acceptance ratio (21) threshold to be 1. For scalability, we added an arrival rate $\lambda \in [1, 50]$ to inflate the number of requests, and consequently the number of tasks. As noted in (3), an increase in the IoT traffic volume percentile ρ leads to an increase in the arrival rate λ by similar factors; therefore, we simplify the simulation by only including a varying arrival rate λ. The service rate μ_w for a fog-DC node $w \in W$ was selected in the range $\mu_w \in [10, 50]$ ms. We set the congestion percentile ω to 0.5.

Over the set of tasks, resource requirements were mostly selected from a uniform distribution in $[a_T^r, b_T^r]$ with low and high values of $a_T^{\text{CPU}} = 0$ GHz and $b_T^{\text{CPU}} = 2.5$ GHz for CPU, $a_T^{\text{MEM}} = 0$ GB and $b_T^{\text{MEM}} = 2.5$ GB for MEM, and $a_T^{\text{STR}} = 0$ GB and $b_T^{\text{STR}} = 15$ GB for STR; for each resource, a high resource requirement was assigned to a task above 10 GHz, 10 GB, and 100 GB for CPU, MEM, and STR, respectively. We set the latency requirement for each task at 500–1100 ms to simulate IoT time sensitivity. We set $s_t^+ \in (0, 10]$ and $s_t^- \in (0, 10]$ in megabytes, bandwidth requirement $b_t \in (0, 10]$ mB, and the processing time $\xi_t \in [10, 100]$ ms.

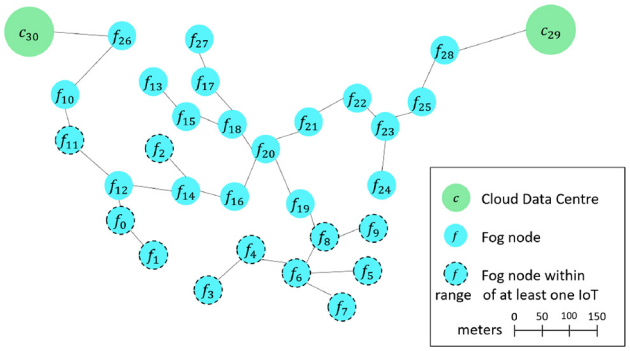

Fig. 6. Network architecture of candidate fog nodes used for comparison.

We chose to design a fog infrastructure with 29 candidate fog locations. Similar to the IoT settings, the resource capacities for fog nodes were chosen from a uniform distribution range $[a_F^r, b_F^r]$ with low and high values of $a_F^{\text{CPU}} = 0$ and $b_F^{\text{CPU}} = 25$ for CPU, $a_F^{\text{MEM}} = 0$ and $b_F^{\text{MEM}} = 125$ for MEM, and $a_F^{\text{STR}} = 0$ and $b_F^{\text{STR}} = 1000$ for STR. For each resource, at least one fog location has a resource capacity of zero; we acknowledge that a fog node may not have the capability for every resource. The network architecture of candidate fog nodes is shown in Fig. 6.

We designed the infrastructure to include two fixed cloud data centers, with capacity in $[a_C^r, b_C^r]$ with $b_F^r \ll b_C^r$ to allow Cloud nodes to accept any task not accepted by fog nodes. The low and high values selected were $a_C^{\text{CPU}} = 100\,000$ and $b_C^{\text{CPU}} = 200\,000$ for CPU, $a_C^{\text{MEM}} = 51\,200$ and $b_C^{\text{MEM}} = 102\,800$ for MEM, and $a_C^{\text{STR}} = 500\,000$ and $b_C^{\text{STR}} = 1\,000\,000$ for STR.

As noted in Fig. 3, a fog-DC node may be accessed directly by a task t if the distance is no more than γ_t, or via an access point that routes through other Internet channels. Equation (7) uses an upper bound $h_{t,w}$ on the number of hops between task t and fog-DC node w, however, these may be through access points and not the fog-DC infrastructure.

For two IP addresses, the number of hops between them for a single ping can be determined using the `traceroute` command. We tested several IPs at different distances from ourselves to estimate $h_{t,w}$ based on distance $d_{t,w}$. We found for $d_{t,w} \in (0, 10]$, $h_{t,w} \leq 5$ and for $d_{t,w} \in [100, 1000]$, $h_{t,w} \leq 14$; our findings are supported by [45]. Since all the fog node candidates are within 10 km of the tasks, we set $h_{t,f} = 5$ for $f \in F$. For Cloud nodes $c \in C$, we let $h_{t,c} = \max\{5, d_{t,c}/10\}$. The maximum number of hops between two fog-DC nodes $h_{w,w'}$ is defined as the maximum number of links for a path $\pi \in \Pi_{(w,w')}$ in the fog-DC architecture of Fig. 6.

B. Dimensioning/Partitioning Scheme Comparison

Fig. 7 shows the optimal design and dimensioning solutions obtained for Fog-DC-MILP versus Fog-DC-CG. For each model in Fig. 7, the dimensioned nodes are identified, the resource utility is detailed, and the used paths are shown. Given that the portrayed network architecture in Fig. 7 is based on assumed latitude and longitude of fog nodes, we can infer that fog nodes in similar areas are dimensioned, allowing

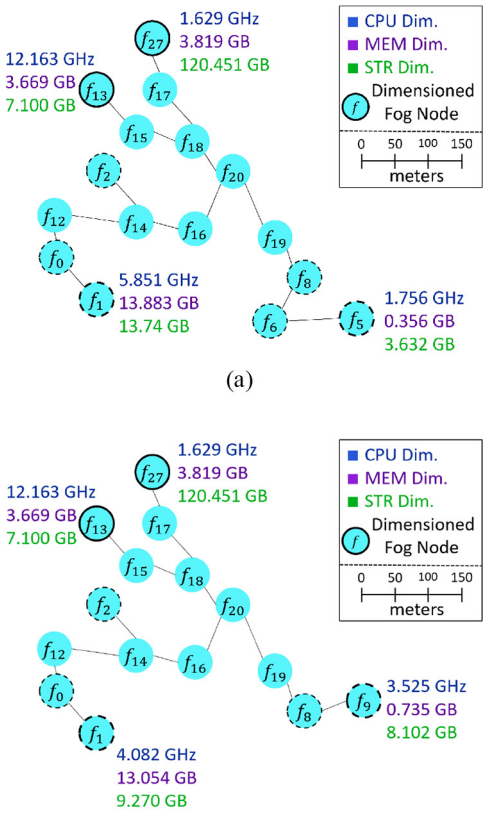

Fig. 7. Mapping solution for the MILP and CG formulation over five requests and ten tasks. (a) Fog-DC-MILP. (b) Fog-DC-CG.

TABLE V
DIMENSIONING SOLUTIONS OF FOG-DC-MILP VERSUS FOG-DC-CG (FIG. 7) FOR FOG NODES IN SIMILAR AREAS

Fog	CPU	MEM	STR
f_1 (MILP)	5.851 GHz	13.883 GB	13.774 GB
f_1 (CG)	4.082 GHz	13.054 GB	9.270 GB
f_5 (MILP)	1.756 GHz	0.356 GB	3.632 GB
f_9 (CG)	3.525 GHz	0.735 GB	8.102 GB

services to be provided to IoT devices in the same regions with real-time responses.

Considering the fog nodes with differing dimensioning solutions in Table V, we observe the similarities in both solutions with the smallest difference being in MEM allocation and the largest difference in STR allocation. Based on our map scale in Figs. 6 and 7, f_5 and f_9 are approximately 25 m apart, whereas f_1 is approximately 500 m from both; this larger distance is still close enough to allow for the benefits of minimal latency between an IoT device that sits 500 m from any fog node, making the fog-DC-CG solution of allocating more STR resources in f_9 instead of f_1 a reasonable change that does not change the latency viability of our current configuration and IoT traffic predictions; however, this change may affect the IoT traffic that is higher than the ρ-percentile of data for which we have accounted, or new IoT devices on the edge of our designed area.

TABLE VI
COST COMPARISON IN PERCENTAGE BETWEEN FOG-DC-MILP AND FOG-DC-CG

No. of Tasks	Fog-DC-MILP Solution	Fog-DC-CG Solution	% Gap	Gap
10	214.977	220.866	2.739%	5.889
20	402.828	411.208	2.080%	8.380
30	590.348	596.716	1.078%	6.368
40	773.547	783.713	1.314%	10.166
50	958.205	968.259	1.049%	10.051
60	1144.030	1153.480	0.826%	9.150
70	1331.56	1342.040	0.787%	10.480
80	1518.440	1529.040	0.698%	10.600

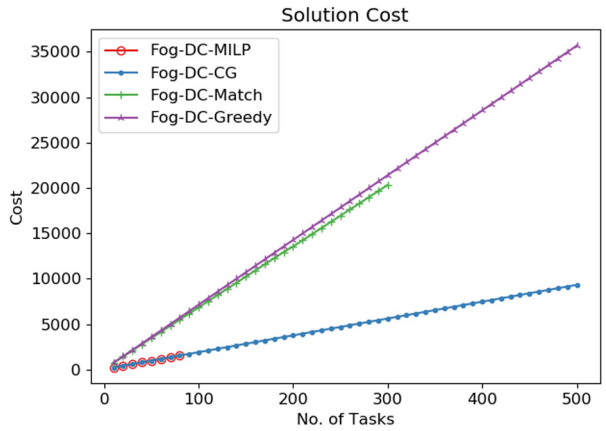

Fig. 8. Objective cost comparison of Fog-DC-MILP, Fog-DC-CG, Fog-DC-Match, and Fog-DC-Greedy.

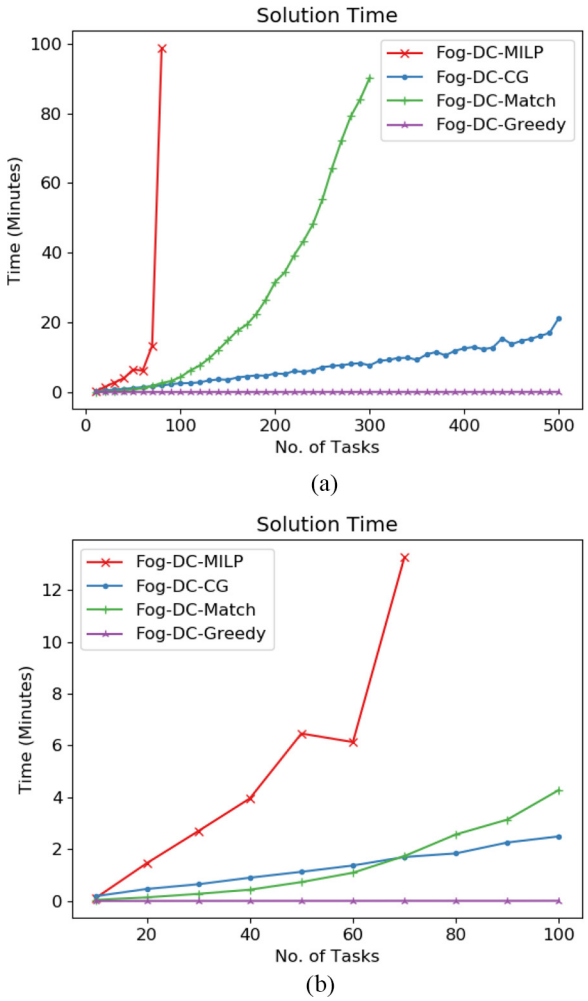

Fig. 9. Solution time comparison of Fog-DC-MILP and Fog-DC-CG. (a) Full scale. (b) Small scale.

C. Dimensioning Cost

By increasing the arrival rate of each IoT request class, we can increase the total number of tasks arriving to the fog-DC system to compare the four models given greater IoT traffic. We are most interested in whether Fog-DC-CG can approximate the optimal MILP model in cost with a substantial decrease in time. Table VI shows that Fog-DC-CG attains a near-optimal design and dimensioning cost of the fog Infrastructure, with the cost difference below 3% for up to 80 tasks. The absolute difference varies between 5.889 and 10.480, but does not increase monotonically with increasing IoT traffic, leading us to hypothesize that the percentage difference gradually decreases with the increasing number of tasks. A more expansive study into the cost differences of MILP and CG approaches is left for future work. Fig. 8 shows the significantly reduced cost of Fog-DC-CG compared to the other two heuristics Fog-DC-Match and Fog-DC-Greedy.

D. Computation Time

Our performance comparison of Fog-DC-MILP and Fog-DC-CG in Fig. 9 shows that Fog-DC-CG calculates a dimensioning solution in significantly reduced time. While Fog-DC-Match performs moderately better than Fog-DC-MILP, the computation time is not scalable for the higher number of tasks. In Fig. 9, we executed Fog-DC-MILP for at most 80 tasks as a higher number of tasks proved computationally infeasible, whereas Fog-DC-Match becames computationally infeasible after around 300 tasks. We were able to execute Fog-DC-CG for up to 500 tasks within a practical and scalable amount of time with a near-linear time growth. Though Fog-DC-Greedy is extremely fast, Fig. 8 shows that it is also the worst performing by cost.

Fig. 9(b) is identical to Fig. 9(a) with the x-axis restricted to [0, 100]. This allows us to see the similar time performances between Fog-DC-CG, Fog-DC-Match, and Fog-DC-Greedy for the low number of tasks. We also see the point at which the three models begin to deviate from each other in performance. In both cases, Fog-DC-MILP is growing at an alarming rate from the beginning.

To better observe the performance of Fog-DC-CG, we simulated 600 independent fog-DC system configurations. Each configuration had 10–100 fog candidates in different topological organizations, 10–50 IoT requests, and 1–5 tasks per request. The latitude and longitude of the fog and IoT devices were selected uniformly in a selected region of radius of 5 km, affecting the reachability of fog nodes from IoT for each simulated instance.

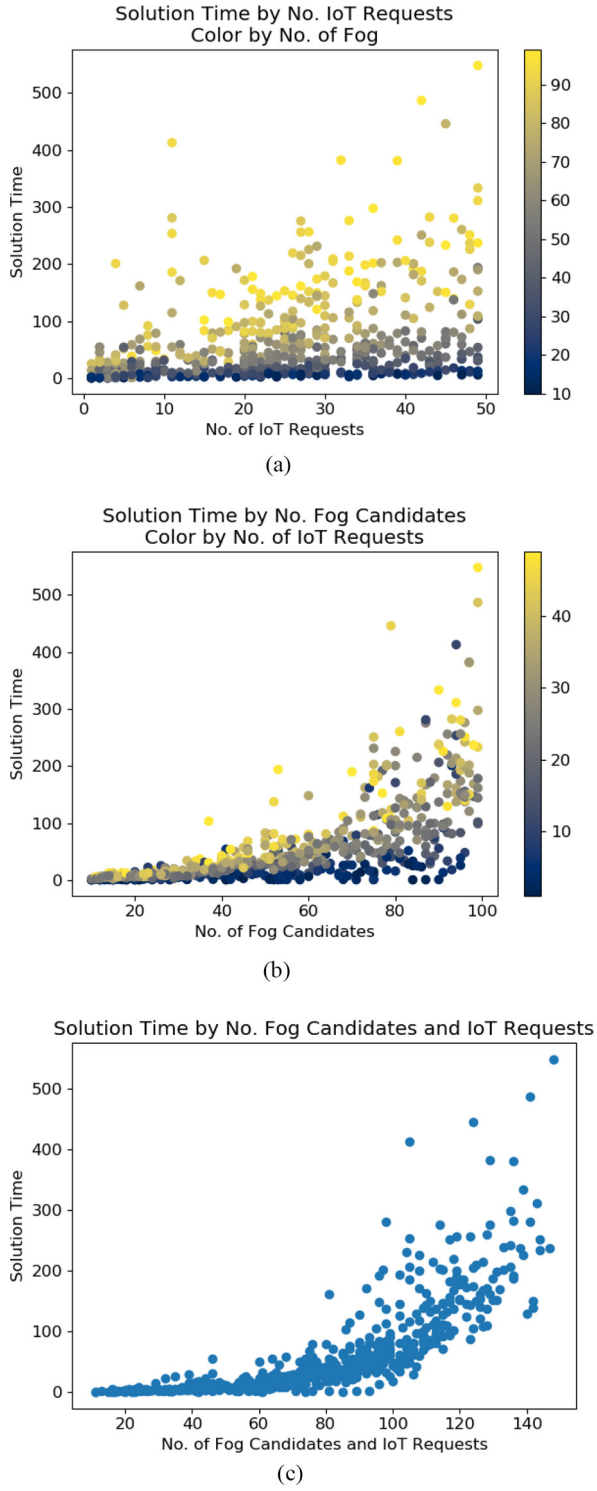

Fig. 10. Solution time of Fog-DC-CG by the number of IoT requests and fog candidates. (a) Requests with fog colormap. (b) Fogs with request colormap. (c) Fogs and requests.

The solution time per fog-DC configuration setup is shown in Fig. 10(a) and (b) with either number of IoT requests or fog candidates on the x-axis, and the other metric as a colormap. Fig. 10(a) shows that for at most 50 IoT requests, the number of requests does not have a strong influence on the solution time; no evident correlation is observed, with only the fog colormap having a clear pattern of increasing with solution time.

TABLE VII
SPEARMAN'S CORRELATION OF SIMULATED SOLUTIONS BY THE NUMBER OF REQUESTS AND FOG CANDIDATES USED

Correlation against Time	Spearman's Correlation
No. of Requests vs. Time	0.496
No. of Fog Candidates vs. Time	0.805

On the other hand, Fig. 10(b) shows that it is the number of fog candidates that largely dictates the solution time. These observations are further solidified by the correlations calculated in Table VII that shows the number of requests and of fog candidates, respectively, have a correlation of 0.478 and 0.829 with solution time. Based on the results of Fig. 10(c) with both the number of fog candidates and IoT requests, we observe some scalability with increasing fog and IoT devices, though further simulation is needed. Referring to Fig. 10, no clear linearity is observed in any of the metrics; we leave further simulation and statistical analysis for future work.

IX. CONCLUSION

In this article, we have proposed an optimal design and dimensioning formulation of the fog infrastructure using MILP to minimize infrastructure costs. To overcome scalability issues while keeping cost effectiveness, we proposed a near-optimal column generation formulation. The simulation results show that design and dimensioning solution are with under 3% difference from the optimal solution with significantly reduced computation time. Results also show column generation cost is much lower than matching-based and greedy heuristics. Simulation and analysis of fog-DC configurations conclude computation time is highly correlated with the number of fog candidates, and moderately correlated with IoT requirements.

In future work, we proposed to determine more accurate estimation of network congestion resulting from fluctuating IoT traffic by means of a simulation toolkit such as iFogSim [39], which also leads to a more accurate estimation of transmission time for all tasks in a request. Due to the intractability of MILP, we could not perform large simulations; we plan to use geographic zoning techniques to further reduce the time complexity of our MILP and CG models. We can then perform a larger scale simulation and comparison of MILP and CG models over a wider variety of current IoT and fog technologies. Our current proposition allows for the extensibility of a fog design and dimensioning scheme assuming all else remains constant; we propose to look further at the extensibility of existing/modified fog infrastructures to add greater coverage and/or resource availability for an increase in IoT traffic. Given our simulation results of computation time per fog-DC configurations [see Fig. 10(b)], our relationships are not linear; we intend to do further statistical studies with the goal of predicting the expected solution time given a set of fog-DC configurations. This would solidify our approach as a means for real and practical design, dimensioning, and future deployment.

REFERENCES

[1] *Fog Computing and the Internet of Things: Extend the Cloud to Where the Things Are*, Cisco, San Jose, CA, USA, 2015. Accessed: Apr. 2019.

[2] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the Internet of Things," in *Proc. 1st ed. MCC Workshop Mobile Cloud Comput.*, 2012, pp. 13–16.

[3] O. Skarlat, S. Schulte, M. Borkowski, and P. Leitner, "Resource provisioning for IoT services in the fog," in *Proc. IEEE 9th Int. Conf. Service Orient. Comput. Appl. (SOCA)*, 2016, pp. 32–39.

[4] M. Asif-Ur-Rahman *et al.*, "Towards a heterogeneous MIST, fog, and cloud based framework for the Internet of healthcare things," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4049–4062, Jun. 2019.

[5] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of Things for smart cities," *IEEE Internet Things J.*, vol. 1, no. 1, pp. 22–32, Feb. 2014.

[6] J. Jin, J. Gubbi, S. Marusic, and M. Palaniswami, "An information framework for creating a smart city through Internet of Things," *IEEE Internet Things J.*, vol. 1, no. 2, pp. 112–121, Apr. 2014.

[7] B. Cheng, G. Solmaz, F. Cirillo, E. Kovacs, K. Terasawa, and A. Kitazawa, "*FogFlow*: Easy programming of IoT services over cloud and edges for smart cities," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 696–707, Apr. 2018.

[8] B. Li, Z. Fei, and Y. Zhang, "UAV communications for 5G and beyond: Recent advances and future trends," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2241–2263, Apr. 2019.

[9] M. A. A. Faruque and K. Vatanparvar, "Energy management-as-a-service over fog computing platform," *IEEE Internet Things J.*, vol. 3, no. 2, pp. 161–169, Apr. 2016.

[10] S. Yangui *et al.*, "A platform as-a-service for hybrid cloud/fog environments," in *Proc. IEEE Int. Symp. Local Metropolitan Area Netw. (LANMAN)*, 2016, pp. 1–7.

[11] C. T. Do *et al.*, "A proximal algorithm for joint resource allocation and minimizing carbon footprint in geo-distributed fog computing," in *Proc. IEEE Int. Conf. Inf. Netw. (ICOIN)*, 2015, pp. 324–329.

[12] E. Saurez, K. Hong, D. Lillethun, U. Ramachandran, and B. Ottenwälder, "Incremental deployment and migration of geo-distributed situation awareness applications in the fog," in *Proc. 10th ACM Int. Conf. Distrib. Event Based Syst.*, 2016, pp. 258–269.

[13] L. Gu, D. Zeng, S. Guo, A. Barnawi, and Y. Xiang, "Cost efficient resource management in fog computing supported medical cyber-physical system," *IEEE Trans. Emerg. Topics Comput.*, vol. 5, no. 1, pp. 108–119, Jan.–Mar. 2015.

[14] G. L. Santos *et al.*, "Analyzing the availability and performance of an e-Health system integrated with edge, fog and cloud infrastructures," *J. Cloud Comput.*, vol. 7, no. 1, p. 16, 2018.

[15] S. S. Gill, R. C. Arya, G. S. Wander, and R. Buyya, "Fog-based smart healthcare as a big data and cloud service for heart patients using IoT," in *Proc. Int. Conf. Intell. Data Commun. Technol. Internet Things*, 2018, pp. 1376–1383.

[16] M. I. Naas, P. R. Parvedy, J. Boukhobza, and L. Lemarchand, "iFogstor: An IoT data placement strategy for fog infrastructure," in *Proc. IEEE 1st Int. Conf. Fog Edge Comput. (ICFEC)*, 2017, pp. 97–104.

[17] V. B. C. Souza, W. Ramírez, X. Masip-Bruin, E. Marín-Tordera, G. Ren, and G. Tashakor, "Handling service allocation in combined fog-cloud scenarios," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2016, pp. 1–5.

[18] V. Cardellini, V. Grassi, F. L. Presti, and M. Nardelli, "On QoS-aware scheduling of data stream applications over fog computing infrastructures," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, 2015, pp. 271–276.

[19] M. Aazam and E.-N. Huh, "Dynamic resource provisioning through fog micro datacenter," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops (PerCom Workshops)*, 2015, pp. 105–110.

[20] O. Skarlat, M. Nardelli, S. Schulte, M. Borkowski, and P. Leitner, "Optimized IoT service placement in the fog," *Service Oriented Comput. Appl.*, vol. 11, no. 4, pp. 427–443, 2017.

[21] K. Intharawijitr, K. Iida, and H. Koga, "Analysis of fog model considering computing and communication latency in 5G cellular networks," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops (PerCom Workshops)*, 2016, pp. 1–4.

[22] H. R. Arkian, A. Diyanat, and A. Pourkhalili, "MIST: Fog-based data analytics scheme with cost-efficient resource provisioning for IoT crowdsensing applications," *J. Netw. Comput. Appl.*, vol. 82, pp. 152–165, Mar. 2017.

[23] C. Yu, B. Lin, P. Guo, W. Zhang, S. Li, and R. He, "Deployment and dimensioning of fog computing-based Internet of Vehicle infrastructure for autonomous driving," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 149–160, Feb. 2019.

[24] M. Sookhak *et al.*, "Fog vehicular computing: Augmentation of fog computing using vehicular cloud computing," *IEEE Veh. Technol. Mag.*, vol. 12, no. 3, pp. 55–64, Sep. 2017.

[25] X. Hou, Y. Li, M. Chen, D. Wu, D. Jin, and S. Chen, "Vehicular fog computing: A viewpoint of vehicles as the infrastructures," *IEEE Trans. Veh. Technol.*, vol. 65, no. 6, pp. 3860–3873, Jun. 2016.

[26] Z. Zhou, H. Liao, X. Zhao, B. Ai, and M. Guizani, "Reliable task offloading for vehicular fog computing under information asymmetry and information uncertainty," *IEEE Trans. Veh. Technol.*, vol. 68, no. 9, pp. 8322–8335, Sep. 2019.

[27] C. Chang, S. N. Srirama, and R. Buyya, "Indie fog: An efficient fog-computing infrastructure for the Internet of Things," *Computer*, vol. 50, no. 9, pp. 92–98, 2017.

[28] W. Zhang, Z. Zhang, and H.-C. Chao, "Cooperative fog computing for dealing with big data in the Internet of Vehicles: Architecture and hierarchical resource management," *IEEE Commun. Mag.*, vol. 55, no. 12, pp. 60–67, Dec. 2017.

[29] M. Taneja and A. Davy, "Resource aware placement of IoT application modules in fog-cloud computing paradigm," in *Proc. IFIP/IEEE Symp. Integr. Netw. Service Manag. (IM)*, 2017, pp. 1222–1228.

[30] Y. Xia, X. Etchevers, L. Letondeur, T. Coupaye, and F. Desprez, "Combining hardware nodes and software components ordering-based heuristics for optimizing the placement of distributed IoT applications in the fog," in *Proc. 33rd Annu. ACM Symp. Appl. Comput.*, 2018, pp. 751–760.

[31] M. Aazam and E.-N. Huh, "Fog computing micro datacenter based dynamic resource estimation and pricing model for IoT," in *Proc. IEEE 29th Int. Conf. Adv. Inf. Netw. Appl.*, 2015, pp. 687–694.

[32] A. Yousefpour *et al.*, "QoS-aware dynamic fog service provisioning," 2018. [Online]. Available: arXiv:1802.00800.

[33] S. Tuli, R. Mahmud, S. Tuli, and R. Buyya, "FogBus: A blockchain-based lightweight framework for edge and fog computing," *J. Syst. Softw.*, vol. 154, pp. 22–36, Aug. 2019.

[34] E. Yigitoglu, M. Mohamed, L. Liu, and H. Ludwig, "Foggy: A framework for continuous automated IoT application deployment in fog computing," in *Proc. IEEE Int. Conf. AI Mobile Services (AIMS)*, 2017, pp. 38–45.

[35] B. Donassolo, I. Fajjari, A. Legrand, and P. Mertikopoulos, "Fog based framework for IoT service provisioning," in *Proc. 16th IEEE Annu. Consum. Commun. Netw. Conf. (CCNC)*, 2019, pp. 1–6.

[36] L. Ni, J. Zhang, C. Jiang, C. Yan, and K. Yu, "Resource allocation strategy in fog computing based on priced timed Petri nets," *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1216–1228, Oct. 2017.

[37] W. Zhang, Z. Zhang, S. Zeadally, H.-C. Chao, and V. C. M. Leung, "MASM: A multiple-algorithm service model for energy-delay optimization in edge artificial intelligence," *IEEE Trans. Ind. Informat.*, vol. 15, no. 7, pp. 4216–4224, Jul. 2019.

[38] W. Zhang, Z. Zhang, S. Zeadally, and H.-C. Chao, "Efficient task scheduling with stochastic delay cost in mobile edge computing," *IEEE Commun. Lett.*, vol. 23, no. 1, pp. 4–7, Jan. 2019.

[39] H. Gupta, A. V. Dastjerdi, S. K. Ghosh, and R. Buyya, "iFogSim: A toolkit for modeling and simulation of resource management techniques in the Internet of Things, edge and fog computing environments," *Softw. Pract. Exp.*, vol. 47, no. 9, pp. 1275–1296, 2017.

[40] F. A. Salaht, F. Desprez, A. Lebre, C. Prud'Homme, and M. Abderrahim, "Service placement in fog computing using constraint programming," in *Proc. IEEE Int. Conf. Services Comput.*, 2019, pp. 19–27.

[41] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Comput. Netw.*, vol. 54, no. 15, pp. 2787–2805, 2010.

[42] J. F. Shortle, J. M. Thompson, D. Gross, and C. M. Harris, *Fundamentals of Queueing Theory*. Hoboken, NJ, USA: Wiley, 2018, pp. 73–90.

[43] J. S. Witte and R. S. Witte, *Statistics*, 11th ed. Hoboken, NJ, USA: Wiley, 2016.

[44] B. Latré, P. De Mil, I. Moerman, N. Van Dierdonck, B. Dhoedt, and P. Demeester, "Maximum throughput and minimum delay in IEEE 802.15.4," in *Proc. Int. Conf. Mobile Ad Hoc Sensor Netw.*, 2005, pp. 866–876.

[45] Y. Wang, P. Lai, and D. Sui, "Mapping the Internet using GIS: The death of distance hypothesis revisited," *J. Geograph. Syst.*, vol. 5, no. 4, pp. 381–405, 2003.

[46] D. G. Andersen. *Theoretical Approaches to Node Assignment*. Accessed: Jun. 30, 2018. [Online]. Available: https://kilthub.cmu.edu articlesTheoretical_Approaches_to_Node_Assignment6610829/1, doi: 10.1184/R1/6610829.v1.

[47] C. Barnhart, E. L. Johnson, G. L. Nemhauser, M. W. Savelsbergh, and P. H. Vance, "Branch-and-price: Column generation for solving huge integer programs," *Oper. Res.*, vol. 46, no. 3, pp. 316–329, 1998.

[48] A. Federgruen and H. Groenevelt, "The greedy procedure for resource allocation problems: Necessary and sufficient conditions for optimality," *Oper. Res.*, vol. 34, no. 6, pp. 909–918, 1986.