Service-based, Multi-Provider, Fog Ecosystem with Joint Optimization of Request Mapping and Response Routing

Mays AL-Naday, *Member, IEEE,* Nikolaos Thomos, *Senior Member, IEEE,* Jiejun Hu, Bruno Volckaert, *Senior Member, IEEE,* Filip de Turck, *Fellow, IEEE,* Martin J. Reed *Member, IEEE*

***Abstract*—Digital transformation is increasingly reliant on *service*-based operations in fog networks. The latter is a geo-dispersed form of the cloud, extending resources closer to end-users for improved privacy and reduced latency. The dispersion leverages diversity of compute-network capacities and energy prices, while promotes the coexistence of multiple providers. This drives variation in operational cost, coupled with limited information sharing across providers. Consequently, there is a critical need for an orchestration solution that preserves autonomy and optimizes operational cost across domains, while meeting service requirements. This paper proposes a novel service-based fog management and network orchestrator (sbMANO), which utilizes service metadata in enabling multi-provider resource management. The sbMANO is empowered with a novel optimization algorithm for service-based joint request mapping and response routing. The algorithm acts on partial information and preserves the edge for delay-critical services. The performance of the algorithm is evaluated analytically for *delay-aware* and *delay-agnostic* variants. The results show that both achieve near-optimal performance in maximizing user satisfaction with minimum operational cost. Furthermore, the delay-aware variant outperforms the agnostic counterpart, with higher user satisfaction and lower operational cost.**

***Index Terms*—Fog computing networks, service-based networking, joint optimization, request mapping, response routing, service management**

I. Introduction

FOG computing extends the cloud closer to end-users, to enable digital transformation [1]–[4]. The reference architecture of [5] defines fog nodes as data centers (DC) of variant size, ranging from small nano ones at the edge to large cloud counterparts closer to the core. The variation of capacity among fog nodes and the likely autonomy of the fog provider from the network operator, causes challenges in meeting service level agreements (SLAs) while optimizing operational costs. On one hand, the constrained capacity and connectivity of edge data centers increases susceptibility to resource straining, leading to higher processing and transmission latency. Managing latency across autonomous entities is not a trivial task. Because, it typically requires sharing state information, deemed sensitive by providers.

On the other hand, current indicators suggest that the operational cost of a small DC is higher than that of a large cloud counterpart. This is driven by the difference in wholesale offers of energy [6], and the lower efficiency of small DCs in utilizing power [7], [8]. A qualitative study by the European Union (EU) supports this argument, as it shows the Power Utilization Effectiveness (PUE) is better in medium to large data centers than small counterparts [9], [10]. Moreover, the work of [11], [12] reveal that only a marginal $3-8\%$ of total energy consumption is incurred by communications. This illustrates that reducing communications by utilizing the edge has a limited benefit in energy saving.

The argument above does not preclude the advantages of the edge in latency reduction and privacy preservation. It elucidates the need for optimized resource management based on service characteristics, in addition to capacity constraints. So far, this need

M. AL-Naday, N. Thomos, and M. Reed are with the school of Computer Science and Electronic Engineering, University of Essex, UK. Email: mfhaln@essex.ac.uk

J. Hu is with the Lancaster University Leipzig, Germany.

B. Volckaert and F. de Turck are with IDLab, Department of Information Technology (intec), Ghent University - imec, Belgium

978-1-5386-5541-2/18/$31.00

has not been met by existing solutions. Instead, the edge is generally prioritized in workload allocation, to minimize network latency (examples [13], [14]).

However, this work argues that a nondiscriminatory allocation to the edge - irrespective of application needs - can strain resources with latency tolerant applications. This risks negating the sought advantages of this scarce infrastructure. Not to mention the side effect in reducing the consistency of workload offered to the cloud, exposing it to higher variation. Orthogonally, higher workload at the edge translates to higher energy expenditure, resulting in sub-optimal operational cost without significant energy savings. Instead, allocation on need-basis can promote the advantages of the edge and reduce the operational costs.

This work addresses the need above by proposing novel, decentralized: service-based fog Management and Network Orchestrator (sbMANO); and, service-based *Alternating Direction Method of Multipliers (sbADMM)* algorithm. Together, they provide joint request mapping and response routing in a multi-provider fog ecosystem. The sbMANO facilitates the exposure of applications and resources as attributed ***services***. The fog and network providers utilize the capability to share only the cost of their resources with each other. Each side utilizes the other's cost with their internal information in sbADMM, to make optimized decisions. The main contributions of this work are:

- First, we introduce the fog sbMANO, incorporating novel discovery and mapping services.
- Second, we model the fog ecosystem and formulate the problem of service-based request mapping and response routing.
- Third, we develop the novel sbADMM, having partial information sharing between the fog and network providers. We evaluate the algorithm analytically and show that it achieves near optimal performance in maximizing users' satisfaction with minimum operational cost.

The remainder of this paper is structured as follows: Section II reviews state of the art related work, while Section III introduces the proposed sbMANO within a fog ecosystem. Section IV models the fog ecosystem and presents the problem formulation. Section V describes the proposed sbADMM algorithm, while Section VI presents the analytical evaluation of the algorithm. Finally, Section VII draws the conclusions and outlines future work.

II. Related Work

Cloud-based ecosystems have been studied extensively in the literature [3], [4], [15]–[17]. So far, the proposed solutions for resource management either focus on homogeneous DC networks, or favor the edge in workload allocation. The work of [13], [18] tackles the problem of joint request mapping and response routing in large DC networks. Their solution ignores constraints on computing capacity when minimize the propagation latency and energy cost. This results in prioritizing closer nodes to end-users for workload allocation. The work of [14] tackles the same problem in content distribution networks, and the solution similarly favors closer nodes. In a fog ecosystem with the constrained edge being the closest infrastructure, these solutions risk straining edge resources. Rather than minimizing latency, this work introduces awareness of response time requirements per service. Consequently, preserving the edge for demand of delay-critical services, while allocating delay-tolerant counterparts to farther points.

The works of [19], [20] define the fog as the middle layer of infrastructure, between the cloud and the edge. The work of [19] addresses the problem of service placement in large-scale, volatile, edge networks. They propose a greedy algorithm for minimizing the distance between fog and edge nodes, constrained by the compute capacity of fog nodes and their responsibility area. The work of [20] addresses the problem of service placement and request routing in Mobile Edge Computing (MEC) networks. They propose an offloading solution that takes into account compute and storage constraints of MEC nodes. Both solutions prioritize closer nodes irrespective of service needs. Additionally, both solutions are expensive as they follow a centralized approach that assumes full knowledge of compute-network state. This limits applicability to scenarios where the fog and network resources are managed by a single provider. In contrast, this work proposes a decentralized solution that relies on sharing cost information only, thereby suitable for both single and multi-provider fog ecosystems.

Other work considers both the edge and the cloud. The work of [21] solves the problem of fog node planning and workload allocation, using particle swarm optimization. The work of [22] proposes a solution for minimizing the response time when

selecting fog node(s) for offloading. Both solutions follow a two-stage offloading approach, as workload is first allocated to fog nodes and only offloaded to the cloud when fog resources run low. In contrast, the sbADMM algorithm proposed here does not suffer from this sub-optimality, as it incorporates response time awareness in one-stage workload distribution across the spectrum of edge to cloud nodes.

Furthermore, although strands of the work above are at the granularity of services, they do not utilize service popularity metrics to estimate the demand per service and utilize it in planning resource allocations. However, existing work in DC networks such as [23] show that considering service popularity for demand prioritization offers significant gains.

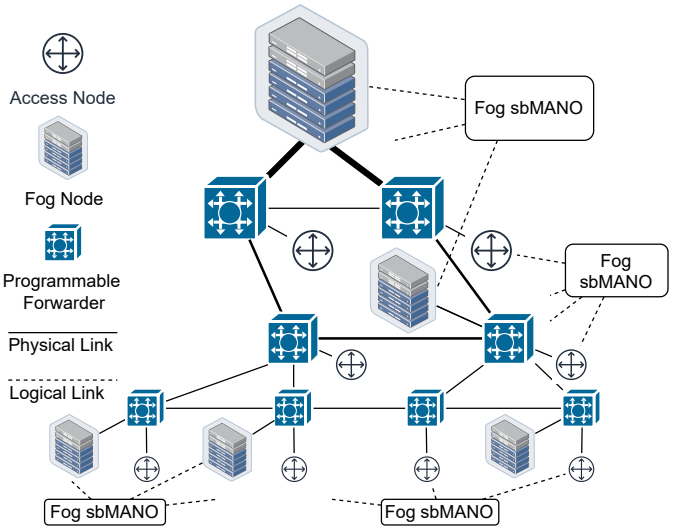


Fig. 1. High-level view of a service-based fog ecosystem, showing the decentralized sbMANO at the fog provider and access sides.

III. Proposed Fog service-based MANO (sbMANO)

A. Preliminary on the Fog Ecosystem

The fog ecosystem, illustrated in Figure 1, follows the OpenFog reference architecture [5]. It has a tier-based hierarchical distribution; ranging from a few large data centers at Tier-1 (cloud nodes) to a large number of nano data centers at Tier-n (edge nodes). Cloud nodes have virtually unlimited capacity, while edge nodes have constrained counterpart. Fog nodes are connected by an underlying programmable network of forwarders. They follow a similar hierarchy to that of fog nodes. Core links of high bandwidth capacity and long distance interconnect Tier-1 cloud nodes. While edge links of constrained bandwidth and short distance connect Tier-n edge nodes. Vertical links connect fog nodes of different tiers, while horizontal links connect nodes of the same tier. The ecosystem may have one provider managing all infrastructure, or it may have a separate fog provider and network operator. This work assumes separate providers, with fog nodes being managed by the fog provider, while links, forwarders and access nodes are managed by the network operator.

B. The Fog sbMANO

The ecosystem is controlled by a multi-provider decentralized sbMANO, shown in Figure 1. It offers service-based resource management and orchestration, particularly *service discovery* and *demand mapping*. The structure of the sbMANO (Figure 2)

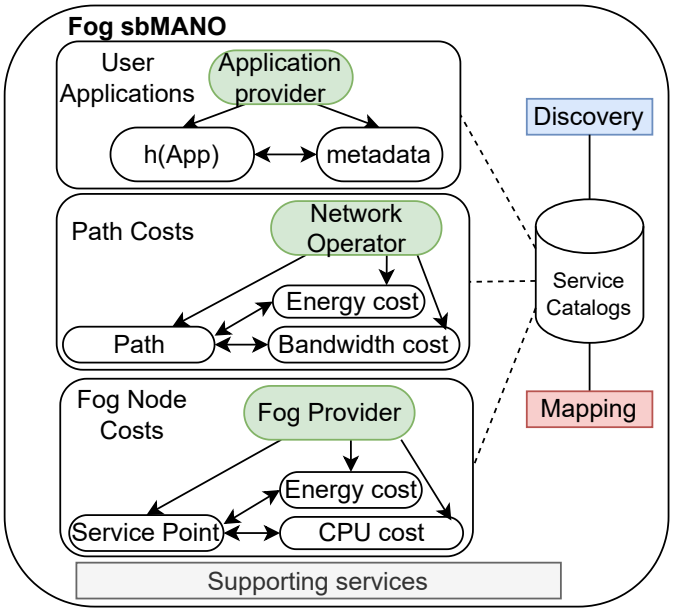


Fig. 2. A simplified view of the Fog sbMANO showing how the catalogs expose metadata to the discovery and mapping services. The figure further show reliance on existing, supporting, services.

has the flexibility to operate with different configurations for co-existing providers. This work presents an example of two infrastructure providers: one manages the computing side while the other manages the access side. Each shares cost information with the other for coordinated decision-making.

Service discovery is proposed through a publish/subscribe approach such as that of [24], for exposing and disseminating services and their metadata. Here, service identifiers and metadata are

published and subscribed to in *service catalogs*, accessible by the discovery service. Services can be any commodity offered on demand. Figure 2 illustrates an example of three services: user applications, network paths and compute nodes. Application providers publish metadata as part of their application offering to the fog provider. The latter decides on application deployment in fog nodes, based on user demand and requirements extracted from the metadata.

The metadata model depends on the ecosystem. Here, we assume three categories of information: resource requirements, QoS thresholds and popularity. Application popularity is determined by the fog provider following demand analysis. So far, analysis of workload observed in public clouds suggests that application popularity follows a power law [25], [26]. Orthogonally, the fog provider subscribes to paths offered by the network, with path costs as the metadata. Equivalently, the network operator subscribes to fog nodes offered by the fog provider, having node costs as the metadata.

The ***mapping service*** resolves a request for application to a service point, hosted by a fog node. It is comparable to the Domain Name Service (DNS). However, unlike DNS, the proposed mapping service decouples service and location identifiers, while linking them in a many-to-many mapping base (illustrated in Figure 3). The latter can be updated periodically and on event basis, using an optimization algorithm that solves the problem of request mapping and response routing. Hence, the mapping base is customized for each access node, depending on the demand observed by the node.

The problem of ***request mapping and response routing*** is treated here as a load management optimization, similar to that presented in [13]. It focuses on the trade-off between operational cost (OPEX) and performance in a heterogeneous fog ecosystem. This includes the diversity of: energy prices, compute-bandwidth capacities and link length. The problem cannot be solved centrally without disclosing internal state information. However, this is not a feasible option typically, because providers perceive such disclosure as revealing sensitive knowledge to competitors. Furthermore, with the geographic dispersion comes the need for autonomous optimization, to allow for operation continuity within the risk of disconnectivity. Therefore, this work takes a decentralized approach in solving the problem.

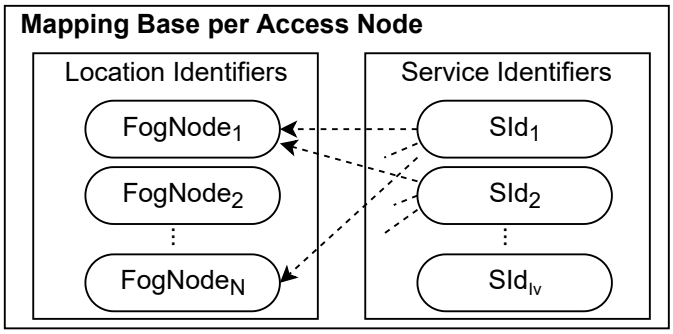


Fig. 3. Schematic view of the proposed mapping base per access node, maintained by the mapping service after the joint optimization decides the fraction of application demand to be mapped to service point at each fog node.

Specifically, developing a novel decentralized algorithm based on ADMM [27]. The algorithm allows for autonomy in solving the problem, by decoupling the problem objectives and constraints with respect to their owners. Each solves the problem from their side and share their decision with the other.

Request mapping is treated as an access problem, where each access node determines the fraction of service demand mapped to a fog node. While response routing is a fog provider problem, where each fog node decides the fraction of service demand to be processed and response routed back to the access node. The two sides of the problem are solved independently and each side share their results with the other. The process is repeated until convergence, then followed by updating the mapping base of each instance of the sbMANO.

Notably, we assume the path between each pair of access and fog nodes has been calculated separately by the network operator and provided to both nodes. Hence, the problem of path calculation is outside the scope of this work. This does not preclude that paths share common links, thus competition over bandwidth resources may occur. The next section introduces the model of the fog ecosystem and formulates the two-side problem ahead of introducing the novel sbADMM algorithm.

IV. The Ecosystem Model

A summary of the model notations is provided in Table I.

A. The Network

Recall that the ecosystem is assumed to follow a tier-based hierarchical structure. For simplicity

TABLE I
SUMMARY OF NOTATIONS

Notation	Definition
$\mathcal{V}$, $\mathcal{N}$	Set of forwarders and fog nodes in the ecosystem
v, n	single access node, single fog node
e, $\mathcal{E}$	single edge, set of edges connecting forwarders
w_{uv}, l_{uv}	bandwidth capacity on edge e_{uv}, length of e_{uv}
p_{nv}	directional path in set of links from n to v
c_n, γ_n	CPU capacity, CPU energy price per fog node
i, $\mathcal{I}$	single service, set of services in the ecosystem
$\mathcal{N}^i$	subset of $\mathcal{N}$ hosting service points of i
q^i, c^i, r^i	request data, task, response data sizes of i
d^i	maximum response time tolerance of i
c_n^i	CPU capacity of service point of i at fog node n
λ_v^i	request rate of service i at access node v
δ_v^i	computation workload of service i by v's demand
$\omega_v^{i,q}, \omega_v^{i,r}$	request and response traffic of i by v's demand
γ_n	CPU energy price of fog node n
γ_{nv}	bandwidth energy price on the path from n to v
θ_n, θ_{nv}	CPU price of n, bandwidth price of path p_{nv}
τ_n^i	processing latency of i service point at n
τ_{nv}^i	transmission latency of i's traffic on path p_{nv}
α_{vn}^i	mapping decision variable
β_{nv}^i	response routing decision variable
t	iteration step of the algorithm
μ_{vn}^i, ρ	the Lagrange multiplier, the penalty parameter
σ^i	the Lagrange multipliers of the access problem
χ^i	the Lagrange multipliers of the fog problem
s_{pri}, s_{dual}	residual parameters of the algorithm
$\epsilon_{pri}, \epsilon_{dual}$	stopping criterion of the algorithm
H_p	sum of ratios of links' bandwidth on path p

without loss of generality, the rest of this work assumes a three-tier ecosystem. The forwarding network is modeled as a set of vertices $\mathcal{V},\ V = |\mathcal{V}|$, interconnected by a set of edges $\mathcal{E} = \{e_{uv} | u, v \in V, u \neq v\}$. Each edge $e_{uv} \in \mathcal{E}$ is characterized by $\langle w_{uv}, l_{uv} \rangle$, w_{uv} is the bandwidth capacity in Mbps and l_{uv} is the length in meters. A set of fog nodes $\mathcal{N},\ N = |\mathcal{N}|, N \leq V$ are co-located with a subset of forwarders. A fog node may be: a large tier-1 cloud node, a teir-2 cloudlet of moderate capacity (micro DC), or tier-3 small edge node (nano DC). Each $n \in \mathcal{N}$ is characterized by a tuple $\langle c_n, \gamma_n \rangle$, where c_n is the CPU capacity in MIPS and γ_n is the energy cost in Penny per MIPS (PpMIPS).

Each forwarder $v \in \mathcal{V}$ connects an access node, which acts as a gateway that connects end-users to the fog ecosystem. The access node receives users' requests for applications and map them to fog nodes, as well as relays the response back to users. The number of users connected to an access node depends on its resources at the user-facing side. This includes physical layer resources, such as spectrum availability for wireless communications. Notably, users connect to their access node using a wide range of technologies and this is usually separated from the edge-to-core network, considered in this work. Consequently, managing the access resources is outside the scope of this work and does not impact its outcomes. Nevertheless, we assume access resources are optimized separately and the solution provides a setting of the number of connected users.

Each access-fog nodes pair (v, n) are likely to be of the same tier. They are provided with request and response forwarding paths, p_{vn} and p_{nv} respectively, pre-established by the network operator. Paths may traverse common links, hence sharing bandwidth capacity. This is generally the case when the network implements link/node-based forwarding solutions, such as [28], [29]. The bandwidth capacity on a path is constrained by the thinnest link, i.e. $w_p = \min(\{w_e | e \in p\})$. The access node of v is aware of p_{vn}, p_{nv} length and bandwidth availability, while fog node n is only aware of their costs. Equivalently, the access node of v only knows the CPU cost of n, but not its capacity.

B. Services

A set of services $\mathcal{I},\ I = |\mathcal{I}|$ is exposed in the ecosystem. For simplicity, without loss of generality, we focus on user application type of services. Each service $i \in \mathcal{I}$ is identified by a resource tuple $\langle q^i, c^i, r^i \rangle$ and a QoS parameter, d^i. The resource tuple specifies: the size of request data (q^i bits), the size of computation task of the service (c^i MIPS) and the size of response data (r^i bits). The QoS parameter, d^i, identifies the latency threshold of the service in seconds, that is the maximum response time tolerance. Furthermore, the fog provider may extend the profile of a service i with such metrics as popularity distribution and ranking; to facilitate demand estimation as modeled in Section IV-D.

C. Service Points

Each fog node $n \in \mathcal{N}$ hosts service point(s) for one or multiple services. A service point of $i \in \mathcal{I}$ is a virtual resource (i.e., a virtual machine, a container or a mixture of both) that process requests for i. It is worth noting that multiple instances of a virtual machine or a container of i running on n would still be exposed as a single service point.

This is based on the assumption that a load balancer exists to receive requests for i and delegate them to the appropriate instance. $\mathcal{N}^i \subseteq \mathcal{N}$ is defined as a subset of fog nodes hosting service points of i. A service point of i hosted at n is characterized by c_n^i, the compute capacity of the service point in MIPS. The capacity of a service point is limited by the total capacity of the fog node, c_n, formulated in the constraint below:

$$\sum_{i \in \mathcal{I}} c_n^i \leq c_n \tag{1}$$

D. Service Demand and Decision Variables

During a defined time slot, each access node receives a number of requests for a subset of services. We define λ_v^i as the average request rate per second for service i, received by the access node connected to $v \in \mathcal{V}$. λ_v^i is assumed to depend on the popularity of the service. So far, models suggest that the popularity of realistic cloud applications follows a Zipf distribution [25], [26]. This does not restrict the ecosystem model. However, the distribution affects the expected volume of demand per service. Because, λ_v^i translates into computation and upload-download communication demands, δ_v^i and $\omega_v^{i,q}$, $\omega_v^{i,r}$ respectively defined as:

$$\delta_v^i = \lambda_v^i \cdot c^i \tag{2}$$
$$\omega_v^{i,q} = \lambda_v^i \cdot q^i \tag{3}$$
$$\omega_v^{i,r} = \lambda_v^i \cdot r^i \tag{4}$$

An access node of $v \in \mathcal{V}$ with demand for service i decides $\alpha_{vn}^i \in [0, 1]$, the fraction of its demand for i to be mapped to fog node $n \in \mathcal{N}^i$. Equivalently, n decides $\beta_{nv}^i \in [0, 1]$ fraction of v's demand for i to be processed and response routed back to v. It is desired that α_{vn}^i should equal β_{nv}^i, indicating that all the mapped demand is admitted for processing.

E. Operational Cost (OPEX)

This work focuses on two key cost drivers in a fog ecosystem: *energy* and *resource scarcity*.

1) Energy: The ***computation energy*** cost, γ_n PpMIPS is derived from the wattage consumption per MIPS and Watt price in pennies per Watt. For simplicity, we assume each 1 MIPS to consume ≈ 1 Watt of power[1], hence the Wattage consumption can be directly derived from c^i MIPS, for any service $i \in \mathcal{I}$. The Watt price is dependent on the underlying infrastructure, including the energy price offered at the site. Orthogonally, each path p_{vn} or p_{nv} has a ***communication energy*** cost, γ_{vn},γ_{nv} penny per Mbps (PpMbps) respectively. This is derived from the wattage consumption for sending 1Mbps of data and Watt price. For simplicity of analysis, we assume 1 Mbps to consume $\approx$ 1Watt of power [30]. Hence, the wattage consumption on p_{vn} and p_{nv} can be derived from q^i and r^i, respectively. Notably, since optimizing the network OPEX with respect to upstream data is outside the scope of this work, γ_{vn} is not considered further here.

2) Resource Scarcity: This is an abstract, unitless, cost that indicates the *preciousness* of a resource. The higher the constraints on capacity, the more precious the node. Hence, compute resources of a large cloud node are considerably cheaper than their edge constrained counterparts. The same applies to network links, particularly in an edge-densified network. There, edge forwarders are directly connected to each other by constrained links, in addition to being connected to the core by aggregate links. As such, the cost of utilizing a core link is significantly cheaper than a constrained edge. We define θ_n as the CPU cost for executing 1 MIPS on a service point of i, hosted at fog node n. While θ_{nv} is the bandwidth cost for transmitting 1Mbps of response data on path p_{nv}, from n to v. Notably, the bandwidth cost on the request path p_{vn} is not considered, as recall that optimizing upstream data OPEX is outside the scope of this work.

F. Quality of Service: Response Time

The response time is the elapsed time between an access node sending a request to a service point and receiving a response back. The term can be split into: *processing latency* and *communication latency*. The former depends on the task size, c^i, and the processing capacity of a service point, c_n^i. The communication latency is subdivided into: *transmission latency* and *propagation latency*. Since this work focuses on a fog ecosystem with an average size of a large city, propagation latency is assumed to be negligible and the dominant term is the transmission latency. This depends on the request and response sizes, q^i and r^i, and the bandwidth capacity on the path w_p. Now, given δ_v^i and Little's Law [31], a

[1] https://ourworldindata.org/grapher/computing-efficiency

service point of i at n can be modeled as a M/M/1 queue with a service rate c_n^i/c^i. The arrival rate equal the sum of request rates mapped to n from all access nodes. Hence the processing latency is formulated as:

$$\tau_n^i = \frac{c^i}{c_n^i - \sum_{v\in\mathcal{V}} \alpha_{vn}^i \delta_v^i} \tag{5}$$

The transmission latency is incurred by the tandem queues of the request and response paths, p_{vn}, p_{nv}. Each is an interconnected series of M/M/1 queues of the on-path forwarding nodes. Hence, the latency on p_{vn} and p_{nv} can be defined as:

$$\tau_{vn}^i = \sum_{e\in p_{vn}} \frac{q^i}{w_e - \alpha_{vn}^i \omega_v^{i,q}} \tag{6}$$

$$\tau_{nv}^i = \sum_{e\in p_{nv}} \frac{r^i}{w_e - \alpha_{vn}^i \omega_v^{i,r}} \tag{7}$$

G. Problem Formulation

Given the above, the problem of service-based joint request mapping and response routing can be defined as a constrained two-cost minimization problem. It is to decide α_{vn}^i that minimizes the computing cost for the access side, and β_{nv}^i that minimizes the response communication cost for the fog side, for each $i \in \mathcal{I}, v \in \mathcal{V}$ and $n \in \mathcal{N}$. Mathematically, the problem can be formulated as:

$$\min_{\alpha,\beta} \sum_{i\in\mathcal{I}} \sum_{n\in\mathcal{N}^i} \sum_{v\in\mathcal{V}} (\delta_v^i \gamma_n \theta_n)\alpha_{vn}^i + (\omega_v^{i,r}\gamma_{nv}\theta_{nv})\beta_{nv}^i \tag{8}$$

subject to:

$$\text{C1:} \quad \sum_{n\in\mathcal{N}^i} \alpha_{vn}^i = 1, \ \forall v \in \mathcal{V} \tag{9}$$

$$\text{C2:} \quad \tau_n^i + \tau_{vn}^i + \tau_{nv}^i \le d^i \tag{10}$$

$$\text{C3:} \quad \sum_{v\in\mathcal{V}} \beta_{nv}^i \delta_v^i \le c_n^i, \ \forall i \in \mathcal{I}, \forall n \in \mathcal{N}^i \tag{11}$$

$$\text{C4:} \quad \alpha_{vn}^i \omega_v^{i,r} \le w_p, \ p = p_{nv} \tag{12}$$

$$\text{C5-a:} \quad \alpha_{vn}^i \ge 0 \tag{13}$$

$$\text{C5-b:} \quad \beta_{nv}^i \ge 0 \tag{14}$$

$$\text{C6:} \quad \alpha_{vn}^i = \beta_{nv}^i \tag{15}$$

C1 (9) ensures that all demand of an access node is allocated to a service point(s). C2 (10) is the QoS constraint, restricting allocation such that the actual response time does not exceed the service tolerance threshold. C3 (11) and C4 (12) are the CPU and bandwidth capacity constraints, which ensure that allocated demand does not exceed the capacities of the service point and the response path, respectively. C5-a(13) and C5-b(14) are the non-negativity constraints of the decision variables, α_{vn}^i and β_{nv}^i, respectively. Finally, C6 ensures the fraction of demand mapped to a fog node equals that which is admitted by the node. Notably, C1, C4 and C5-a constrain only the access side problem (i.e. α_{vn}^i), while C3 and C5-b constrain the fog part of the problem (i.e. β_{nv}^i). The constraint on p_{vn} is omitted in this work, as request data is assumed to be small enough not to cause congestion.

Next, we introduce *sbADMM*, a decentralized approximation algorithm for solving the problem of (8) with limited information sharing across sides.

V. Algorithmic Solution

Appendix A provides a preliminary on the classic ADMM algorithm.

A. Service-based ADMM (sbADMM)

ADMM cannot be applied directly to solve the problem of (8) as C2 (10) couples all variables. This means, C2 (10) cannot be met without one side revealing leverage information to the opposite side. For example, containing C2 wholly within the access problem requires an access node to calculate the processing latency. This, in turn, requires the fog provider to reveal the processing capacity and workload of each service point to the network operator. The opposite is equally true, i.e., if C2 is handled by the fog side, the latter need to calculate the transmission latency.

However, given the access side knowledge of τ_{vn}^i and τ_{nv}^i, and the fog side of τ_n^i; an alternative, leverage-preserving, approach is to decompose C2 into two parts. Each part is computed at their side and communicated to the other. The latter treats the received part as a constant value and calculates the total response time. Hence, C2 is reformulated as C2-a (access side) and C2-b (fog side). C2-a can be written as:

$$\tau_n^i + \sum_{e\in p_{vn}} \frac{q^i}{w_e - \alpha_{vn}^i \omega_v^{i,q}} + \sum_{e\in p_{nv}} \frac{r^i}{w_e - \alpha_{vn}^i \omega_v^{i,r}} \le d^i \tag{16}$$

while C2-b is written as:

$$\frac{c^i}{c_n^i - \sum_{v\in\mathcal{V}} \beta_{nv}^i \delta_v^i} + \tau_{vn}^i + \tau_{nv}^i \le d^i \tag{17}$$

Now, the augmented Lagrangian of the two-side problem can be formulated as:

$$\mathcal{L}(\alpha,\beta,\mu)=\sum_{i\in\mathcal{I}}\sum_{n\in\mathcal{N}^i}\sum_{v\in\mathcal{V}}(\delta_v^i\gamma_n\theta_n)\alpha_{vn}^i+ (\omega_v^{i,r}\gamma_{nv}\theta_{nv})\beta_{nv}^i+(\alpha_{vn}^i-\beta_{nv}^i)\mu_{vn}^i+ \rho/2(\alpha_{vn}^i-\beta_{nv}^i)^2 \tag{18}$$

Each side solves their problem and exchange the solution with the opposite. This is repeated until convergence is reached, within a predefined error gap. The access problem is a reduction of (18), including α_{vn}^i terms only. This can be decomposed into a distributable set of access-node sub-problems, having each access node solve its problem independently of the rest. As such, the access-node problem at any iteration $t+1$ can be written as :

$$\min_{\alpha_v^i}\sum_{n\in\mathcal{N}^i}\alpha_{vn}^i\Big((\delta_v^i\gamma_n\theta_n)+\mu_{vn}^{i,t}+ \rho/2(\alpha_{vn}^i-2\beta_{nv}^{i,t})\Big) \tag{19}$$

subject to: C1 (9), C2-a (16), C4 (12), C5-a (13) and C6 (15). Notice that C2-a indirectly introduces tighter bounds on the solution space than C4. Because, meeting the latency threshold require less loaded paths. Therefore, C4 need to be verified only if C2-a cannot be satisfied. Hence, the set of constraints of problem (19) at an iteration $t+1$ will include either C2-a or C4, but not both.

The fog problem is a reduction of (18), involving only β_{nv}^i. It can be decomposed into a set of fog-node sub-problems, having each node solve its problem independently from others. Hence, at step $t+1$ the minimization problem can be written as:

$$\min_{\beta_n^i}\sum_{v\in\mathcal{V}}\beta_{nv}^i\Big((\omega_v^{i,r}\gamma_{nv}\theta_{nv})-\mu_{vn}^{i,t}+ \rho/2(\beta_{nv}^i-2\alpha_{vn}^{i,t+1})\Big) \tag{20}$$

subject to: C2-b (17), C3 (11), C5-b (14) and C6 (15). Upon obtaining optimal $\alpha_{vn}^{i,t+1}$ and $\beta_{nv}^{i,t+1}$, the dual variable can be updated as follows:

$$\mu_{vn}^{i,t+1}=\mu_{vn}^{i,t}+\rho(\alpha_{vn}^{i,t+1}-\beta_{nv}^{i,t+1}) \tag{21}$$

sbADMM is outlined in Algorithm 1. Since the problems are strictly convex, given the penalty term and obeying Lipschitz continuous gradient, sbADMM is guaranteed to converge to near optimal solution within a predefined error gap [32].

Algorithm 1 sbADMM

1: Initialize $t=0$, $s_{pri}=\infty$, $s_{dual}=\infty$, $\epsilon_{pri}>0$, $\epsilon_{dual}>0$
2: each $v\in\mathcal{V}$ initializes $\alpha_{vn}^{i,t}=0,\forall i\in\mathcal{I},n\in\mathcal{N}^i$ and publish the energy and bandwidth costs of their paths γ_{vn},θ_{vn} to the service points.
3: each $n\in\mathcal{N}^i,\forall i\in\mathcal{I}$ initializes $\beta_{nv}^{i,t}=0,\mu_{vn}^{i,t}=0,\ \forall v\in\mathcal{V}$ and publish their energy and processing costs, γ_n,θ_n to the access nodes.
4: **for** $i\in\mathcal{I}$ **do**
5: Each access node $v\in\mathcal{V}$ calculates the communication latency $\tau_{vn}^i+\tau_{nv}^i$ to each valid $n\in\mathcal{N}^i$ and publish it to the respective n.
6: Each service point of $n\in\mathcal{N}^i$ calculates the processing latency τ_n^i and publish it to the access nodes. If the service point reaches its capacity limits, the fog node notifies access nodes to remove it from the candidates set.
7: Both sides calculate the response time $\tau_{vn}^i+\tau_{nv}^i+\tau_n^i$
8: **while** ($||s_{pri}||_2>\epsilon_{pri}$ OR $||s_{dual}||_2>\epsilon_{dual}$) **do**
9: Each $v\in\mathcal{V}$ solves its access problem of (19) for α_v^i, given $\beta^{i,t}=\{\beta_{nv}^{i,t}\mid n\in\mathcal{N}^i\}$ and publish the optimal solution $\{\alpha_{vn}^{i,t+1}|n\in\mathcal{N}^i\}$ to fog nodes $\mathcal{N}^i$
10: Each $n\in\mathcal{N}^i$ solves its fog problem of (20) for β_n^i, given $\alpha_n^{i,t+1}=\{\alpha_{vn}^{i,t+1}\mid v\in\mathcal{V}\}$
11: Each $n\in\mathcal{N}^i$ updates dual variable μ_{vn}^i as per (21) and share the optimal solution $\beta_{nv}^{i,t+1}$ and the new dual value $\mu_{vn}^{i,t+1}$ with each $v\in\mathcal{V}$
12: **end while**
13: **end for**

1) Solving the Access-node and Fog-node Problems: The access and fog problems at each step $t+1$ are conic quadratic programs that can be solved analytically as follows:

Lemma 1: for any access node of $v\in\mathcal{V}$ and service point of i hosted by $n\in\mathcal{N}^i$, if $(\beta_{nv}^{i,t}-\frac{\mu_{vn}^{i,t}+\delta_v^i\gamma_n\theta_n}{\rho})\omega_v^{i,r}<\frac{|p_{nv}|r^i}{d^i-\tau_n^i-\tau_{vn}^i}$, the optimum $\alpha_{vn}^{i,t+1}$ is:

$$\alpha_{vn}^{i,t+1}=\max\Big\{\beta_{nv}^{i,t}-\Big(\frac{\delta_v^i\gamma_n\theta_n+\mu_{vn}^{i,t}}{\rho}\Big),0\Big\} \tag{22}$$

The proof is provided in Appendix B.

Lemma 2: for any service point of i at $n\in\mathcal{N}^i$, if $\sum_{v\in\mathcal{V}}(\frac{\mu_{vn}^{i,t}-\omega_v^{i,r}\gamma_{nv}\theta_{nv}}{\rho})\delta_v^i<\frac{c_n^i}{c^i}-\frac{1}{d^i-\tau_{nv}^i+\tau_{vn}^i}$, the

optimum $\beta_{nv}^{i,t+1}$ is:

$$\beta_{nv}^{i,t+1} = \max\left\{\alpha_{vn}^{i,t+1} + \frac{\mu_{vn}^{i,t} - \omega_v^{i,r}\gamma_{nv}\theta_{nv}}{\rho}, 0\right\} \quad (23)$$

The proof is provided in Appendix C.

Notably, a node's problem is a system of linear equations that develops into a non-invertible matrix, which cannot be solved in a straightforward manner. Schur decomposition [33] is applied to obtain the orthogonal and upper triangular matrices, used to back-solve the system. Furthermore, there are cases of stalemate when the two sides cannot converge on a solution. To avoid these and expedite the operation of the algorithm, we calculate the cumulative average of $||s_{pri}||_2$ at each iteration in the form:

$$\overline{||s_{pri}||_2^t} = \frac{\sum_{k=2}^{t}||s_{pri}||_2^k - ||s_{pri}||_2^{k-1}}{t} \quad (24)$$

and apply $\min(||s_{pri}||_2^t, \overline{||s_{pri}||_2^t}) \leq \epsilon_{pri}$ as the stopping criteria. This allows flexibility in converging faster when the optimum solution is found and no better solution is achieved within a controlled number of iterations. At the same time, it allows the algorithm sufficient number of iterations when the variation of s_{pri} is high. This technique along with varying the penalty parameter [27] lead to better performance than the simpler alternatives of [13].

B. Complexity Analysis

For a service i, an access node connected to v and a service point hosted at n respectively, the computation complexity is comprised of the calculations of: the latency terms $\mathcal{O}(\tau^i)$, the mapping decision at the access node $\mathcal{O}(\alpha_v^i)$ and the response routing decision at the service point $O(\beta_n^i)$. The complexity in calculating the latency terms is: $\mathcal{O}(\tau^i) = \mathcal{O}(N+1)$ as each service point calculates its own processing delay and each access node calculates the communication delay to each fog node. The mapping decision complexity, without considerations of parallel computations, is $\mathcal{O}(\alpha_v^i) = t(\mathcal{O}(N^3) + 3 \times \mathcal{O}(N^2))$. The first term corresponds to the complexity of Schur decomposition to obtain the triangular matrices, while the second term corresponds to three solving operations of the linear system. This is repeatable for t iterations, until convergence. Similarly, the complexity of the response routing decision is $\mathcal{O}(\beta_n^i) = t(\mathcal{O}(V^3) + 3 \times \mathcal{O}(V^2))$. Considering efficiencies of computation parallelism, for a small value of $t = 19$ and $N = 20, V = 26$, solving for α_v^i takes an average of $30 - 50$ msec, while solving for β_n^i takes an average of $5-7$ msec. For a large $t = 731$, solving for α_v^i takes on average $1.5-2$ sec while for β_n^i $25-50$ msec. The algorithm is run on a 64-bit server, having Intel Xeon Bronze 3106 CPU of 1.70GHz and L2 cache of 1024K. Orthogonally, the spacial complexity of the mapping and response routing decisions across all services are: $\mathcal{O}(\alpha_v) = I \times \mathcal{O}(\alpha_v^i)$ and $O(\beta_n) = I \times \mathcal{O}(\beta_n^i)$.

VI. Performance Evaluation

This section analytically evaluates the performance of the proposed sbADMM algorithm for service-based request mapping and response routing. Four key performance indicators are assessed: *delay satisfaction rate* for SLA compliance; *CPU and bandwidth utilization*; along with *CPU and bandwidth Energy costs*. The latter reflect the costs incurred by the two providers. Furthermore, *convergence* is analyzed as an indicator of the speed to converge on an optimum solution.

For the ecosystem: AT&T topology of 25 nodes and 114 links [34] is assumed as the operator's network, each node representing a forwarder. A set of fog nodes are colocated with a subset of network forwarders. Their distribution follows the centrality order of forwarders in the network. Hence, tier-1 clouds are likely to be colocated with highly connected tier-1 forwarders, while tier-3 edge nodes are likely to be placed with less connected tier-3 forwarders. The remainder forwarders do not have fog nodes colocated with them. Hence, their access nodes need to map demand to remote fog nodes.

1000 services are exposed in the network with a Zipf-based popularity distribution of exponent value 0.9. This generates realistic workload based on service popularity, as has been observed in existing cloud data centers [25], [26]. The penalty parameter, ρ, is initialized to 1 and subsequently tuned by $\zeta = 10$ as suggested by [27] for faster convergence. The evaluation parameters and their common settings are summarized in Table II. Notably, the CPU energy price varies within a range, depending on the tier of the fog node. However, the bandwidth energy price is fixed irrespective of the link's tier. This is a deliberate choice to reduce the evaluation complexity and focus on illustrating the impact of CPU energy cost on the algorithm's performance.

TABLE II
EVALUATION PARAMETERS AND THEIR SETTINGS

Parameter	**Setting**
number of services	1000
popularity distribution	zipf($s = 0.9$)
task size per service [MIPS]	$[80 - 100]$
Average upstream data per service [Mb]	0.4
Average downstream data per service [Mb]	4.0
Average response time tolerance [msec]	$[60 - 200]$
fog nodes per tier	$\{t_1 : 2, t_2 : 6, t_3 : 12\}$
Average CPU capacity per node [MIPS]	$\{t_1 : [10^7 - 10^8], t_2 : [10^6 - 10^7], t_3 : [10^5 - 10^6]\}$
Average CPU energy price per node [PpMIPS]	$\{t_1 : [10^{-3} - 10^{-2}], t2 : [10^{-2} - 10^{-1}], t3 : [10^{-1} - 10^{0}]\}$
Average bandwidth capacity per link [Mbps]	$\{t_1 : [10^6 - 10^7], t_2 : [10^5 - 10^6], t_3 : [10^4 - 10^5]\}$
Average bandwidth energy price per link [PpMbps]	10^{-3}
Average link length [Km]	$\{t_1 : [10 - 100], t_2 : [1 - 10], t_3 : [0.1 - 1]\}$
error gaps	$\epsilon_{pri} : 10^{-2}, \epsilon_{dual} : 10^{-4}$
ρ, ζ	1, 10

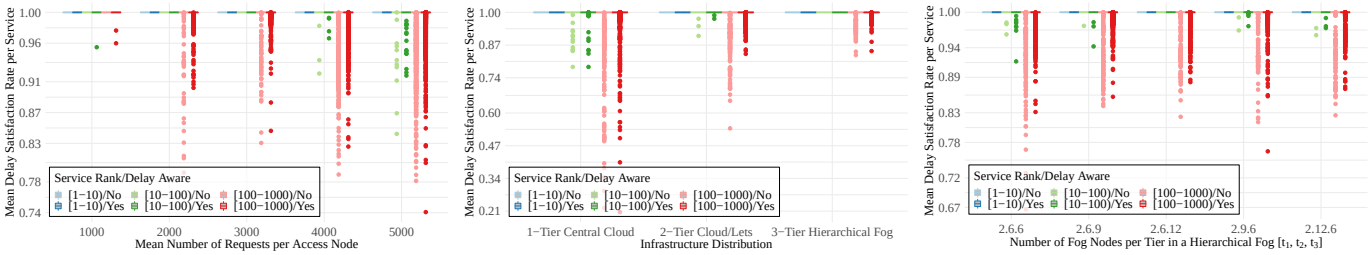
Three scaling scenarios have been examined: 1) **scaling demand:** by increasing the number of requests per access node; 2) **infrastructure distribution:** by extending the fog from 1-tier cloud to 2-tier cloud/cloudlets and 3-tier hierarchical fog of cloud/cloudlets/edge; and, 3) **tier scaling:** by increasing either tier-2 or tier-3 nodes. For each scenario, 10 simulations have been conducted in which the locations of fog nodes are selected following a centrality-guided probability. The algorithm is assessed for two variants: *delay-aware* and *delay-agnostic*. The former incorporates C2-a and C2-b, and only when they cannot be satisfied it falls back to the capacity constraints. The delay-agnostic counterpart ignores C2-a and C2-b and only incorporates the capacity constraints.

A. Delay Satisfaction Rate

This section presents the achieved mean delay satisfaction rate per service. A value below 1.0 indicates a violation of the response time tolerance of the service. Figure 4a shows for over 90% of demand (i.e. for the 10% most popular services), the satisfaction rate remains at $\approx$ 100% for increasing number of requests. The rate only drops at the highest 5000 request/access-node where outlier violations are observed. The lowest is at 83% for the delay-agnostic variant compared to 90% for the delay-aware counterpart. The 10% demand for the lower ranks of services $[100, 1000]$ has $75 - 95\%$ satisfaction rate. The number of violations is less significant for the delay-aware variant, having the most frequent outliers in the range of $\approx 87\% - 95\%$. In comparison, the delay-agnostic counterpart has frequent outliers in the range of $\approx 82\% - 91\%$.

The violations by the delay-aware variant are instigated by the error gap between the two optimization sides. Besides, there are instances that lack a solution satisfactory of C2-a and C2-b, particularly for latency-critical unpopular services. In the delay-agnostic variant, limiting constraints to capacities and energy cost translates into a larger misalignment between the delay requirement of a service and the solution. This leads to delay-non-discriminatory mapping of popular services to the edge, straining its resources and leaving insufficient capacity for latency-critical services. Consequently, demand for the latter is mapped to the middle or central tier, leading to violation of the tolerance threshold.

Figure 4b shows the delay satisfaction rate when the fog infrastructure expands from 1-tier cloud to 3-tier hierarchical fog. The number of requests in this scenario is fixed to an average of 3000 requests/access-node. The results show the satisfaction rate improves significantly, from $\approx$ 40% to $\approx$ 90% when moving towards a hierarchical distribution. Complementary, while in 1-tier setting the difference of satisfaction rates between the two variants is marginal, it is considerably larger in the 2-tier and 3-tier settings. The delay-aware variant is showing superiority over the agnostic counterpart by $\approx 10 - 25\%$. Figure 4c shows the satisfaction rate when scaling the number of fog nodes at tier-2 or tier-3 of a hierarchical fog. The average number of requests/access-node is 3000. The results show superior improvement of the satisfaction rate when



(a) Scaling Demand (b) Infrastructure Distribution (c) Tier Scaling

Fig. 4. Mean delay satisfaction rate per service for three scaling scenarios: load, distribution and capacity per tier.

increasing the number of edge nodes, compared to increasing cloudlet nodes.

Notably, the Zipf distribution of service popularity allows for efficient prioritization of demand, by service rank. This has shown to maximize the ratio of satisfied demand. Albeit, it comes at the cost of disadvantaging demand of unpopular services. In contrast, distributions such as the uniform have a flattening effect on the volume of demand across services. Consequently, alternative prioritization criteria to service rank would be needed. However, preliminary indicators suggest that this increases the risk of reducing the overall demand satisfaction.

B. CPU and Bandwidth Utilization

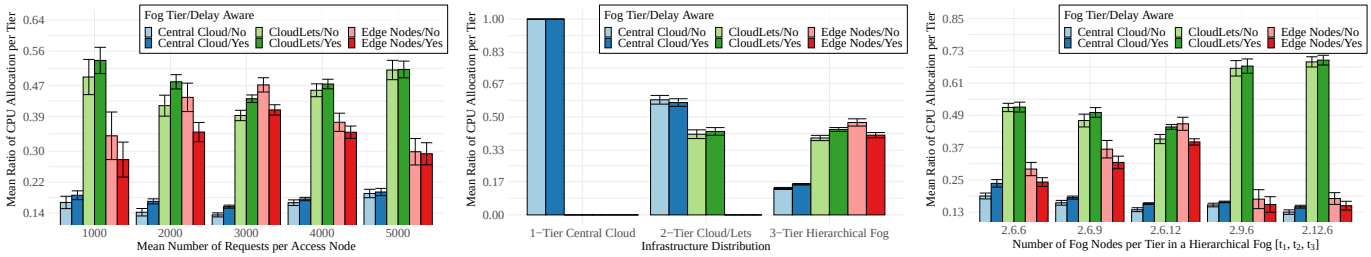
This section presents the results of CPU and bandwidth utilization. Figure 5a shows the results when scaling up the demand. Generally, allocation is higher in tier-2 cloudlets with a maximum average of $\approx 0.5-0.55$, when demand is lowest at 1000 and highest at 5000 requests/access-node. However, the ratio drops to a lowest value of $\approx 0.4-0.45$ in the middle at 3000 requests/access-node. The second highest ratio of allocation is on tier-3, the edge, exhibiting opposite pattern to that of tier-2. Tier-1 has the lowest allocation rate, following similar pattern to that of tier-2.

Orthogonally, the delay-aware variant shows a higher demand allocation to tiers 1 and 2 by $\approx 7-10\%$ compared to their delay-agnostic counterpart. This is counter intuitive, revealing a non-trivial interplay of costs. In the delay-agnostic case, as the path cost is dominantly a function of bandwidth capacity and distance, allocation to the edge is overall cheaper by saving communication cost. However, in the delay-aware case, the algorithm attempts to push allocation as close to the flourished cloud as the latency tolerance allows. Consequently, alleviating pressure on the precious capacity of the edge, and sparing it for latency-critical services. This difference in allocation has a considerable impact on the satisfaction rate as has been shown in Figure 4a, and the energy cost analyzed in Section VI-C.

Figure 5b shows the CPU allocation when the infrastructure extends from a 1-tier cloud to a 3-tier fog, for a fixed average of 3000 requests/access-node. The results show a significant shift of allocations, from tier-1 cloud to tiers 2 and 3. However, introducing the edge does not change the fraction of allocation to cloudlets at tier-2. This is because tier-2 continues to offer the cheapest combination of compute and communication cost. The difference between the delay-aware and delay-agnostic variants is marginally visible, with the delay-aware variant allocating highest fraction to tier-2.

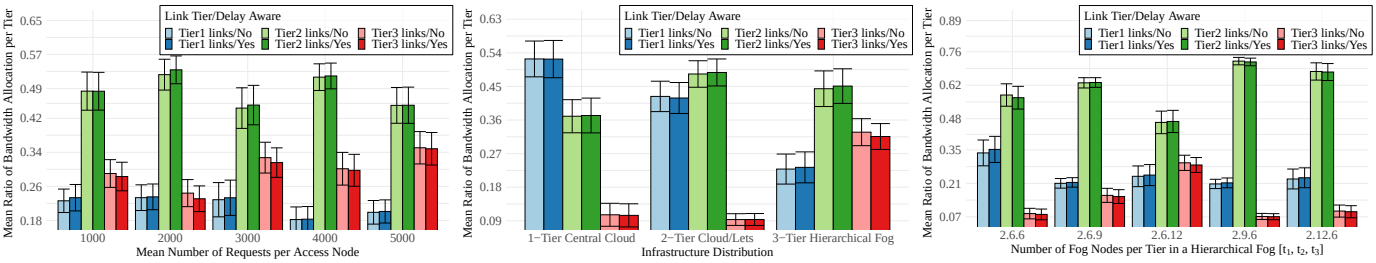
Figure 5c illustrates the CPU allocation when scaling tier-2 or tier-3. Expanding tier-2 results in a significant shift of $\approx 30-35\%$ of allocations, from tier-3 towards tier-2. In comparison, allocation is shifted by $\approx 15-25\%$ when expanding tier-3 instead. This comes back to the cheaper rate of tier-2 combined with the suitable proximity to end-users, achieving the desired response time at a cheaper cost. This result together with the earlier ones of Figure 4c present a critical trade-off to service providers, between the cost and benefit of extending the edge as opposite to the middle cloudlet.

Figure 6 shows the bandwidth utilization for the three scenarios. Given that the energy price per link tier is fixed, link usage does not skew towards one tier or the other. Although, the delay-aware variant shows marginal favor of tier-1 and tier-2 links compared to tier-3. This is caused by two factors: 1) the cheaper cost of bandwidth on tier-1 and tier-2 links, outweighing the longer distance of these links; and 2) the higher CPU allocation to tier-2 nodes, often reachable via paths that incorporate more tier-1 and



(a) Scaling Demand (b) Infrastructure Distribution (c) Tier Scaling

Fig. 5. CPU utilization per fog tier as a fraction of total workload allocation for all services.



(a) Scaling Demand (b) Infrastructure Distribution (c) Tier Scaling

Fig. 6. Bandwidth utilization per link tier as a fraction of total response data of demand allocation for all services.

tier-2 links than tier-3 counterparts.

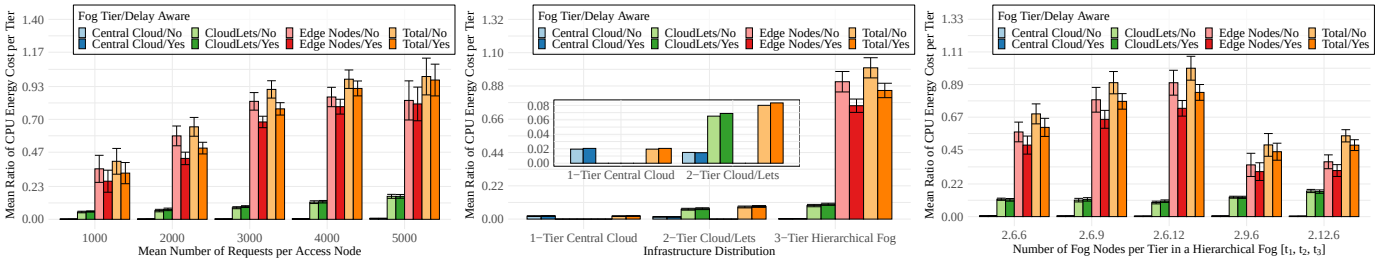
C. CPU and Bandwidth Energy Cost

This section presents the CPU and bandwidth energy cost, incurred by the allocations shown earlier in Section VI-B. The results are presented as a relative value to the most expensive allocation instance. Figure 7a shows the minimum computing energy cost, which increase approximately linearly and flattens out for the high end of load (i.e. $4000-5000$ requests). The cost growth is correlated with demand rather than energy price, which shows the algorithm's ability to bound the growth in cost by usage. Orthogonally, the cost incurred by the delay-aware variant is $\approx 13-15\%$ less than that of the delay-agnostic counterpart, at the lower end of load (i.e. $1000-3000$ requests). This is due to the higher allocation to tier-2 by the delay-aware variant, not only conserving tier-3 resources but facilitating lower energy cost. Notably, the delay-agnostic variant does not always result in cheaper CPU energy cost. Because, although the overall cost is minimized by higher usage of the edge, higher energy prices there drive the CPU energy cost higher than that of the delay-aware variant.

Figure 7b illustrates the CPU energy cost incurred by different fog distributions. The figure shows the ratio of computing energy cost to have strong correlation with the price range of each tier. Complementary, Figure 7c shows the computing energy cost when scaling tier-2 or tier-3. As expected, the computing cost incurred by scaling tier-2 is considerably lower than that of scaling tier-3, $\approx 40-50\%$. These results, together with those of Figures 4b and 4c, present a cost-benefit analysis to service providers. The trade-off is between improving the satisfaction rate by scaling the edge, and the cost of such expansion.

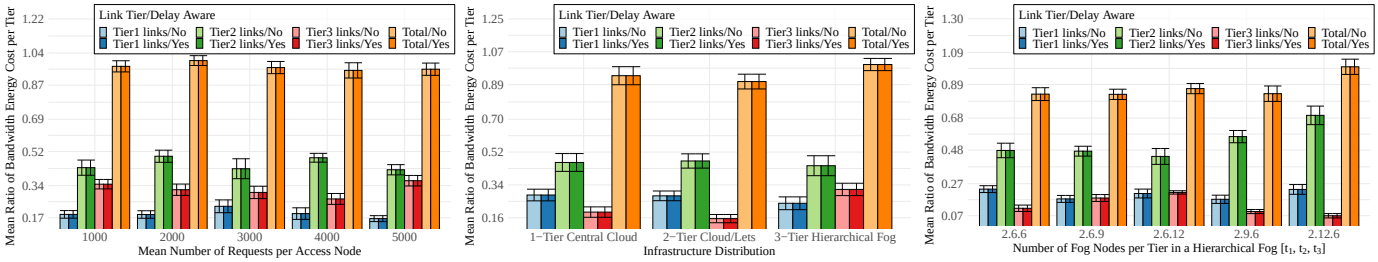
The results of Figures 8a-8c show the relative communication energy cost per link tier. Tier-2 links incur the highest fraction of energy cost, $\approx 44-50\%$, followed by tier-3 incurring $\approx 30-35\%$ of cost. The lowest is tier-1 with a share of $\approx 18-20\%$. The higher fraction of cost of tier-2 is driven by the higher allocation to cloudlets, which increases the likelihood of utilizing tier-2 links. Hence, the cost correlates with the utilization rate.

Complementary to the above, it is worth noting that higher allocation to the edge changes the characteristics of aggregate demand expected by network operators and cloud providers. For instance, exposing these deeper parts of the ecosystem to higher demand variation hinders forecast and prediction exercises. Consequently, introduces a higher likelihood of sub-optimal utilization of existing infrastructure and challenges in longer terms planning.



(a) Scaling Demand (b) Infrastructure Distribution (c) Tier Scaling

Fig. 7. Computation Energy Cost per fog tier as a relative value to the most expensive allocation instance.



(a) Scaling Demand (b) Infrastructure Distribution (c) Tier Scaling

Fig. 8. Communication Energy Cost per link tier as a relative value to the most expensive allocation instance.

For these reasons, edge planning and utilization need to take into account service requirements along with infrastructure constraints, to maximize the social welfare of all entities in the ecosystem.

D. Convergence

Figure 9 show the distribution of number of iterations needed to reach convergence, in each of the scaling scenarios. The lower and upper whiskers indicate the 5% and 95% percentiles, respectively. The results show overall the number of iterations is lowest for scenario (b), scaling infrastructure distribution. Because, when the infrastructure is limited to 1-tier cloud or 2-tier cloudlets, the set of candidate solutions is small with relaxed capacity constraints. The number is higher for the delay-aware variant, because the delay constraint enforces tighter restriction on the solutions space. The impact of it intensifies when the infrastructure includes the constrained tier-3 edge. This results in higher variation of constraints and costs across the two sides of the problem, causing slower convergence.

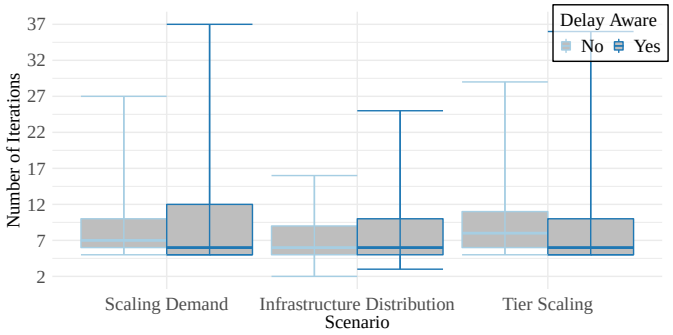


Fig. 9. Number of iterations of sbADMM per scaling scenario, for all services, tests and control variables.

VII. Conclusion

This work proposed a service-based, decentralized, fog management and network orchestrator (sbMANO). The sbMANO offers novel discovery and mapping services, allowing for autonomous optimization at the granularity of services. The fog ecosystem has been modeled analytically and the problem of joint request mapping and response routing has been formulated. To solve the latter, the work further proposed a novel approximation algorithm based on ADMM, having provable near-optimal performance with bounded violations. The algorithm minimizes the joint computing and communication costs. This takes into account diverse: energy prices, CPU-bandwidth capacities and physical distance. The algorithm's performance has been evaluated analytically for two variants: delay-aware and delay-agnostic. Evaluation results have shown that unnecessary allocation to the edge can strain its resources, hindering its ability to serve demand for

latency-critical services. Instead, allocation to the middle tier of cloudlets results in superior performance. This is reflected by the higher satisfaction rate of $\approx 100\%$ for over 90% of total demand, at a lower energy cost. Future work will tackle problems of service management and workload allocation for cloud-native applications, given data storage constraints and variation in processor architectures.

REFERENCES

[1] R. Mahmud and R. Buyya, "Fog computing: A taxonomy, survey and future directions." *CoRR*, 2016.

[2] C. Mouradian, D. Naboulsi, S. Yangui, R. H. Glitho, M. J. Morrow, and P. A. Polakos, "A comprehensive survey on fog computing: State-of-the-art and research challenges," *IEEE Communications Surveys Tutorials*, vol. 20, no. 1, pp. 416–464, Firstquarter 2018.

[3] R. K. Naha, S. Garg, D. Georgakopoulos, P. P. Jayaraman, L. Gao, Y. Xiang, and R. Ranjan, "Fog computing: Survey of trends, architectures, requirements, and research directions," *IEEE Access*, vol. 6, pp. 47 980–48 009, 2018.

[4] H. F. Atlam, R. J. Walters, and G. B. Wills, "Fog computing and the internet of things: A review," *Big Data and Cognitive Computing*, vol. 2, no. 2, 2018. [Online]. Available: https://www.mdpi.com/2504-2289/2/2/10

[5] O. C. A. W. Group, "Openfog reference architecture for fog computing," Fremont, CA, USA, Tech. Rep. OPFRA001.020817, Feb 2017.

[6] Y. Sverdlik, "What is the Data Center Cost of 1kW of IT Capacity?" https://www.datacenterknowledge.com/archives/2016/08/23/what-is-the-data-center-cost-of-1kw-of-it-capacity, Aug 2016, [Online; accessed 12-March-2022].

[7] M. Koot and F. Wijnhoven, "Usage impact on data center electricity needs: A system dynamic forecasting model," *Applied Energy*, vol. 291, p. 116798, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306261921003019

[8] P. S. Moura, T. Vasques, and A. T. Almeida, "Energy efficiency insight into small and medium data centres: A comparative analysis based on a survey," in *13th European Council for an Energy Efficient Economy Summer Study on Energy Efficiency (ECEEE 2017)*, Jun 2017, pp. 1541–1550.

[9] A. M, B. P, and C. L, "Trends in data centre energy consumption under the european code of conduct for data centre energy efficiency," *ENERGIES*, vol. 10, no. 10, p. 1470, 2017.

[10] L. Altamira, J. Viegand, D. Polverini, B. Huang, and S. Flucker, "The role of data centres in reducing energy consumption through policy measures," vol. 2019-June, 2019, pp. 1581–1591. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85085203856&partnerID=40&md5=3718881d8893fa4c958f54fc025ea57e

[11] E. Innovation, "How Much Energy Do Data Centers Really Use?" https://energyinnovation.org/2020/03/17/how-much-energy-do-data-centers-really-use/, March 2020, [Online; accessed 12-March-2022].

[12] M. Dayarathna, Y. Wen, and R. Fan, "Data center energy consumption modeling: A survey," *IEEE Communications Surveys Tutorials*, vol. 18, no. 1, pp. 732–794, 2016.

[13] H. Xu and B. Li, "Joint request mapping and response routing for geo-distributed cloud services," in *2013 Proceedings IEEE INFOCOM*, Apr 2013, pp. 854–862.

[14] Q. Fan, H. Yin, L. Jiao, Y. Lyu, H. Huang, and X. Zhang, "Towards optimal request mapping and response routing for content delivery networks," *IEEE Transactions on Services Computing*, vol. 14, pp. 606–613, 2021.

[15] M. S. Aslanpour, S. S. Gill, and A. N. Toosi, "Performance evaluation metrics for cloud, fog and edge computing: A review, taxonomy, benchmarks and standards for future research," *Internet of Things*, vol. 12, p. 100273, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2542660520301062

[16] B. Ali, M. A. Gregory, and S. Li, "Multi-access edge computing architecture, data security and privacy: A review," *IEEE Access*, vol. 9, pp. 18 706–18 721, 2021.

[17] I. Martinez, A. S. Hafid, and A. Jarray, "Design, resource management, and evaluation of fog computing systems: A survey," *IEEE Internet of Things Journal*, vol. 8, no. 4, pp. 2494–2516, 2021.

[18] C. Feng, H. Xu, and B. Li, "An alternating direction method approach to cloud traffic management," *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 8, pp. 2145–2158, 2017.

[19] T. Goethals, F. De Turck, and B. Volckaert, "Near real-time optimization of fog service placement for responsive edge computing," *Journal of Cloud Computing*, vol. 9, no. 1, p. 34, 2020. [Online]. Available: https://doi.org/10.1186/s13677-020-00180-z

[20] K. Poularakis, J. Llorca, A. M. Tulino, I. Taylor, and L. Tassiulas, "Service placement and request routing in mec networks with storage, computation, and communication constraints," *IEEE/ACM Transactions on Networking*, vol. 28, no. 3, pp. 1047–1060, 2020.

[21] D. Zhang, F. Haider, M. St-Hilaire, and C. Makaya, "Model and algorithms for the planning of fog computing networks," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 3873–3884, 2019.

[22] M. Mukherjee, S. Kumar, C. X. Mavromoustakis, G. Mastorakis, R. Matam, V. Kumar, and Q. Zhang, "Latency-driven parallel task data offloading in fog computing networks for industrial applications," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 9, pp. 6050–6058, 2020.

[23] M. F. AL-Naday, N. Thomos, and M. J. Reed, "Information-Centric Multilayer Networking: Improving Performance Through an ICN/WDM Architecture," *IEEE/ACM Transactions on Networking*, vol. 25, no. 1, pp. 83–97, Feb 2017.

[24] M. Al-Naday and I. Macaluso, "Flexible semantic-based data networking for iot domains," in *2021 IEEE 22nd International Conference on High Performance Switching and Routing (HPSR)*, 2021, pp. 1–6.

[25] B. Liu, Y. Lin, and Y. Chen, "Quantitative workload analysis and prediction using google cluster traces," in *2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2016, pp. 935–940.

[26] S. Di, D. Kondo, and F. Cappello, "Characterizing cloud applications on a google data center," in *2013 42nd International Conference on Parallel Processing*, 2013, pp. 468–473.

[27] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, p. 1122, jan 2011. [Online]. Available: https://doi.org/10.1561/2200000016

[28] M. J. Reed, M. Al-Naday, N. Thomos, D. Trossen, G. Petropoulos, and S. Spirou, "Stateless multicast switching in software defined networks," in *2016 IEEE International Conference on Communications (ICC)*, 2016, pp. 1–7.

[29] A. Giorgetti, A. Sgambelluri, F. Paolucci, N. Sambo, P. Cas-

toldi, and F. Cugini, "Bit index explicit replication (bier) multicasting in transport networks," in *2017 International Conference on Optical Network Design and Modeling (ONDM)*, 2017, pp. 1–5.
[30] D. Costenaro and A. Duer, "The megawatts behind your megabytes: Going from data-center to desktop," 2012.
[31] J. L. Gustafson, *Little's Law*. Boston, MA: Springer US, 2011, pp. 1038–1041.
[32] X. Cao and K. Liu, "Distributed linearized admm for network cost minimization," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 4, no. 3, pp. 626–638, 2018.
[33] P. Bartlett, "Lecture 5: The schur decomposition," 2014.
[34] S. Knight, H. X. Nguyen, N. Falkner, R. Bowden, and M. Roughan, "The Internet Topology Zoo," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 9, pp. 1765–1775, Oct 2011.
[35] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Athena Scientific, 1997.

Mays AL-Naday (Member, IEEE) received her PhD degree from the University of Essex, United Kingdom, in 2015. She is currently an Assistant Professor in the School of Computer Science and Electronic Engineering, University of Essex. She has actively worked on a number of EU H2020 research projects in the area of future networking architectures. Her current research focuses on microservice networking, smart resource management, fog computing networks, networks for federated learning and security and Quality of Service in B5G/6G. She has been the organizer of prestigious workshops in Sigcomm 17-18 and IFIP 17.

Nikolaos Thomos (S'02,M'06,SM'16) received the Diploma and Ph.D. degrees from the Aristotle University of Thessaloniki, Greece, in 2000 and 2005, respectively. He was a Senior Researcher with the Ecole Polytechnique Federale de Lausanne (EPFL) and the University of Bern, Switzerland. He is currently a Professor with the School of Computer Science and Electronic Engineering at the University of Essex, U.K. His research interests include machine learning for communications, multimedia communications, network coding, information-centric networking, networking, joint source and channel coding, video coding for machines, signal processing, and sensor networks. He is an elected member of the IEEE MMSP Technical Committee (MMSP-TC) for the period 2019-2024. He is the recipient of the highly esteemed Ambizione Career Award from the Swiss National Science Foundation (SNSF).

Jiejun Hu received her Ph.D. and MSc in the School of Computer Science and Technology from Jilin University, China, in 2019 and 2015, respectively. She is currently an Assistant Professor at the Lancaster University Leipzig. She was a senior research officer in the University of Essex, UK. Then, she served as a postdoctoral fellow in the Max-Planck Institute for Human Development, Germany. Her research focuses on incentive mechanisms design and game theory in various scenarios, such as the Internet of Things, Mobile CrowdSensing, software-defined networks, blockchain, and digital contact tracing systems.

Bruno Volckaert (Senior Member, IEEE) received the Ph.D. degree in resource management for grid computing from Ghent University, in 2006. He is currently a professor in advanced distributed systems at Ghent University and senior researcher at imec. He has worked on over 45 national and international research projects and is author or co-author of more than 150 peer-reviewed papers published in international journals and conference proceedings. His current research deals with reliable and high performance distributed software systems for a.o. Smart Cities, scalable cybersecurity detection and mitigation architectures and autonomous optimization of cloud-based applications.

Prof. Filip De Turck (Fellow, IEEE) leads the network and service management research group at Ghent University, Belgium and imec. He has coauthored over 700 peer reviewed papers. His research interests include design of secure and efficient softwarized network and cloud systems. He was elevated as an IEEE Fellow for outstanding technical contributions. He is involved in several research projects with industry and academia, served as chair of the IEEE Technical Committee on Network Operations and Management (CNOM), and steering committee member of the IFIP/IEEE IM, IEEE/IFIP NOMS, IEEE/IFIP CNSM and IEEE NetSoft conferences. He serves as Editor-in-Chief of IEEE Transactions on Network and Service Management (TNSM).

Martin Reed (M'99) received the PhD degree from the University of Essex, United Kingdom, in 1998. He is currently a Professor with the University of Essex. His research interests include network control planes, information centric networking, network security, and multimedia networking. He has been involved in a number of EPSRC, EU, and industrial projects in these areas which have made contributions to IETF and 3GPP 5G standards. He has held a Research Fellowship at BT in the area of access networks. He has led a number of international research testbeds that demonstrate and evaluate novel networking protocols and architectures.

APPENDIX A

Preliminary on ADMM

Alternating Direction Method of Multipliers (ADMM) is a long existing algorithm. It has been attracting increasing attention for its ability to provide scalable approximation solution for large convex optimization problems. The algorithm solves problems of type:

$$\min_{x,z} f(x) + g(z) \tag{25}$$

subject to:

$$Ax + Bz = y$$
$$x \in J_x,\ z \in J_z$$

where: $A \in \mathbb{R}^{p\times n}$, $x \in \mathbb{R}^n$, $B \in \mathbb{R}^{p\times m}$, $z \in \mathbb{R}^m$ and $y \in \mathbb{R}^p$; J_x, J_z are non-empty polyhedral sets. The objective function is separable over two sets of variables tied by an equality constraint. The augmented Lagrangian of the problem can be formed by introducing a penalty-controlled $\mathcal{L}-2$ norm term, in the form:

$$\begin{aligned}\mathcal{L}_p = f(x) + g(z) + \mu^T(Ax + Bz - y) \\ + (\rho/2)||Ax + Bz - y||_2^2\end{aligned} \tag{26}$$

where $\rho > 0$ is the penalty parameter and $\rho = 0$ gives the standard Lagrangian of the problem. By introducing the augmented Lagrangian, the problem is transformed into a strictly convex form even if f or g are linear. This allows the freedom in working on the dual problem without strong assumptions of the convexity of f and g. ADMM solves the dual problem with iterative alternations between the x and z variables. At the $(t+1)$th iteration, the solution takes the form:

$$x^{t+1} := \arg\min_x \mathcal{L}_p(x, z^t, \mu^t) \tag{27}$$
$$z^{t+1} := \arg\min_z \mathcal{L}_p(x^{t+1}, z, \mu^t) \tag{28}$$
$$\mu^{t+1} := \mu^t + \rho(Ax^{t+1} + Bz^{t+1} - c) \tag{29}$$

x is minimized first using the last values of z^t and the dual variable μ^t. x^{t+1} is then used to calculate minimum z and the new values x^{t+1} and z^{t+1} are then used to update μ, the dual variable. The optimality of ADMM can be guaranteed with some basic assumptions from [35]: if the optimal solution set of problem (25) is non empty and either the feasible set of x is bounded or the matrix $AA^\intercal$ is invertible, then any set $\{x^t, z^t, \mu^t\}$ is bounded and the solution $\{x^t, z^t\}$ is an optimal one.

APPENDIX B

PROOF OF LEMMA 1

The KKT conditions of the access-node set of problems of (19) is a system of linear equations, comprised of:

The primal feasibility conditions:

$$\sigma^i_{v,C1}(\sum_{n\in\mathcal{N}^i} \alpha^{i,t+1}_{vn} - 1) = 0 \tag{30}$$

$$\sigma^i_{vn,C2}\Big(\frac{H_p r^i}{w_p - \alpha^{i,t+1}_{vn}\omega^{i,r}_v} + \tau^i_{vn} + \tau^i_n - d^i\Big) = 0 \tag{31}$$

$$\sigma^i_{vn,C4}(\alpha^{i,t+1}_{vn}\omega^{i,r}_v - w_p) = 0 \tag{32}$$

$$\sigma^i_{vn,C5}\alpha^{i,t+1}_{vn} = 0 \tag{33}$$

Where $H_p = \sum_{e\in p} w_e/w_p$, w_p is the bandwidth on the bottleneck link of $p = p_{nv}$. When $w_e = w_p \forall e \in p$, $H_p = |p|$, i.e. $|p_{nv}|$. The condition of (31) is a simplification of the latency constraint without loss of generality. This is because we solve for $\alpha^{i,t+1}_{vn}$ by the latency on the path direction with the most intensive data. The latency on the opposite (low-data) direction is calculated as a fraction of the total latency budget for the service. If there exists a service point on a path that satisfies C2-a, the transmission latency on the opposite low-data path is guaranteed to be lower than the latency budget (i.e. $d^i - \min(\tau^i_{vn}, \tau^i_{nv}) > 0$ always holds). Following the latter, we assume $\min(\tau^i_{vn}, \tau^i_{nv})$ is calculated from the $\beta^{i,t}_{nv}$ provided to access nodes.

The dual feasibility conditions:

$$\sigma^i_{v,C_1} \geq 0,\ \sigma^i_{vn,C_j} \geq 0,\ j \in \{2,4,5\} \tag{34}$$

The stationarity conditions:

$$\delta^i_v\gamma_n\theta_n + \mu^{i,t}_{vn} + \rho(\alpha^i_{vn} - \beta^{i,t}_{nv}) + \sigma^i = 0 \tag{35}$$

where $\sigma^i = \sigma^i_{v,C1} + (\sigma^i_{vn,C2} + \sigma^i_{vn,C4})\omega^{i,r}_v + \sigma^i_{vn,C5}$ is the sum of Lagrange multipliers for the constraints: C1-C2-a and C4-C5-a at the access side. At any iteration, either $\sigma^i_{vn,C2}$ or $\sigma^i_{vn,C4}$ is active but not both. This depends on whether or not C2 can be satisfied. The conditions of (31)-(33) correspond to the inequality constraints of C2, C4 and C5-a. $\sigma^i \neq 0$ when one or multiple of these constraints are their tight state (i.e. approaching or violating their upper limit). When $\beta^{i,t}_{nv} - \mu^{i,t}_{vn} - \delta^i_v\gamma_n\theta_n \leq 0$, $\alpha^{i,t+1}_{vn}$ must equal zero, otherwise the left hand side of (35) is always positive and C1 is violated. Hence, first $\beta^{i,t}_{nv} - \mu^{i,t}_{vn} - \delta^i_v\gamma_n\theta_n > 0$ must hold in order for $\alpha^{i,t+1}_{vn} > 0$.

To satisfy the harder C2-a: initially consider a service point is a valid candidate if none of the terms in the feasibility condition of (31) is $\inf$, i.e. $\tau_n^i \neq \inf$ is controlled by the fog side. Hence, $(\beta_{nv}^{i,t} - \frac{\mu_{vn}^{i,t}+\delta_v^i\gamma_n\theta_n}{\rho})\omega_v^{i,r} < w_p$ must hold. Furthermore, for C2-a not to be violated, a service point should hold the inequality $(\beta_{nv}^{i,t} - \frac{\mu_{vn}^{i,t}+\delta_v^i\gamma_n\theta_n}{\rho})\omega_v^{i,r} < \frac{|p_{nv}|r^i}{d^i-\tau_n^i-\tau_{vn}^i}$. The latter further satisfies C4 and as such C5-a too. thus completing the proof.

APPENDIX C
PROOF OF LEMMA 2

The fog problems of (20) constitute a system of linear equations comprised of:

The primal feasibility conditions:

$$\chi_{nv,C2}^i\Big(\frac{ci}{c_n^i - ci\sum_{v\in\mathcal{V}}\beta_{nv}^i\delta_v^i} + \tau_{vn}^i + \tau_{nv}^i - d^i\Big) = 0 \quad (36)$$

$$\chi_{n,C3}^i\Big(\sum_{v\in\mathcal{V}}\beta_{nv}^i\delta_v^i - c_n^i\Big) = 0 \quad (37)$$

$$\chi_{nv,C5}^i\beta_{nv}^{i,t+1} = 0 \quad (38)$$

The dual feasibility conditions

$$\chi_{nv,C_j}^i \geq 0,\ j \in \{2,3,5\} \quad (39)$$

and, the *stationarity conditions:*

$$\omega_v^{i,r}\gamma_{nv}\theta_{nv} - \mu_{vn}^{i,t} + \rho(\beta_{nv}^{i,t} - \alpha_{vn}^{i,t+1}) + \chi^i = 0 \quad (40)$$

where $\chi^i = (\chi_{nv,C2}^i + \chi_{n,C3}^i)\delta_v^i + \chi_{nv,C5}^i$ are the multipliers for the constraints C2-b, C3 and C5-b at the fog side. Similar to the access problem, at any iteration, either $\chi_{nv,C2}^i$ or $\chi_{n,C3}^i$ is active but not both. This depends on whether or not C2 can be satisfied. When $\alpha_{vn}^{i,t+1} + \frac{\mu_{vn}^{i,t}-\omega_v^{i,r}\gamma_{nv}\theta_{nv}}{\rho} \leq 0$, $\beta_{nv}^{i,t+1}$ must equal zero otherwise the left hand side of (40) is always positive, violating C3 as proven by [13]. However, when considering the response time, the set of candidate service points must satisfy $c^i\sum_{v\in\mathcal{V}}(\frac{\mu_{vn}^{i,t}-\omega_v^{i,r}\gamma_{nv}\theta_{nv}}{\rho})\delta_v^i < c_n^i$ in order for $\tau_n^i < \inf$. Now, to satisfy C2-b the set of candidates is further reduced to satisfy $\sum_{v\in\mathcal{V}}(\frac{\mu_{vn}^{i,t}-\omega_v^{i,r}\gamma_{nv}\theta_{nv}}{\rho})\delta_v^i < \frac{c_n^i}{c^i} - \frac{1}{d^i-\tau_{nv}^i+\tau_{vn}^i}$. Such a set satisfies C3 by virtue of satisfying the harder C2-b constraint and as such C5-b is similarly satisfied. This results in $\chi^i = 0$ and $\beta_{nv}^{i,t+1} = \max\Big\{\alpha_{vn}^{i,t+1} + \frac{\mu_{vn}^{i,t}-\omega_v^{i,r}\gamma_{nv}\theta_{nv}}{\rho}, 0\Big\}$. Hence the proof.