

# Distributed Optimization for Energy-efficient Fog Computing in the Tactile Internet

Yong Xiao, *Senior Member, IEEE* and Marwan Krunz, *Fellow, IEEE*

**Abstract**—Tactile Internet is an emerging concept that focuses on supporting high-fidelity, ultra-responsive, and widely available human-to-machine interactions. To reduce the transmission latency and alleviate Internet congestion, fog computing has been advocated as an important component of the Tactile Internet. In this paper, we focus on energy-efficient design of fog computing networks that support low-latency Tactile Internet applications. We investigate two performance metrics: *Service response time of end-users* and *power usage efficiency of fog nodes*. We quantify the fundamental tradeoff between these two metrics and then extend our analysis to fog computing networks involving cooperation between fog nodes. We introduce a novel cooperative fog computing concept, referred to as *offload forwarding*, in which a set of fog nodes with different computing and energy resources can cooperate with each other. The objective of this cooperation is to balance the workload processed by different fog nodes, further reduce the service response time, and improve the efficiency of power usage. We develop a distributed optimization framework based on *dual decomposition* to achieve the optimal tradeoff. Our framework does not require fog nodes to disclose their private information nor conduct back-and-forth negotiations with each other. Two distributed optimization algorithms are proposed. One is based on the subgradient method with dual decomposition and the other is based on distributed alternating direction method of multipliers via variable splitting (ADMM-VS). We prove that both algorithms can achieve the optimal workload allocation that minimizes the response time under the given power efficiency constraints of fog nodes. Finally, to evaluate the performance of our proposed concept, we simulate a possible implementation of a city-wide self-driving bus system supported by fog computing in the city of Dublin. The fog computing network topology is set based on a real cellular network infrastructure involving 200 base stations deployed by a major cellular operator in Ireland. Numerical results show that our proposed framework can balance the power usage efficiency among fog nodes and reduce the service latency for users by around 50% in urban scenarios.

**Index Terms**—Tactile Internet, fog computing, power efficiency, dual decomposition, ADMM, self-driving vehicle.

## I. INTRODUCTION

With the widespread deployment of high-performance computing infrastructure and advancement of networking and com-

munication technology, it is believed that the vision of the Tactile Internet (TI) will soon become a reality, transforming the existing content-delivery-based Internet into skill-set delivery-based networks [2]. According to the Next-Generation Mobile Network (NGMN) Alliance [3], TI is defined as the capability of remotely delivering real-time control over both real and virtual objects as well as physical haptic experiences through the Internet. It will be able to contribute to the solution of many complex challenges faced by our society, enabling novel services and applications that cannot fit well in the current state-of-the-art networking and cloud computing architectures. Examples of these applications include long-distance education with immersive learning experience, high-precision remote medical diagnosis and treatment, high-sensitive industry control and automation, collision-avoidance for high-speed autonomous vehicles, high-fidelity virtual/augment reality (VR/AR), etc. Recent analysis shows that the TI has the potential to generate up to \$20 trillions global market, accounting for around 20% of the global GDP [4]. Because the TI will provide critical services, it needs to be extremely reliable, ultra-responsive, and widely available. More specifically, according to International Telecommunication Union (ITU), the TI must support latencies as low as 1 ms, the reliability of around one second of outage per year, enhanced security, as well as sufficient computational resources within the range of a few kilometers from each end-user [5].

Fog computing has been recently introduced as a promising solution to accommodate the stringent requirements of the TI. It complements the cloud infrastructure by involving a large number of low-cost, often decentralized devices, commonly referred to as fog nodes, to perform computing, storage, control, and network functions closer to end-users [6]–[8]. Fog nodes embody a variety of devices between end-users and cloud data centers (CDCs), including routers, smart gateways, access points (APs), base stations (BSs), as well as portable devices such as drones, robots, and vehicles with computing and storage capabilities [9]. The success of the TI will hinge on widespread deployment of fog nodes with high computational capabilities and reliable energy supply [3]. However, the theoretical foundations for optimizing distributed fog computing systems to meet the demands of the TI are still lacking. In particular, computationally intensive services requiring low latencies generally demand more energy consumption from fog nodes. At the same time, many TI applications involve portable devices such as robots, drones, and vehicles with limited power supplies. In addition, the fast growing power consumption of information and communication technologies and its impact on climate change have recently raised significant concerns

Manuscript received April 15, 2018; accepted August 18, 2018. Part of the results in this paper was presented at the IEEE INFOCOM Conference, Atlanta, May 2017 [1]. This research was supported in part by NSF (grants IIP-1265960, CNS-1409172, CNS-1513649, CNS-1563655, and CNS-1731164) and by the Broadband Wireless Access & Applications Center. Any opinions, findings, conclusions, or recommendations expressed in this paper are those of the author(s) and do not necessarily reflect the views of NSF.

Y. Xiao is with the School of Electronic Information and Communications at the Huazhong University of Science and Technology, Wuhan, China (e-mail: xyong.2012@gmail.com).

M. Krunz is with the Department of Electrical and Computer Engineering at the University of Arizona, Tucson, AZ (e-mail: krunz@email.arizona.edu) and is also affiliated with the University Technology Sydney (UTS), Sydney, Australia

in both industry and academia [10], [11]. Existing cloud data centers in the US have already constituted more than 2% of the country's total electricity usage. Power consumption is expected to be significantly increased with the deployment of a large number of fog computing servers throughout the world. How to improve the efficiency of the power usage for fog computing networks while taking into consideration the stringent requirements of the TI services is still an open problem.

Another issue is that, in contrast to other applications, workload generated by the TI can exhibit much higher temporal and geographical variations due to the bursty nature of human-generated traffic. For example, an autonomous vehicle that is trying to pass another will create a much larger computational workload and require much lower latency service compared to other vehicles that stick to pre-planned routes. How to efficiently distribute and orchestrate the workload of different fog nodes for parallel execution under real-time constraints is still an open issue.

In this paper, we take steps towards addressing the above issues. In particular, we study energy-efficient workload offloading for fog computing systems that support the TI applications and services. We focus on optimizing two important performance metrics: (1) *Service response time*, including the round-trip transmission latency between users and fog nodes, queueing delays, and workload transmission and forwarding latency among fog nodes as well as that between fog nodes and CDCs; and (2) *fog nodes' power efficiency*, measured by the amount of power consumed by fog nodes to process a unit of workload. We perform detailed analysis under different scenarios and derive the optimal amount of workload to be processed by fog nodes so as to minimize the response time under a given power efficiency. We quantify the fundamental tradeoff between these two metrics.

To address the issue of skewed workload distribution among fog nodes, we study a cooperative setting in which the workload of a fog node can be partially processed by other nodes in proximity. We observe that the response time and power-efficiency tradeoff is closely related to the cooperation strategy among fog nodes. Accordingly, we propose a novel cooperation strategy called *offload forwarding*, in which each fog node can forward a part or all of its unprocessed workload to other nearby fog nodes, instead of always forwarding workload that exceeds its processing capability to a remote CDC. We study the offload allocation problem in which all fog nodes jointly determine the optimal amount of workload to be forwarded and processed by each other to further reduce the response time.

Based on our analysis, we observe that, for most TI applications, it is generally impossible to optimize the workload distribution among fog nodes in a centralized fashion due to the following reasons: (1) Deploying a central controller to calculate the amounts of workload processed by every fog node may result in intolerably high information collection and coordination delay as well as high computation complexity at the controller; (2) the workload received by each fog node can be highly dynamic, and constantly exchanging information about workload as well as computational resource availability

among fog nodes can result in network congestion; and (3) fog nodes may not want to reveal their private information. Motivated by these observations, we propose a novel distributed optimization framework for cooperative fog computing based on *dual decomposition*. Our proposed framework does not require fog nodes to have back-and-forth negotiation or disclose their private information. Two distributed algorithms are developed under the proposed framework. The first one is based on the *subgradient method with dual decomposition*. We show that this algorithm converges to the globally optimal solution with low computational complexity at each fog node. To further improve the convergence rate, we propose another distributed algorithm based on *distributed alternating direction method of multipliers via variable splitting* (ADMM-VS). We prove that ADMM-VS can converge to the globally optimal solution in linear time with a slight increase in the computational complexity at each fog node.

Motivated by the fact that the self-driving vehicle has been considered as one of the key use cases for the TI [2], [5], as a case study, we evaluate the performance of our framework by simulating a city-wide implementation of a self-driving bus system supported by a fog computing network. We analyze over 2500 traffic images of 8 existing bus routes operated at the city of Dublin and consider the scenario that these traffic data can be submitted and processed by a fog computing network deployed in a real wireless network infrastructure consisting of over 200 base stations of a major cellular operator in Ireland to ensure safe and efficient decision making and driving guidance for all the buses. We evaluate the service response time and power efficiency of fog computing networks in different areas of the city with different densities of fog node deployment. Numerical results show that our algorithms can almost double the workload processing capacity of fog nodes in urban areas with high density of fog node deployment. To the best of our knowledge, this is the first work that studies distributed workload allocation among cooperative fog nodes with energy efficiency awareness.

## II. RELATED WORK

ITU and NGMN Alliance identify fog computing as one of the key components for the TI to achieve ultra-low service latency for users [2], [4], [5]. In contrast to CDCs, fog nodes can be built much closer to users so the workload transmission latency can be significantly reduced. However, due to the limited computational capability of each fog node, offloading a large amount of workload to fog nodes will result in a high processing delay. Therefore, previous works focused on how to optimize resource provisioning of fog nodes so as to reduce the processing delay. For example, in [12], a virtual machine (VM) synthesis approach was proposed to allow each end user to quickly provision the resources of neighboring fog nodes and create the required VM images to support the requested service. The resource provisioning problem for a CDC network has been modeled in [13] as an auction-based market, where users develop bidding strategies to compete for the CDC at low costs. In [14], a service-oriented resource estimation and management framework for fog computing was introduced to maximize the resource utilization of the CDCs. The authors in

[15] introduced a system that allows fine-grained energy-aware offloading of users' mobile codes to the infrastructure. In [16], resource provisioning was investigated for tactical cloudlets. A provisioning mechanism was proposed for the infrastructure to support computation offloading and data staging at the tactical edge. In [17], a hierarchical architecture was proposed in which edge cloud servers are organized into different tiers according to their distances to the edge. If the workload received by an edge cloud server of a given tier exceeds its computational capacity, the extra workload is forwarded to higher-tier servers. In [18], a scalable online algorithm was introduced to optimize the workload dispatched to a fog node.

One of the premises of fog computing is that a large number of fog nodes are widely deployed across a large geographical area. This, however, can result in a significant increase in energy consumption. In addition, some fog computing systems such as computational ferry [19], mobile cloudlet [20] and fog nodes deployed in tactical environments [21], have limited access to reliable power sources (e.g., power grid). Therefore, developing energy-efficient solutions for fog computing is critical to ensure the sustainability, availability, and ubiquity of services. In [22], an energy adaptive scheme was proposed for fog nodes to operate at different transmit powers with variable data rates. The authors in [23] compared the energy consumption of applications using centralized CDCs with applications that use nano data centers under the fog computing architecture. They verified that the most energy-efficient strategy for content storage and distribution in cloud applications is a combination of centralized data centers and distributed fog nodes.

In this paper, we evaluate our proposed fog node cooperation framework by simulating a fog computing-supported self-driving bus system. Most existing self-driving vehicle platform, including Google's Waymo self-driving car project [24] and Stanford's Junior self-driving vehicle platform [25], focus on the scenarios that the driving decision will be made by an on-board computer according to a pre-set policy and/or human instructions<sup>1</sup>. Recently, fog computing-supported self-driving systems have attracted significant interest [27], [28]. In contrast to the existing on-vehicle computing solution, fog computing-assisted driving guidance and decision making can support high-performance image and video processing as well as high capacity storage for storing high-definition maps with instantaneous traffic updating.

So far, the impact of the energy-efficient fog computing design on the service response time has not been well investigated. Motivated by this observation, in this paper, we study the relationship between the response time of users and the power efficiency of fog nodes considering the possibility of fog node cooperation to support various TI applications.

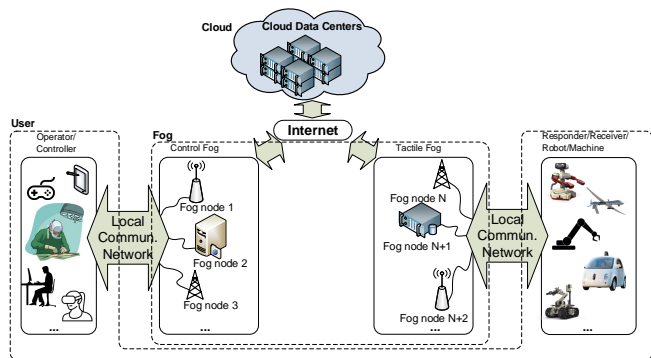


Fig. 1. Fog computing-supported Tactile Internet architecture.

### III. FOG COMPUTING-SUPPORTED TACTILE INTERNET ARCHITECTURE

A generic fog computing-supported Tactile Internet architecture consisting of four major components as illustrated in Figure 1 [2], [4], [5]:

- 1) *Operator*—A human operator and/or human-to-machine (H2M) interface that can manipulate virtual and/or real objectives using various input signals such as human gestures, touch, and voice. In some applications such as self-driving vehicles and autonomous robots, operators can also correspond to pre-calculated policies that can mimic the human behaviors/decision making processes. Operators can also expect feedback within a given time duration depending on the particular applications. For example, a real-time AR/VR game may require as low as 1ms response time. Other applications, such as remotely controlled robots, can tolerate up to one second response time.
- 2) *Responder*—One or multiple teleoperators (remotely controlled robots, machines, drones, etc.) that can be directly controlled by the operators. Responders interact with the environment and send feedback signals to the operators.
- 3) *Fog*—The composition of a number of low-cost fog nodes that are characterized by limited computing capabilities and power supplies. Fog nodes are deployed closer to end-users. According to the types of users served by fog nodes, the fog can be further divided into *control fog* and *tactile fog*. Control fog consists of fog nodes that can support computation-intensive services, such as analog-to-digital conversion, coding, signal processing, data compression, etc. Tactile fog consists of the fog nodes that are responsible for processing, compressing, and sending feedback data generated through the interactions between responders and the environment. Fog nodes may take the form of mini-servers within the wireless edge network infrastructure, reside in base stations (BSs), roadside units (RSUs), etc. Each fog node serves a distinct set of users in its coverage area. Users first submit their workloads to the closest fog node. Each fog node will then need to carefully decide the amount of workload it can process locally. If multiple closely located fog nodes can communicate with

<sup>1</sup>Society of Automotive Engineers (SAE) International defines autonomous driving systems into six levels: no automation, driver assistance, partial automation, conditional automation, high automation, and full automation. Note that even when a vehicle has been defined as full automation, human intervention is required under emergency situations [26].

each other (e.g., using high-speed backhaul connection links in a cellular network), some fog nodes may forward part of their workload to other nearby nodes to further improve the processing capability and balance the workload among nodes. 4) *Cloud*–Large-scale CDCs equipped with powerful processing units. These data centers are often built in remote areas, far from end users.

Fog nodes and users (operators or responders) may correspond to the same type of devices. For example, in some distributed mobile computing grid systems (e.g., IBM’s world community grid project), the idle computing resources of some computers (i.e., fog nodes) can be used by other computers (i.e., users) to perform computationally intensive tasks.

We consider a fog computing system that contains a set of  $N$  fog nodes  $\mathcal{F} = \{1, 2, \dots, N\}$ . Any user can be associated with one or more of these fog nodes. The association between users and fog nodes can be made based on physical proximity, channel conditions, or prior agreements between users and network service provider. For example, if the fog is deployed by the cloud provider, users can send their service requests to CDCs following the same procedure of a traditional cloud computing system. The cloud provider can then delegate one or more nearby fog nodes to process the workload submitted by these users. Each fog node  $j$  can process a nonnegative portion  $\alpha_j$  of its received workload using its local resources. Remaining workload, if any, is forwarded to the cloud. Note that  $\alpha_j = 1$  means that fog node  $j$  will process all its received workload. The workload arrival rate at each fog node  $j$ , denoted by  $\lambda_j$ , is assumed to be fixed.

We focus on two performance metrics:

1) *(Service) response time of end-users*: The response time includes the round-trip time for transmitting the workload between a user and the associated fog node as well as the queueing delay at the fog. Given their proximity to users, fog nodes are likely to exhibit smaller transmission times than remote CDCs. However, due to their limited resources, fog nodes that process a large amount of workload will likely have a long queueing delay. Therefore, it is important to balance the workload offloaded by fog nodes. Note that the response time associated with fog node  $i$ , denoted as  $R_i(\alpha_i)$ , depends on the portion of workload locally processed by fog node  $i$ .

2) *Power efficiency of fog nodes*: We consider the power efficiency by the amount of power spent on processing a unit of received workload. Maximizing the power efficiency amounts to minimizing the power consumption for processing a given workload. It is known that the total amount of power consumed by any electronic device (e.g., a fog node) depends on the power usage effectiveness (PUE) as well as the static and dynamic power consumption. PUE is the input power from the power grid divided by the power consumption of the given device. Static power consumption, also called leakage power, is mainly caused by the leakage currents, and is unrelated to the usage of the computing resources at a fog node. Dynamic power consumption is the result of the circuit activity and is determined by the activity of computing resources. Let  $e_i$  and  $w_i^S$  be the PUE and static power consumption of fog node  $i$ , respectively. Let  $w_i^D$  be the dynamic power consumed by fog node  $i$  to offload each unit of workload. We can write

the total power consumption of fog node  $i$  per time unit as  $w_i = e_i (w_i^S + w_i^D \alpha_i \lambda_j)$ . The power efficiency of fog node  $i$  can then be written as

$$\eta_i(\alpha_i) = \frac{w_i}{\alpha_i \lambda_i} = e_i \left( \frac{w_i^S}{\alpha_i \lambda_i} + w_i^D \right). \quad (1)$$

One of the main objective of this paper is to develop workload allocation strategies for determine the appropriate portion of workload to be processed locally so as to minimize the response time under given power-efficiency constraints. Formally, each fog node  $i$  tries to find the optimal value  $\alpha_i^*$  by solving the following optimization problem:

$$\begin{aligned} \alpha_i^* &= \arg \min_{\alpha_i \in [0,1]} R_i(\alpha_i) \\ \text{s.t. } \eta_i(\alpha_i) &\leq \bar{\eta}_i, \end{aligned} \quad (2)$$

where  $\bar{\eta}_i$  is the maximum power efficiency that can be supported by the hardware of fog node  $i$ . In Section IV, we will give a more detailed discussion of the response time of fog node  $i$  under different scenarios.

As mentioned earlier, different fog nodes can have different workload arrival rates. Therefore, allowing fog nodes to cooperate with each other and jointly process their received workload can further improve the overall workload processing capability. Specifically, fog nodes that receive more workload than their processing capabilities can seek help from nearby fog nodes with surplus computing resources. The main objective in this case is to minimize the average response time of users associated with all cooperative fog nodes. The total amount of workload processed by each fog node in this case will not only depend on its own received workload, but also on the workload forwarded from other fog nodes. We can write the response time of fog node  $i$  under cooperation as  $R_i^C(\boldsymbol{\alpha})$  where  $\boldsymbol{\alpha} = \langle \alpha_1, \alpha_2, \dots, \alpha_N \rangle$ . The optimal workload distribution under cooperative fog computing can then be written as

$$\begin{aligned} \boldsymbol{\alpha}^* &= \arg \min_{\boldsymbol{\alpha}} \sum_{i \in \mathcal{F}} R_i^C(\boldsymbol{\alpha}) \\ \text{s.t. } \eta_i(\alpha_i) &\leq \bar{\eta}_i, 0 \leq \alpha_i \leq 1, \forall i \in \mathcal{F}. \end{aligned} \quad (3)$$

Later on, we provide a more detailed discussion of the strategies for cooperative fog computing.

## IV. RESPONSE TIME AND POWER EFFICIENCY TRADEOFF

### A. Response Time Analysis and Minimization

Let  $\tau_j^u$  be the average round trip time (RTT) between fog node  $j$  and its users. Typically, fog nodes and CDCs have fixed locations. Thus, we assume the average workload transmission time between each fog node  $j$  and the cloud can be regarded as a constant denoted as  $\tau^f$ . Node  $j$  can directly forward its received workload to CDCs through the backbone IP network [6]. In this case, the fog computing network becomes equivalent to the traditional cloud computing network with all the workload being processed by the cloud. As mentioned before, CDCs are generally installed with high-performance workload processing units, and therefore their processing times are much smaller than the workload transmission time [29]. For simplicity, we ignore the processing time of CDCs. In

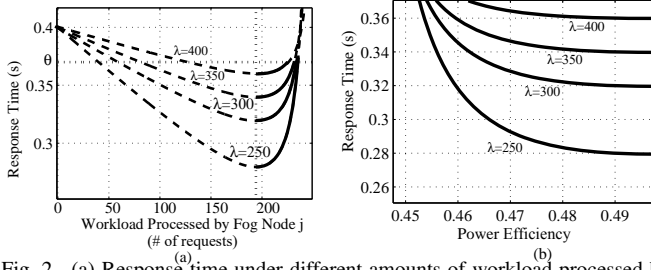


Fig. 2. (a) Response time under different amounts of workload processed by fog node  $j$ . (b) response time under different power efficiency values.

this case, the response time of fog node  $j$  can be written as  $R_j^{W1} = \tau_j^u + \tau^f$ . Since in this case fog node  $j$  does not activate any computing resources to process its received workload, the power efficiency will not depend on the response time.

In another extreme case, node  $j$  may process all its received workload using its local computing resources, i.e.,  $\alpha_j = 1$ . If we follow a commonly adopted setting and consider an M/M/1 queueing system for each fog node to process the received request, we can write the response time of fog node  $j$  as  $R_j^{W2}(\lambda_j) = \tau_j^u + \frac{1}{\mu_j - \lambda_j}$  where  $\mu_j$  is the maximum amount of workload that can be processed by the on-board computing resources of fog node  $j$ . We have  $\lambda_j \leq \mu_j$ .

Compared to CDCs, each fog node can only have limited computing resources. It is generally impossible to always allow each fog node to process all the received workload. We now consider the cases that fog node  $j$  processes only a portion  $\alpha_j$ ,  $0 \leq \alpha_j < 1$ , of its received workload and forwards the remaining  $1 - \alpha_j$  of its workload to CDCs, i.e., we still require  $\alpha_j \lambda_j < \mu_j$ . We can write the expected response time for fog node  $j$  as:

$$R_j^{W3}(\alpha_j) = \tau_j^u + \alpha_j \left( \frac{1}{\mu_j - \alpha_j \lambda_j} \right) + (1 - \alpha_j) \tau^f. \quad (4)$$

Consider the solution of problem (2) by substituting the response time equation in (4). We can observe that problem (2) is a convex optimization problem, and hence can be solved using standard approaches. We omit the detailed derivation and directly present the solutions of these problems as follows. The minimum response time for users associated with fog node  $j$  is  $R_j^{W3}(\alpha_j^*)$ , where  $\alpha_j^*$  has the following closed-form solution:

$$\alpha_j^* = \begin{cases} 1, & \text{if } \mu_j < \frac{\lambda_j}{\tau^f + 1}, \\ \frac{1}{\lambda_j} \left( \frac{w_j^s e_j}{\bar{\eta}_j - e_j w_j^p} \right), & \text{if } \mu_j \geq \frac{\chi_j}{2\chi_j - \lambda_j(1 - \tau^f)}, \\ \frac{\mu_j}{\lambda_j} - \frac{\mu_j}{\lambda_j} \sqrt{1 - \frac{\lambda_j}{\mu_j} (1 - \tau^f)}, & \text{Otherwise,} \end{cases} \quad (5)$$

where  $\chi_j \triangleq \frac{w_j^s e_j}{\bar{\eta}_j - e_j w_j^p}$  is the maximum amount of workload that can be processed by fog node  $j$  under power efficiency constraint  $\eta_j(\alpha_j) \leq \bar{\eta}_j$ .

### B. Tradeoff between Response Time and Power Efficiency

In Figure 2(a), we consider a single fog node serving 5 users, and we compare the response time under different amounts of workload (number of requests) processed by the fog node. There exists an optimal amount of workload to be processed by fog node  $j$  that minimizes the response time. As observed in (1), the power consumption for the fog node to

process one unit of workload decreases with the total amount of processed workload. In many practical applications, there is a maximum tolerable response time for users. We can therefore observe that the power efficiency maximization solution for the fog node in this case will be achieved when the response time approaches the maximum tolerable service response time  $\theta$ . In Figure 2(a), we use a solid line to highlight the segment between the response time minimization solution and the power efficiency maximization solution at the maximum tolerable response time  $\theta$ . We can observe a fundamental tradeoff between the response time and the power efficiency of the fog node. This tradeoff can be characterized by substituting (4) into (2), as shown in Figure 2(b). We can observe that starting from the power consumption minimization point, the response time decreases with the power consumption for the fog node to process each unit (request) of workload. As the power consumption of the fog node continues to grow, the rate of reduction in the response time decreases. This means that, for non-delay-sensitive applications such as voice/image processing services (e.g., voice/image recognition), the fog node can choose a low power consumption solution as long as the resulting response time is tolerable for the users. On the other hand, for delay-sensitive applications such as online gaming and virtual reality (VR), it is ideal for the fog node to choose a high power consumption solution to satisfy users' low latency requirement. In Figure 2(b), we also present the tradeoff solutions with different workload arrival rates at fog node  $j$ . We can observe that the response time increases with the workload arrival rate under a given power efficiency. The higher the workload arrival rate, the smaller the changes in the response time. As the amount of workload processed by a fog node approaches its maximum processing capability, the response time approaches infinite. In other words, allowing the fog node to handle all its arriving workload cannot always reduce the response time for end-users especially when the amount of workload to be processed by the fog node cannot be carefully chosen.

## V. COOPERATIVE FOG COMPUTING

### A. Response Time Analysis for Cooperative Fog Computing with $N$ Fog Nodes

In cooperative fog computing network, We introduce a fog node cooperation strategy, referred to as offload forwarding. In this strategy, each fog node can forward a part or all of its offloaded workload to multiple neighboring fog nodes in the fog and/or help multiple other fog nodes process their workloads. Each node  $j$  divides its received workload into  $N + 1$  partitions:  $\varphi_{j1}, \varphi_{j2}, \dots, \varphi_{jN}, \varphi_{jc}$  where  $\varphi_{jc}$  is the workload forwarded to the remote CDC and  $\varphi_{ji}$ ,  $i \in \mathcal{F} \setminus \{j\}$ , is the workload forwarded to fog node  $i$  (this includes  $\varphi_{jj}$ , the workload processed by node  $j$  itself). We denote  $\varphi_{j\bullet} \triangleq \langle \varphi_{jk} \rangle_{k \in \mathcal{F}}$ . Note that it is not necessary for each fog node to always forward a non-zero workload to other fog nodes, i.e.,  $\varphi_{ji} = 0$  means that fog node  $i$  does not process any workload for fog node  $j$ . We refer to  $\varphi_{j\bullet}$  as the request vector of fog node  $j$ . We also refer to  $\varphi_{\bullet i} = \langle \varphi_{ji} \rangle_{j \in \mathcal{F}}$  as the service vector of fog node  $i$ . Let  $\varphi = \langle \varphi_{ji} \rangle_{i, j \in \mathcal{F}}$  be the workload processing matrix for the entire fog. We have  $0 \leq \varphi_{jk} \leq 1$

and  $\sum_{k \in \mathcal{F}} \varphi_{jk} \leq 1, \forall j \in \mathcal{F}$ . The response time of fog node  $j \in \mathcal{F}$  can then be written as

$$R_j^C(\xi_j, \phi_{j\bullet}) = \tau_j^u + \frac{1}{\sum_{i \in \mathcal{F}} \lambda_i} \sum_{i \in \mathcal{F}} \phi_{ji} \left( \tau_{ji} + \frac{1}{\mu_i - \sum_{k \in \mathcal{F}} \phi_{ki}} \right) + \varphi_{jc} \tau^c, \quad (6)$$

where  $\varphi_{jc} = 1 - \sum_{i \in \mathcal{F}} \varphi_{ji}$ ,  $\phi_{jk} = \lambda_j \varphi_{jk}$  is the amount of workload processed by fog node  $k$  for fog node  $j$ . Please see [1] for more detailed explanation of (6). Note that if fog node  $j$  cannot help other fog nodes to process their workload, but forward its own workload to other fog nodes to process, we have  $\phi_{kj} = 0$  and  $\phi_{ji} \neq 0 \forall k, i \in \mathcal{F} \setminus \{j\}$ .

We can rewrite the optimization problem in (3) as follows:

$$\min_{\phi_{1\bullet}, \dots, \phi_{N\bullet}} \sum_{j=1}^N R_j^C(\xi_j, \phi_{j\bullet}) \quad (7)$$

$$\text{s.t.} \quad \sum_{k \in \mathcal{F}} \phi_{jk} + \phi_{jc} = \lambda_j, \quad (8)$$

$$\sum_{k \in \mathcal{F}} \phi_{kj} \leq \chi_j \quad \text{and} \quad 0 \leq \phi_{jk} \leq \lambda_j, \forall k, j \in \mathcal{F}. \quad (9)$$

It can be observed that, in order for each fog node  $j$  to calculate the portions of workload to be forwarded to other fog nodes, fog node  $j$  needs to know the workload processing capabilities and the workload arrival rates of all the other fog nodes, which can be private information and impossible to be known by fog node  $j$ . In the next section, we will propose a distributed optimization framework to allow all the fog nodes to jointly optimize the average response time of the fog without disclosing their private information.

### B. Response Time and Power Efficiency Tradeoff for Cooperative Fog Computing Networks

In Figure 3(a), we present the minimum response time of the fog in a cooperative fog computing network derived from solving problem (7). Note that the workload processed by each fog node can consist of both its own received workload and the workload sent from other fog node. We can observe that the response time of the fog is closely related to the amount of workload processed by each fog node. We also use black grid to highlight the area between the response time minimization solution and the power efficiency maximization solution with a given maximum tolerable response time in Figure 3(a). By substituting the power efficiency defined in (1) into (7), we can also present the relationship between the fog's response time and each fog node's power efficiency for a two-node cooperative fog computing network with offload forwarding in Figure 3(b). Similar to the single-node fog computing, we can observe a fundamental tradeoff between the response time of all the users served by the fog and the power efficiency of each fog node. In addition, we can observe that by allowing offloading forwarding, even if the power consumption of each fog node to process each unit of workload has been limited to a very small value, it is still possible to achieve the response time constraint if there exist other nearby fog nodes with users that are more delay tolerant.

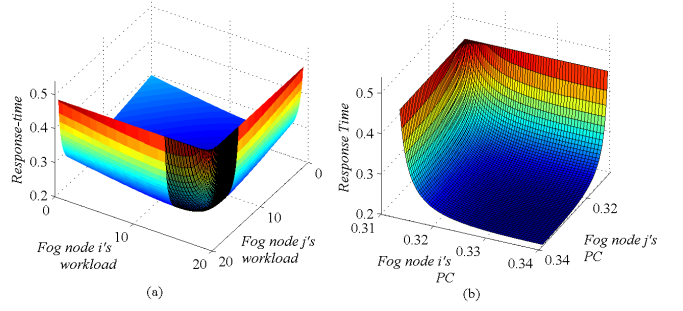


Fig. 3. Response time under different amounts of processed workload and power consumptions (PC) for each fog node to offload one unit of workload.

## VI. DISTRIBUTED OPTIMIZATION FOR COOPERATIVE FOG COMPUTING

As mentioned previously, deciding the proper amount of workload to be processed by each fog node is essential to achieve the optimal response time and power efficiency tradeoff for fog computing networks. Unfortunately, solving problem (7) involves carefully deciding the amounts of workload processed and forwarded by every individual fog node according to global information such as the computational capacities of all the fog nodes and the round-trip workload transmission latency between any two fog nodes as well as that between fog nodes and cloud. Deploying a centralized controller to collect all these global information and calculate the optimal service and request vectors for all the fog nodes may result in a huge communication overhead and intolerably high information collection and processing delay. In addition, it can also be observed that (7) is non-smooth and therefore cannot be solved by traditional optimization approaches that can only handle smooth objective functions.

To address the above challenges, we need to develop a distributed framework that can solve problem (7) with the following two main design objectives:

*O1) Distributed and Scalable:* we would like to develop a framework that can separate the optimization problem in (7) into  $N$  sub-problems each of which can be solved by each fog node using its local information. The framework should also be scalable in the sense that the computation complexity for each fog node to solve its sub-problem should not increase significantly with the number of fog nodes that have the potential to cooperate with each other.

*O2) Privacy Preserving:* Each fog node may not be willing to reveal its private proprietary information such as the maximum computational capacity and the round-trip workload transmission latency to others.

We propose a novel distributed optimization framework based on dual decomposition in which problem (7) will be first converted into its Lagrangian form and then the converted problem will be decomposed into  $N$  subproblems each of which can be solved by an individual fog node using its local information. The optimization of all the subproblems will be coordinated through dual variables sent to a workload forwarding coordinator (WFC) which can be established by the cloud data centers or deployed as one of the virtualized components in the cloud. We propose two distributed algorithms:



subgradient method with dual decomposition and distributed ADMM-VS, both of which can achieve the global optimal solution of problem (7) and satisfy objectives O1) and O2).

#### A. Subgradient Method with Dual Decomposition

Before we introduce the algorithm, we need to first remove the inequality constraints in problem (7) by introducing a set of indicator functions. In particular, let us introduce  $N + 1$  indicator functions that include each of separable inequality constraints in (9) and incorporate these indicator functions into the objective function of problem (7). More specifically, we define  $\mathcal{G}_i = \{\phi_{\bullet i} : \sum_{k \in \mathcal{F}} \phi_{ki} \leq \chi_i, 0 \leq \phi_{ki} \leq \lambda_k, \forall k \in \mathcal{F}\}$  as the polyhedra of each constraint corresponding to fog node  $i$  in problem (3) where  $\phi_{\bullet i} = \langle \phi_{ki} \rangle_{k \in \mathcal{F} \setminus \{i\}}$  is the vector of amounts of workload to be processed by fog node  $i$  for other fog nodes. We define an indicator function  $\mathbf{I}_{\mathcal{G}_i}(\phi_{\bullet i})$

$$\mathbf{I}_{\mathcal{G}_i}(\phi_{\bullet i}) = \begin{cases} 0, & \phi_{\bullet i} \in \mathcal{G}_i, \\ +\infty, & \phi_{\bullet i} \notin \mathcal{G}_i. \end{cases} \quad (10)$$

By including the above indicator functions into the objective function of our optimization problem, we can convert the original problem (3) with inequality constraints into the following optimization problem without inequality constraints.

$$\begin{aligned} \min_{\phi_{1\bullet}, \dots, \phi_{N\bullet}} & \sum_{i \in \mathcal{F}} R_i^C(\xi_i, \phi_{i\bullet}) + \sum_{j \in \mathcal{F} \cup \{c\}} \mathbf{I}_{\mathcal{G}_j}(\phi_{\bullet j}) \quad (11) \\ \text{s.t.} & \sum_{k \in \mathcal{F} \cup \{c\}} \phi_{ik} = \lambda_i, \forall i \in \mathcal{F}. \end{aligned}$$

We can then write the Lagrangian form of problem (11) as

$$\begin{aligned} \mathcal{L}(\phi_{1\bullet}, \dots, \phi_{N\bullet}, \Lambda) \\ = \sum_{i \in \mathcal{F}} R_i^C(\xi_i, \phi_{i\bullet}) + \sum_{j \in \mathcal{F} \cup \{c\}} \mathbf{I}_{\mathcal{G}_j} \\ - \left( \sum_{j \in \mathcal{F} \cup \{c\}} \Lambda \phi_{\bullet j} - \Lambda \lambda^\dagger \right) \quad (12) \end{aligned}$$

where  $\Lambda$  is the vector of dual variables,  $\lambda = \langle \lambda_i \rangle_{i \in \mathcal{F}}$  and  $\cdot^\dagger$  is the transpose.

We can therefore write the optimization of the Lagrangian form as follows:

$$\langle \phi_{1\bullet}, \dots, \phi_{N\bullet} \rangle = \arg \min_{\phi_{\bullet i}} \mathcal{L}(\phi_{1\bullet}, \dots, \phi_{N\bullet}, \Lambda^t). \quad (13)$$

The dual variable  $\Lambda$  can be updated using

$$\Lambda^{t+1} = \Lambda^t - \varrho_t \left( \sum_{j \in \mathcal{F} \cup \{c\}} \Lambda \phi_{\bullet j} - \Lambda \lambda^\dagger \right), \quad (14)$$

where  $\varrho_t > 0$  is the step-size of the iteration.

We can observe that (13) can be reduced to solving  $N$  individual sub-problems each of which can be solved by an individual fog node  $i$  by optimizing the vector of workloads  $\phi_{\bullet i}$  to be processed using its local computing resources, i.e., each fog node  $i$  decides values of  $\phi_{\bullet i}$  and  $\varphi_{ic}$  by solving the following sub-problem:

$$\langle \phi_{\bullet i}^{t+1}, \varphi_{ic}^{t+1} \rangle = \arg \min_{\langle \phi_{\bullet i}, \varphi_{ic} \rangle} \mathcal{L}_{S_i}(\phi_{\bullet i}, \varphi_{ic}, \Lambda^t), \quad (15)$$

where  $\mathcal{L}_{S_i}(\phi_{\bullet i}, \varphi_{ic}, \Lambda^t) = S_i(\phi_{\bullet i}) + \mathbf{I}_{\mathcal{G}_i}(\phi_{\bullet i}) - \Lambda^t \phi_{\bullet i} - \Lambda_i^t \varphi_{ic}$  and  $S_i(\phi_{\bullet i})$  is given by

$$\begin{aligned} S_i(\phi_{\bullet i}, \varphi_{ic}) &= \tau_i^u + \frac{1}{\sum_{i \in \mathcal{F}} \lambda_i} \sum_{j \in \mathcal{F}} \phi_{ji} \left( \tau_{ji} + \frac{1}{u_i - \sum_{k \in \mathcal{F}} \phi_{ki}} \right) \\ &+ \varphi_{ic} \tau^c. \end{aligned} \quad (16)$$

We can prove the following result.

**Theorem 1:** The Lagrangian form of the objective function of our optimization problem in (7) is separable and convex.

*Proof:* We can directly prove the separability of the Lagrangian problem in (12) by verifying  $\mathcal{L} = \sum_{i \in \mathcal{F}} \mathcal{L}_{S_i}$ . Let us now prove that the objective function of problem (7) is also convex. It can be directly shown that the domain of variables in the objective function of (7) is a polyhedra which is a convex set. We can also show that the second derivative of each individual item in  $S_i(\phi_{\bullet i})$  is always positive which means that it is a convex function with respect to each individual variable. Following the property that a nonnegative weighted sum of convex function  $f = \sum_{i=1}^N c_i f_i$ ,  $f : \mathbf{R}^N \rightarrow \mathbf{R}$  is convex if and only if  $f_i$  is convex and  $c_i$  is a constant for all  $i \in \{1, 2, \dots, N\}$ , we can prove that the objective function of problem (3) is convex. This concludes the proof. ■

It has been proved that if step-size  $\varrho_t$  satisfies the following diminishing conditions  $\sum_{t=1}^{\infty} \varrho_t^2 < \infty$  and  $\sum_{t=1}^{\infty} \varrho_t = \infty$ , the subgradient method with dual decomposition is guaranteed to converge to the optimal solution [30]. A common choice of step-size is  $\varrho_t = \bar{\varrho} / \sqrt{k}$  where  $\bar{\varrho}$  is a constant. One of the main advantage of the subgradient method is its low computational complexity for each fog node. However, it has been proved that the convergence rate of subgradient method is given by  $O(1/\sqrt{t})^2$  which is slow.

#### B. Distributed ADMM via Variable Splitting

In this section, we propose a distributed optimization framework based on distributed ADMM-VS.

Similarly, we introduce an indicator function  $\mathbf{I}_{\mathcal{G}_c}(\psi)$  to characterize the inseparable constraint in (8)

$$\mathbf{I}_{\mathcal{G}_c}(\psi) = \begin{cases} 0, & \psi \in \mathcal{G}_c, \\ +\infty, & \psi \notin \mathcal{G}_c, \end{cases} \quad (17)$$

where  $\psi = [\psi_1, \psi_2, \dots, \psi_N]$ ,  $\mathcal{G}_c = \{\psi : \sum_{i=1}^N I_N \psi_i \leq 1\}$ ,  $I_N$  is an identity matrix with size  $N$ ,  $\psi \in \mathbf{R}^{N \times N}$ ,  $\psi_i \in \mathbf{R}^N$ . We can show that the solution of the optimization problem in (11) is equivalent to solving the optimization problem with the following augmented Lagrangian form with two blocks of random variables. We can write the  $\phi$ -optimization subproblem as

$$\begin{aligned} \phi^{t+1} &= \arg \min_{\phi} \mathcal{L}_{\rho}(\phi_{\bullet 1}, \phi_{\bullet 2}, \dots, \phi_{\bullet N}, \psi^t, \Lambda^t) \\ &= \arg \min_{\phi} \sum_{i \in \mathcal{F}} \{ R_i^C(\xi_i, \phi_{i\bullet}) + \mathbf{I}_{\mathcal{G}_i}(\phi_{\bullet i}) \\ &+ \frac{\rho}{2} \|\phi_{\bullet i} - \psi_i^t + \Lambda_i^t\|_2^2 \}, \end{aligned} \quad (18)$$

<sup>2</sup>We follows Bachmann-Landau notations:  $f = O(g)$  if  $\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} < \infty$ .

where  $\rho$  is the augmented Lagrangian parameter and  $\Lambda$  is the vector of the dual variables.

We can write the  $\psi$ -updating problem as

$$\psi^{t+1} = \arg \min_{\psi} \frac{\rho}{2} \|\phi^{t+1} - \psi^t + \Lambda^t\|_2^2 + \mathbf{I}_{\mathcal{G}_c}(\psi). \quad (19)$$

The dual variable update sub-problem can then be written as follows

$$\Lambda^{t+1} = \Lambda^t - \rho(\phi^{t+1} - \psi^{t+1}). \quad (20)$$

We can observe that the subproblem optimization in (18)–(20) is equivalent to the form of the traditional ADMM with two random variables:  $\phi$  and  $\psi$ . In D-ADMM-VS, each fog node  $i$  will calculate the optimal service vector  $\phi_{\bullet,i}^*$  by solving the following sub-problem:

$$\phi_{\bullet,i}^{t+1} = \arg \min_{\phi_{\bullet,i}} \mathcal{L}'_{S_i}(\phi_{\bullet,i}, \psi_i^t, \Lambda_i^t) \quad (21)$$

We present the detailed description of ADMM-VS below.

---

#### Algorithm 2: Distributed ADMM-VS Algorithm

---

*Initialization:* Each fog node  $i$  chooses an initial service vector  $\phi_{\bullet,i}^0$  and WFC chooses an initial dual variable  $\Lambda^0$ .

WHILE  $t=0, 1, \dots$

- i) *Fog node updating:* Each fog node  $i$  calculates  $\phi_{\bullet,i}^{t+1}$  by solving (21) and then sends the resulting  $\phi_{\bullet,i}^{t+1}$  and  $\lambda_k$  to the WFC,
- ii) *WFC Updating:* WFC calculates  $\psi^{t+1}$  by solving  $\psi$ -updating problem in (19).
- iii) *Dual Variable Updating:* WFC updates dual variables  $\Lambda^{t+1} = \Lambda^k - \rho(\phi^{t+1} - \psi^{t+1})$  and sends  $\phi_i^{t+1}$  and  $\Lambda_i^{t+1}$  to fog node  $i$ .

ENDWHILE

---

We have the following result.

**Theorem 2:** Our proposed D-ADMM-VS algorithm converges to the global optimal solution of Problem (3) with convergence rate of  $O(1/t)$ .

*Proof:* The convergence of Algorithm 2 follows directly from the standard ADMM approach [31]. We omit the detailed description due to limit of space. ■

We evaluate the the convergence performance of our proposed algorithms in Figure 5. We can observe that both our proposed algorithms can converge to the global optimal solution within the first few iterations (less than 14 iterations in both cases) as shown in Figure 5. We also present the convergence rate when a centralized ADMM method in [31] is applied to solve optimization problem (7). In this method, a centralized controller can collect all the information from fog nodes and calculate the amount of workload to be processed by each fog node. We can observe that our proposed distributed ADMM-VS presents a similar convergence performance as the centralized ADMM approach and can approach the global optimal solution within first 10 iterations which is much faster than the subgradient method.

## VII. CASE STUDY: A CITY-WIDE DEPLOYMENT OF FOG COMPUTING-SUPPORTED SELF-DRIVING BUS SYSTEM

### A. Simulation Setup for Traffics Generated by Self-driving Bus

In this section, we consider a possible implementation of a fog computing-supported self-driving bus system in the city of Dublin as a case study. A self-driving vehicle relies on

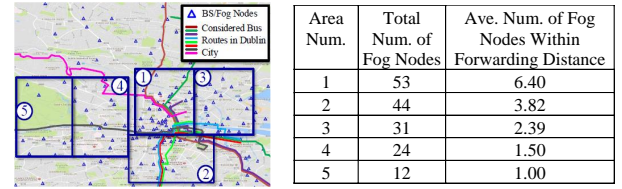


Fig. 4. (a) Distribution of fog nodes, bus routes, and considered areas, (b) deployment density of fog nodes in each considered area, and (c) empirical probability distribution of traffics generated by self-driving buses in each considered area.

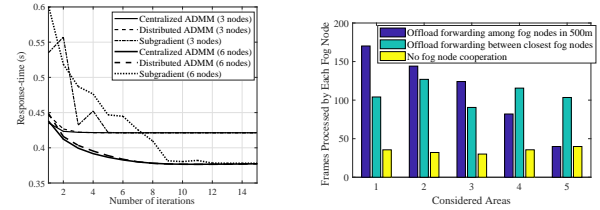


Fig. 5. Convergence rate of the proposed algorithms compared to a centralized ADMM approach.

a combination of sensors including cameras, sonars, radars, light detection and ranging systems (LiDARs), etc., to sense the surrounds and decide the driving behaviors. More specifically, image and sensing data collected by the sensors will be processed by a computer or processor to extract useful information such as the types of objects as well as their specific semantics that may affect the driving decisions in various scenarios. For example, an autonomous vehicle must be able to detect and recognize various types of unintelligent objects such as traffic/road work signs and traffic lights as well as intelligent objects including surrounding vehicles, animals, and pedestrians. It is known that accurate and low-latency object recognition require significant computing power and energy supply [25]. How to develop effective object recognition methods for autonomous driving vehicles is out of the scope of this paper.

In this section, we focus on the scenario that each self-driving bus relies on a fog node in proximity to process the traffic image and feedback the driving decision. We focus on the workload transmission and forwarding between vehicles and the fog nodes. It is known that, for each self-driving vehicle, the amount of data that needs to be collected and processed is different when it drives into different areas.

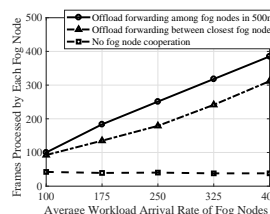


Fig. 7. Average workload (number of requests) processed by each of users with different power efficiency and arrival rates.

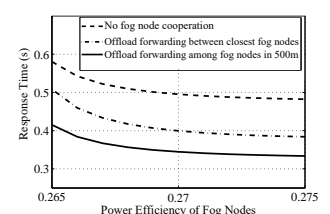


Fig. 8. Average response time of requests processed by each of users with different power efficiency and workload density.



For example, the traffic condition in the city center will be much more complex than that in the countryside. To take into consideration of the geographical diversity of the traffic data generated by each bus, we analyze the statistic feature of the traffics generated by buses operated at 8 existing routes in the city of Dublin. We generate driving videos from over 2500 stitched high-resolution street view images in the considered bus routes extracted from Google street view [32]. We then apply H.265 coding to compress the generated driving videos and keep track of the recorded frame rates of the compressed video to simulate the possible traffic data streaming of each self-driving bus. H.265 is a high efficiency video coding technique that can remove both temporal and spatial redundancies between adjacent frames using the enhanced hybrid spatial-temporal prediction model [33]. In other words, the frame rates generated by H.265 can reflect different levels of traffic complexity as well as the impact of the driving speed and surrounding traffics at different locations. We fit the recorded frame rates of each bus route using a Kernel-based probability distribution function. The bus driving routes, considered areas, and fog node distribution are shown in Figure 4(a). The deployment densities of fog nodes are listed in Figure Figure 4(b). The recorded frame sizes and fitted probability distributions of traffic generated by the self-driving bus when driving in different considered areas are presented in Figure 4(c).

### B. Simulation Setup for a Fog Computing Network

We simulate a possible implementation of fog nodes, e.g., mini-computing servers, over 200 BSs (including GSM and UMTS BSs) deployed by a primary telecom operator in the city of Dublin. The actual distribution and the deployment density of fog nodes are shown in Figure 4. Each bus always submits the traffic images (frame-by-frame) taken by its on-board camera to its closest fog node. In this case, the fog node installed at each BS will be responsible to receive and process the images sent by each bus and feedback the driving decision to each bus when the processing (e.g., objective recognition, tracking, and prediction) is finished. We assume each fog node can process at most 400 frames at the same time and the maximum tolerable response time of each bus is 500 ms. We consider two scenarios of offloading forwarding. In the first one, each fog node can only forward its workload to its closest fog node. In the second scenario, each fog node can forward part of its received workload to other fog nodes within a 500-meter range. We assume there exist local communication links among fog nodes and the round trip workload forwarding time between any two fog nodes within forwarding distance is the same given by  $\tau_{ij} = 20ms$ .

### C. Numerical Results

To evaluate the performance improvement that can be achieved by allowing offload forwarding among fog nodes, we first compare the number of frames that can be processed by each fog node in the five areas highlighted in Figure 4(a). We can observe in Figure 6 that by allowing each fog node to cooperate with all the other fog nodes within

a 500-meter range can significantly improve the numbers of frames processed by fog nodes. We can also observe that even when each fog node can only cooperate with its closest fog node, the average number of frames processed by each fog node can be almost doubled compared to the case without cooperation among fog nodes. Note that in Figure 6, we can also observe that in areas 4 and 5, allowing fog nodes within a 500-meter range cannot achieve a higher workload offloading performance than only allowing each fog node to cooperate with its closest neighboring fog node. This is because in both of these two considered rural areas, some fog nodes cannot have any other fog node located within the 500-meter range.

In Figures 7, we consider the average workload processing capability of all 5 considered areas. We investigate the impact of fog nodes' workload arrival rates on the total amount of workload to be offloaded by the fog computing network. We can observe that the average number of frames that can be offloaded by each fog node increases almost linearly when the workload arrival rate is small. However, with the workload arrival rate continuing to grow, the total amount of offloaded workload that can be offloaded by fog computing network approaches to a fixed value limited by the maximum response time that can be tolerated by end-users.

In Figure 8, we present the average response time and power efficiency tradeoff curves with and without offload forwarding. We observe that our proposed offload forwarding significantly reduces the response time of end-users especially when the power efficiency constraint of fog node  $j$  is low. With the increase of the power efficiency of fog nodes, the response time that can be provided by the fog nodes approaches a fixed value limited by the maximum workload processing capability of the fog.

## VIII. CONCLUSION

In this paper, the workload offloading problem was studied for fog computing networks. We investigated the relationship between users' response time and fog nodes' power efficiency. The tradeoff between these two metrics was discussed for fog computing network with and without fog node cooperation. For cooperative fog computing networks, we introduced a novel fog node cooperation strategy called offload forwarding. In this strategy, each fog node can forward a part of its workload to other neighboring fog nodes to further reduce the response time. We quantify the response time and power efficiency tradeoff for cooperative fog computing with offload forwarding. A distributed optimization framework based on dual decomposition has been proposed. We developed two distributed algorithms under the proposed framework. The first one is based on the subgradient method with dual decomposition and the other algorithm is based on the distributed ADMM-VS. We proved that both proposed algorithms can approach the globally optimal workload allocation solution. Finally, we have considered the possible implementation of a city-wide self-driving bus system supported by a fog computing network as a case study to verify the performance of our proposed approach. Numerical results have shown that our proposed framework can significantly improve the performance of fog computing networks.

## ACKNOWLEDGMENT

Yong Xiao would like to thank Professor Luiz A. DaSilva and Dr. Jacek Kibilda at CONNECT, Trinity College Dublin to provide the BS location data of Dublin city.

## REFERENCES

- [1] Y. Xiao and M. Krunz, "QoE and power efficiency tradeoff for fog computing networks with fog node cooperation," in *Proceeding of the IEEE INFOCOM Conference*, Atlanta, GA, May 2017.
- [2] M. Simsek, A. Ajiz, M. Dohler, J. Sachs, and G. Fettweis, "5g-enabled Tactile Internet," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 3, pp. 460–473, Mar. 2016.
- [3] NGMN Alliance, "5G white paper," Feb. 2015. [Online]. Available: [https://www.ngmn.org/uploads/media/NGMN\\_5G\\_White\\_Paper\\_V1\\_0.pdf](https://www.ngmn.org/uploads/media/NGMN_5G_White_Paper_V1_0.pdf)
- [4] M. Maier, M. Chowdhury, B. P. Rimal, and D. P. Van, "The tactile internet: vision, recent progress, and open challenges," *IEEE Communications Magazine*, vol. 54, no. 5, pp. 138–145, May 2016.
- [5] ITU-T, "The Tactile Internet," ITU-T technology watch report, Aug. 2014. [Online]. Available: [https://www.itu.int/dms\\_pub/itu-t/opb/gen/T-GEN-TWATCH-2014-1-PDF-E.pdf](https://www.itu.int/dms_pub/itu-t/opb/gen/T-GEN-TWATCH-2014-1-PDF-E.pdf)
- [6] M. Chiang, "Fog networking: An overview on research opportunities," *arXiv preprint arXiv:1601.00835*. [Online]. Available: <http://arxiv.org/pdf/1601.00835>
- [7] A. Vahid Dastjerdi, H. Gupta, R. N. Calheiros, S. K. Ghosh, and R. Buyya, "Fog Computing: Principals, Architectures, and Applications," *ArXiv preprints*, Jan. 2016. [Online]. Available: <http://adsabs.harvard.edu/abs/2016arXiv160102752V>
- [8] S. Yi, Z. Hao, Z. Qin, and Q. Li, "Fog computing: Platform and applications," in *Proc. of the IEEE HotWeb*, Washington DC, Nov. 2015.
- [9] D. Hoang, C. Lee, D. Niyato, and P. Wang, "A survey of mobile cloud computing: architecture, applications, and approaches," *Wireless communications and mobile computing*, vol. 13, no. 18, Dec. 2013.
- [10] M. Patel, B. Naughton, C. Chan, N. Sprecher, S. Abeta, A. Neal *et al.*, "Mobile-edge computing introductory technical white paper," *White Paper, Mobile-edge Computing (MEC) industry initiative*, 2014.
- [11] L. M. Vaquero and L. Rodero-Merino, "Finding your way in the fog: Towards a comprehensive definition of fog computing," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 5, Oct. 2014.
- [12] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The case for VM-based cloudlets in mobile computing," *IEEE Pervasive Computing*, vol. 8, no. 4, pp. 14–23, Oct 2009.
- [13] L. Zheng, C. Joe-Wong, C. W. Tan, M. Chiang, and X. Wang, "How to bid the cloud," *ACM SIGCOMM Computer Communication Review*, vol. 45, no. 4, pp. 71–84, 2015.
- [14] M. Aazam and E.-N. Huh, "Dynamic resource provisioning through fog micro datacenter," in *Proc. of the IEEE PerCom Workshops*, St. Louis, MO, Mar. 2015, pp. 105–110.
- [15] E. Cuervo, A. Balasubramanian, D. ki Cho, A. Wolman, S. Saroiu, R. Chandra, and P. Bahl, "Maui: Making smartphones last longer with code offload," in *Proc. of the ACM MobiSys*, Jun. 2010.
- [16] G. Lewis, S. Echeverria, S. Simanta, B. Bradshaw, and J. Root, "Tactical cloudlets: Moving cloud computing to the edge," in *Proc. of the IEEE MILCOM*, Baltimore, MD, Oct. 2014, pp. 1440–1446.
- [17] L. Tong, Y. Li, and W. Gao, "A hierarchical edge cloud architecture for mobile computing," in *Proceeding of the IEEE INFOCOM Conference*, San Francisco, CA, Apr. 2016.
- [18] H. Tan, Z. Han, X.-Y. Li, and F. C. Lau, "Online job dispatching and scheduling in edge-clouds," in *Proceeding of IEEE INFOCOM Conference*, Atlanta, GA, May 2017.
- [19] A. Monfared, M. Ammar, E. Zegura, D. Doria, and D. Bruno, "Computational ferrying: Challenges in deploying a mobile high performance computer," in *IEEE WoWMoM*, Boston, MA, Jun. 2015.
- [20] Y. Zhang, D. Niyato, and P. Wang, "Offloading in mobile cloudlet systems with intermittent connectivity," *IEEE Transactions on Mobile Computing*, vol. 14, no. 12, pp. 2516–2529, Dec 2015.
- [21] S. Echeverria, G. A. Lewis, J. Root, and B. Bradshaw, "Cyber-foraging for improving survivability of mobile systems," in *Proceeding of IEEE MILCOM Conference*, Oct 2015, pp. 1421–1426.
- [22] S. Igder, S. Bhattacharya, and J. M. H. Elmighani, "Energy efficient fog servers for internet of things information piece delivery in a smart city vehicular environment," in *International Conference on Next Generation Mobile Applications, Security and Technologies*, Aug. 2016, pp. 99–104.
- [23] F. Jalali, K. Hinton, R. Ayre, T. Alpcan, and R. S. Tucker, "Fog computing may help to save energy in cloud computing," *IEEE J. Sel. Areas in Commun.*, vol. 34, no. 5, pp. 1728–1739, May 2016.
- [24] Waymo. [Online]. Available: <https://www.waymo.com>
- [25] A. Teichman and S. Thrun, "Practical object recognition in autonomous driving and beyond," in *IEEE Workshop on Advanced Robotics and its Social Impacts*, San Francisco, CA, Oct. 2011, pp. 35–38.
- [26] V. Ilkova and A. Ilka, "Legal aspects of autonomous vehicles - An overview," in *the 21st International Conference on Process Control (PC)*, Strbske Pleso, Slovakia, Jun. 2017, pp. 428–433.
- [27] Q. Yuan, H. Zhou, J. Li, Z. Liu, F. Yang, and X. S. Shen, "Toward efficient content delivery for automated driving services: An edge computing solution," *IEEE Network*, vol. 32, no. 1, pp. 80–86, Jan 2018.
- [28] D. Grewe, M. Wagner, M. Arumathurai, I. Psaras, and D. Kutscher, "Information-centric mobile edge computing for connected vehicle environments: Challenges and research directions," in *ACM SIGCOMM 2017 Workshop on Mobile Edge Communications*, Los Angeles, CA, Aug. 2017, pp. 7–12.
- [29] M. Keller and H. Karl, "Response-time-optimized distributed cloud resource allocation," *arXiv preprint arXiv:1601.06262*, 2016. [Online]. Available: <http://arxiv.org/abs/1601.06262>
- [30] A. Falsone, K. Margellos, S. Garatti, and M. Prandini, "Dual decomposition for multi-agent distributed optimization with coupling constraints," *arXiv preprint arXiv:1607.00600v2*. [Online]. Available: <https://arxiv.org/abs/1607.00600v2>
- [31] S. Boyd *et al.*, "Distributed optimization and statistical learning via the alternating direction method of multipliers."
- [32] Google Street View. [Online]. Available: <https://www.google.com/streetview/>
- [33] G. J. Sullivan, J. R. Ohm, W. J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649–1668, Dec 2012.



**Yong Xiao** (S'09-M'13-SM'15) received his B.S. degree in electrical engineering from China University of Geosciences, Wuhan, China in 2002, M.Sc. degree in telecommunication from Hong Kong University of Science and Technology in 2006, and his Ph. D degree in electrical and electronic engineering from Nanyang Technological University, Singapore in 2012. He is now a professor in the School of Electronic Information and Communications at the Huazhong University of Science and Technology (HUST), Wuhan, China. His research interests include machine learning, game theory, distributed optimization, and their applications in cloud/fog/mobile edge computing, green communication systems, wireless communication networks, and Internet-of-Things (IoT).



**Marwan Krunz** (S'93-M'95-SM'04-F'10) is the Kenneth VonBehren Endowed Professor in the Department of Electrical and Computer Engineering at the University of Arizona (UA). He is also the director of the Broadband Wireless Access and Applications Center (BWAC), a joint NSF/industry consortium that includes UA (lead site), Virginia Tech, University of Notre Dame, University of Mississippi, Auburn University, and Catholic University of America, as well as 20+ members from industry and national labs. Dr. Krunz received the Ph.D. degree in electrical engineering from Michigan State University in July 1995. He joined the University of Arizona in January 1997, after a brief postdoctoral stint at the University of Maryland, College Park. In 2010, he was a Visiting Chair of Excellence ("Catedra de Excelencia") at the University of Carlos III de Madrid (Spain). He held numerous other short-term research positions at the University Technology Sydney, Australia (2016), University of Paris V, INRIA-Sophia Antipolis, France, University of Paris VI, HP Labs, Palo Alto, and US West Advanced Technologies. He is the Editor-in-Chief for the *IEEE Transactions on Mobile Computing*, and he previously served on the editorial boards of numerous journals, including *IEEE/ACM Trans. on Networking*, *IEEE Trans. on Cognitive Communications and Networking*, and others.