Distributed Optimization for Computation Offloading in Edge Computing

Rongping Lin, Zhijie Zhou, Shan Luo, Yong Xiao, *Senior member*, *IEEE*, Xiong Wang, Sheng Wang, and Moshe Zukerman, *Life Fellow*, *IEEE*

***Abstract*—Edge computing is a promising technology that offers data analysis and computing for Internet of Things (IoT) services at the network edge. It has the potential to significantly reduce the latency and improve the reliability of IoT services by allowing computation workloads and local data generated by IoT devices to be offloaded to edge nodes. This paper aims to develop algorithms for efficient provision of both job assignment and resource allocation for edge computing networks. The main objective is to minimize the long-term average of the response time delay subject to constraints on computation resources and power consumption. We apply a drift-plus-penalty based Lyapunov optimization approach to convert the original problem into an upper bound optimization problem. We then relax the latter to a convex optimization problem. Finally, a distributed algorithm based on branch-and-bound approach is provided and the gap between the distributed algorithm solution and the optimal solution of the original problem is theoretically analyzed. Numerical results based on extensive experiments have demonstrated that our distributed algorithm can achieve the required performance of edge computing that supports IoT systems, under static traffic conditions as well as under dynamic environments with time-varying traffic.**

***Index Terms*—Edge computing, computation offloading, Lyapunov optimization, branch-and-bound**

I. Introduction

CLOUD computing provides efficient and on-demand services for end users, including computation, storage, software, and services by centrally sharing resources. However, cloud computing has a fundamental limitation related to the geographical distance between end users and datacenters that provide its services. With the fast growing demand for Internet of Things (IoT) services and the proliferation of intelligent devices, cloud data centers face challenges in meeting the service requirements of terminal devices (a.k.a. *IoT devices*) due to the limited availability and locations, especially for some latency-sensitive tasks requiring a low service response time, e.g. as low as a few *ms*. Meanwhile, it is predicted that the number of IoT devices (including all terminal devices connected to the internet) will reach 34.2 billion in 2025, and it is expected that this number will increase to 125 billion by 2030 [1], [2]. The network access from the massive scale of IoT devices introduces novel challenges for network operators. Specifically, the large number of IoT devices will generate a huge amount of data straining the limited network capacity and computation resources of cloud data center, which leads to a long delay.

Edge computing is a promising solution to complement cloud computing [3]–[7], where computation, storage and other resources can be distributively deployed at the edge nodes of the network, e.g. in base stations [8] and access points [9]. This new computing paradigm brings computation resources closer to end users avoiding sending data to the network, and therefore can support ultra-low delay and high computational applications [10]–[12]. In addition, because less data is uploaded to the cloud through the Internet, the security of the data can also be improved [13] under this paradigm. Furthermore, in edge computing, edge nodes provide distributed and limited computation resources to end users. This gives rise to the challenging problem of how to optimize resource allocation, especially, because more and more computation resources are required by end devices due to the emergence of computation demanding applications, such as 3D games and video editing. As a result, because of the increased demand, optimization of resource allocation will lead to, either offloading to the edge (or cloud), or efficient computation at the device level that will satisfy QoS requirements. Therefore, various solutions focusing on jointly addressing the above two challenges, namely, resource allocation and computation offloading, have been introduced in the content of edge computing (as shown in the next section). However, to the best of our knowledge, there are still no solutions that jointly optimize the long-term cost of resource allocation and workload offloading subject to computational resource and energy consumption constraints. Optimizing the long-term performance measures of the edge computing is important because such measures provide an aggregate of short-term (stable or unstable) correlated measures over a long period of time, and such aggregates have important economic and business implications, e.g. long-term cost and QoS measures. Such long-term considerations also introduce

This work was supported by grants from the National Natural Science Foundation of China (NSFC) (61871097, 61701079, 61671130), by Sichuan Science and Technology Program (2019YFG0086), by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China [CityU 11200318], and by National Key R&D Program of China (2019YFB1803304).

Corresponding author, Shan Luo (luoshan@uestc.edu.cn).

R. Lin, Z. Zhou, X. Wang and S. Wang are with the School of Information and Communication Engineering, University of Electronic Science and Technology of China (UESTC), China, 611731 (e-mail: linrp@uestc.edu.cn, Zhouzhijie868@foxmail.com, Wangxiong@uestc.edu.cn, wsh_keylab@uestc.edu.cn).

S. Luo is with the School of Astronautics and Aeronautic, UESTC, China, 611731 (e-mail: luoshan@uestc.edu.cn).

Y. Xiao is with the School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan, China, 430074 (email: xyong.2012@gmail.com).

M. Zukerman is with the Department of Electronic Engineering, City University of Hong Kong, Hong Kong SAR (e-mail: m.zu@cityu.edu.hk).

complexity to the optimization problem because of the need to consider the temporal fluctuation of the resource availability and workload arrival rate as well as the long-term constraints [14]. Such increase in complexity is significant as compared to only optimizing the instantaneous resource allocation and workload offloading under a stationary environment.

In this paper, we consider computation offloading and resource allocation in an edge computing network with the objective to minimize the long-term average response time delay. We represent the response time delay as the combination of data transmission delay, workload queueing and processing delay (the propagation delay is ignored here because of the short transmission distance between the device and the edge node). We focus on the long-term average response time delay minimization problem under two types of resource constraints: the long-term average computation resource and power consumption constraints. Such a perspective includes, for example, considerations for adaptive resource allocation in a long-term perspective to process peak traffic at low traffic states, which is motivated by the fact that a short period of performance degradation is tolerable for most practical networks as long as the long-term performance is guaranteed.

Meanwhile, it is known that optimizing the long-term average performance of a network with multiple continuous variables is a notoriously difficult problem [15]. Most existing solutions simplify the problem by approximating the continuous variables with discrete ones while ignoring the fact that discretizing some key continuous variables results in suboptimal and sometimes unconverged solutions [16]–[18]. To address the long-term average response time delay optimization problem, we propose a new Lyapunov optimization-based distributed algorithm. A virtual queue of Lyapunov optimization method has been proposed to convert the original continuous problem into subproblems at each time slot, and short-term fluctuations are considered in terms of the variation of virtual queues [14]. The time slots we considered include time periods of the order of minutes or even hours as long as it can capture traffic fluctuations. Accordingly, different time slots may have different arrival rates. The main contributions of this paper are as follows.

1) A mathematical model for the computation offloading problem with multiple end devices is provided, where the long-term average response time delay objective is optimized under long-term computation and power consumption constraints, and a perturbed technique-based on the Lyapunov optimization approach is proposed to convert the original problem into a deterministic upper bound problem.
2) A distributed optimization algorithm is proposed to solve the upper bound problem in each time slot, where each IoT device exchanges a limited amount of information with its associated edge node and decides the resource allocation and computation offloading by itself.
3) Both upper and lower bounds of the gap between the proposed distributed algorithm solution and the original optimal solution are derived. We also analyze the convergence of the algorithm, and provide a proof that the algorithm solution satisfies the long-term computation and power consumption constraints.
4) Results of extensive numerical experiments to evaluate the performance of our proposed algorithm are provided. They show that the proposed algorithm stabilizes the virtual queues, achieves the required performance, and is adjustable to the time-varying traffic.

The remainder of this paper is organized as follows. Section II discusses existing work related to computation offloading in edge computing. In Section III, we provide system model and problem formulation for the computation offloading problem in an edge computing network with long-term average objective and constraints. In Section IV, we derive an upper bound for the computation offloading problem. In Section V, a distributed algorithm is provided, and the gap between the solution of the algorithm and the optimal solution is derived. Section VI numerically evaluates and validates the new algorithm. Section VII concludes this paper.

II. Related Work

There are many potential scenarios and applications that can benefit from edge computing, such as Tactile Internet [19], IoT [20], internet of me [21], e-healthcare [22], autonomous driving [23], virtual/augmented reality (VR/AR) [24], caching and preprocessing [25]. For example, Cao et al. [26] proposed a distributed analysis system to monitor falls of stroke patients at real-time based on edge computing. Zao et al. [27] built an augmented brain computer interaction game, where heavy signal processing can be instantaneously processed by edge computing. Zhu et al. [28] proposed a method to improve end user web experiences, where edge computing reduces picture resolution in case of network congestion to reduce the response time.

The functionalities and performances of edge computing applications are highly dependent on the efficiencies of computation offloading and resource allocation, and various problems of the two issues (computation offloading is usually used to denote the combination of computation offloading and resource allocation) have been carried out to improve network performance considering resource limitations, delay and energy consumption [29]–[31]. For example, Xiao et al. [32] addressed the computation offloading problem aiming to improve the quality-of-experience (QoE) of end users considering the power efficiency, where a distributed algorithm was proposed. The same research group [33] proposed a stochastic overlapping coalition-formation game to achieve efficient network slicing among edge nodes, where computation offloading of various services with different quality-of-service (QoS) guarantees and energy harvesting were considered at edge nodes. However, both of these publications did not consider the computation resources and power limitations at end devices. Zhang et al. [34] proposed an online task assignment method in a simple scenario with only one energy harvesting end device, and the objective function was a weighted function of energy consumption and execution delay. Mao et al. [35] investigated the computation offloading problem with energy harvesting devices, and a dynamic algorithm was proposed to consider both execution latency and task failure. However, both [34] and [35] did not consider multiple end devices, which simplifies the resource allocation and computation offloading problems. Chen et al. [36] investigated the computation offloading problem in a scenario where multiple users and multi-channel wireless interferences were considered, and a distributed algorithm was proposed to decide computation offloading and selection of wireless channels based on the game theory. However, the computation resource sharing and allocation problems are ignored there for simplicity. Sardellitti et al. [37] investigated the computation offloading problem across multiple radio access points, and a heuristic algorithm was proposed for applications requiring high computation resources and low energy consumption. However, a given set of static requests was considered in the work, which can not be applied in dynamic traffic scenarios. Chen et al. [38] provided a mixed integer non-linear program with a delay minimization objective for the computation offloading problem, and converted the problem into two sub-problems (task placement and resource allocation). However, the work was based on software defined networks, where the central controller of the network incurs scalability problems when the network size becomes large. Zhu et al. [39] investigated user grouping and resource allocation problems in the hybrid of non-orthogonal multiple access and mobile edge computing, and the balance between energy consumption and delay was achieved. However, the work assumes all end devices offload computation tasks and only the wireless network resource allocation is considered. Wang et al. [40] investigated the energy and task allocation problems in wireless powered mobile edge computing networks, where the fluctuation of wireless channel states and task arrivals were considered. However, the work considered only one end user, and ignored computation resources consumptions in the system.

To consider the methodology for long-term average optimization problems, the Lyapunov optimization method has been applied to convert the problem into a new optimization problem at each time slot [14], where the latter has a significantly lower complexity than the former because there is no need to enumerate all system states. For example, Cui et al. [15] investigated delay-aware resource control problems in wireless systems, where the Lyapunov optimization was applied to solve the problems that had the objectives of throughput, delay and power consumption. He et al. [41] applied the Lyapunov optimization to design a buffer management strategy for the mobile video streaming, where bandwidth fluctuation and stochastic of wireless channels were considered. Qiu et al. [42] applied the Lyapunov optimization to design a transmission strategy in an energy harvesting wireless communication system, where the long-term average battery level and BER limitation were considered.

In this paper, we investigate the combination of computation offloading and resource allocation problems in edge computing networks, where multiple end users (IoT devices) generating variable traffic loads are considered. The objective is to minimize the long-term average response time delay, and the constraints are on long-term average usage of computation and power resources. By extending the Lyapunov optimization, we convert the original problem into an upper bound problem and design a distributed algorithm that is scalable. The notations used in the paper are defined in Table I.

TABLE I
SUMMARY OF USED NOTATIONS

Notation	Description
$\mathcal{N}$	Set of IoT devices
N	Number of IoT devices
w	Channel bandwidth
N_0	Noise power in the channel
$H_i(t)$	Channel gain from device i to the edge node at time slot t
$\lambda_i(t)$	Computation request arrival rate of device i at time slot t
$C_i(t)$	Transmission rate from device i to the edge node at slot t
L_i	Average data size of device i
D_i	Average computation requirement size of device i
σ_i	Standard deviation of computation requirement of device i
$DL_i(t)$	Uplink queuing and transmission time of device i at slot t
$DU_i(t)$	Unoffloaded workload processing delay of device i at slot t
$DO_i(t)$	Offloaded workload processing delay of device i at slot t
$R_i(t)$	Average response time delay of device i at time slot t
F_i	Computation capacity of device i
F_e	Computation capacity of the edge nodes
P_i	Power limitation of device i
v_i	Power consumption for computation resource of device i
T	Number of time slots
A	Virtual queue for the edge node computation constraint
B_i	Virtual queue for power constraint of device i
$\Delta\Theta(t)$	Lyapunov function drift
V	Weight factor of average response time delay to the drift
$\alpha_i(t)$	Offloading portion of device i at time slot t
$f_i(t)$	Computation allocated to device i at the edge node
$p_i(t)$	Transmission power of device i at time slot t
$t_1 \sim t_4$	Artificial variables

III. System Model and Problem Formulation

We consider a scenario, where a set of N homogeneous IoT devices are connected to an edge node for computation offloading as shown in Fig. 1, where the number of IoT devices is $N = |\mathcal{N}|$. This edge node can be, for example, a base station or a WiFi access point with limited computation resources. In this scenario, the computation workload of an IoT device is processed, either by itself, or by the edge node, or by both. If the computation workload is offloaded, uplink transmissions and computation resource allocations are considered. In this paper, the computation workload of the IoT device $i \in \mathcal{N}$ consists of a sequence of computation requests, which are assumed to arrive according to a Poisson process with arrival rate $\lambda_i(t)$ during time slot t, $t = 1, 2, 3, \ldots$. Although the assumption of Poisson arrivals is applicable to many scenarios including fog computing [32], [33], we acknowledge that the statistical characteristics of the arrival process of computation demands generated by a device are very much application dependent. Our assumption of Poisson arrivals will be adequate for applications when the arrivals are generated by many independent sources measured by the device. A currently relevant example of such arrivals will be health assessments of ill people accessing an office building. Information on people that are well can be processed by the device but more complicated situations where a person has fever, for example, may require more computations and are done at the edge. Because arrivals of ill individuals are events associated with a large population of independent individuals, each of which has a small probability to be ill, so their arrival process may be of a pure chance nature that follows a Poisson process. Here we use M/G/1 as a model for the computation workload processing, but for cases where the arrival process does not follow a Poisson process, this model (in particular, the Pollaczek Khinchine formula used here) may be replaced by other models (and formulae) to improve the accuracy of the results depending on the applications. In such a case, the overall procedure of this paper can still be used. Only the module representing the mean delay will be replaced by another formula or numerical procedure. One potential alternative to replace the M/G/1 model is the D/G/1 queueing model [43], [44] that models arrivals of deterministic periodic nature [45]. We assume here time dependent arrival rate to consider having more active time periods and downtime periods during the day. We adopt the widely used task model [36], [38] to describe computation requests, where the computation request of device i includes the average data size L_i and the average and standard deviation of computation requirement (D_i, σ_i) (e.g. number of uniformed CPUs). The offloaded computation workload at time slot t from the IoT device $i \in \mathcal{N}$ is denoted by $\alpha_i(t)\lambda_i(t)$, where the variable $\alpha_i(t) \in [0, 1]$, $t = 1, 2, 3, \ldots$, is the ratio of the offloaded computation workload to the total computation workload at time slot t, and the remainder $(1 - \alpha_i(t))\lambda_i(t)$ portion of computation workload is processed by the IoT device locally. It is noted that each entire computation request (computation task) is either processed at the edge node or at the end device. Accordingly, the value of $\alpha_i(t)$ decides the amount of offloaded workload $\alpha_i(t)\lambda_i(t)$ out of the entire workload $\lambda_i(t)$ to be processed at the edge node.

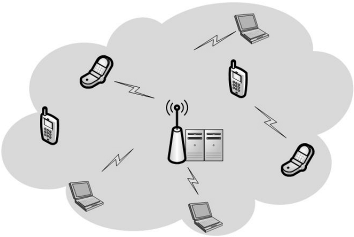


Fig. 1. An edge node with IoT devices.

A. Uplink Transmission

The data is transmitted from the IoT devices to the edge node through wireless uplink channels if the computation workload is offloaded to the edge node for processing. Denote $H_i(t)$ as the channel gain from device i to the edge node at time slot t, and w and N_0 as the channel bandwidth and the noise power in the channel, respectively. The uplink transmission rate is given by

$$C_i(t) = w \log\Big(1 + \frac{p_i(t)H_i(t)}{N_0}\Big). \tag{1}$$

In (1), w and N_0 are assumed to be constant, and $p_i(t)$ is the transmission power of the IoT device, which is a decision variable constrained by the power limitation of the IoT device. For simplicity, the interference among communication channels and the limitation on the number of channels in this paper are not considered. This simplifying assumption can be justified by the fact that new technologies, e.g. 5G, will make communication resources sufficiently large. Meanwhile, the reception delay of uplink transmission at the edge node side is also ignored, for ease of exposition because the edge node, like base station, usually has fast signal receiving and processing units, and in our scenarios, IoT devices usually generate computation tasks with low data but with high computation requirements. Considering the interference will add significant complexity to the problem. However, to consider the reception delay, a constant delay can be added to the uplink transmission time which will not increase the complexity of the problem.

Consider the $\alpha_i(t)$ portion of the total computation workload in device i offloaded to the edge node through the uplink with transmission rate $C_i(t)$, and assume that the arrivals during a time slot follow a Poisson process, and uplink transmission times are exponentially distributed. We also assume that the time slots are sufficiently long so that queueing steady-state conditions hold. Then, considering that in the uplink the variance of the packet length is not very large, and assuming an M/M/1 queueing system in the uplink transmission, by the M/M/1 mean delay formula [46], the uplink delay (including queuing and transmission time) of a computation request is given by

$$DL_i(t) = \frac{1}{\frac{C_i(t)}{L_i} - \alpha_i(t)\lambda_i(t)}. \tag{2}$$

Without loss of generality, the specific packet size is not considered. We only consider the total data to be transmitted to obtain the service rate, but we do not consider the length of individual packets. As long as the total data to be transmitted does not change, the size of individual packets does not affect the optimal decisions and the optimal solution.

B. Computation Workload Processing

An IoT device usually has a limited computation resource, and runs a simple operation system with simple resource management strategies. Accordingly, we assume that the full computation capacity F_i of the IoT device i is applied in the task processing without sophisticated resource slicing strategies. To provide this computation capacity, the power is assumed to be a constant value ν_i in the IoT device i. M/M/1 queueing system has been assumed in the uplink transmission since most data has a limited variance. However, we consider M/G/1 queueing system for the task processing because data processing requests may have large variance because they may be different in nature with significantly different processing times, e.g. Firewall vs. image rendering. For the offloaded computation workloads, two tandem queues (the first M/M/1 queue for uplink transmission and the second M/G/1 queue for computation processing at edge node) are traversed, and the input of the second queue is the output of the first queue. As it is well known that the output process of M/M/1 is Poisson which justifies the assumption of Poisson arrivals to the second queue (modeled by M/G/1). The unoffloaded computation workload is processed locally in the IoT device, then from the mean delay formula of M/G/1 queueing system [46], we have the processing delay of the unoffloaded computation workload $(1-\alpha_i(t))\lambda_i(t)$ is given by

$$DU_i(t) = \frac{(1-\alpha_i(t))\lambda_i(t)(\sigma_i^2 + D_i^2)}{2[F_i^2 - (1-\alpha_i(t))\lambda_i(t)D_iF_i]} + \frac{D_i}{F_i}. \quad (3)$$

In contrast, we assume that the edge node can efficiently allocate computation resources for each computation workload. Hence, from the mean delay formula of M/G/1 queueing system, the processing delay of the offloaded computation workload $\alpha_i(t)\lambda_i(t)$ is

$$DO_i(t) = \frac{\alpha_i(t)\lambda_i(t)(\sigma_i^2 + D_i^2)}{2(f_i^2(t) - \alpha_i(t)\lambda_i(t)D_if_i(t))} + \frac{D_i}{f_i(t)}, \quad (4)$$

where $f_i(t)$ is the computation resource allocated to the IoT device i for the workload processing.

C. Problem Formulation

As mentioned in Section I, the transmission distance between the IoT device and the edge node is relatively short; hence, the propagation delay at the wireless transmission is ignored. In the downlink transmission, the transmission rate is usually sufficiently high, so that the delay (including transmission delay, propagation delay and queueing delay) is negligible. Then, the overall average response time delay is given as

$$\begin{aligned} R_i(t) &= \alpha_i(t)(DL_i(t) + DO_i(t)) + (1-\alpha_i(t))DU_i(t) \\ &= \frac{\alpha_i(t)}{\frac{C_i(t)}{L_i} - \alpha_i(t)\lambda_i(t)} + \frac{\alpha_i^2(t)\lambda_i(t)(\sigma_i^2 + D_i^2)}{2(f_i^2(t) - \alpha_i(t)\lambda_i(t)D_if_i(t))} \\ &+ \frac{\alpha_i(t)D_i}{f_i(t)} + \frac{(1-\alpha_i(t))^2\lambda_i(t)(\sigma_i^2 + D_i^2)}{2[F_i^2 - (1-\alpha_i(t))\lambda_i(t)D_iF_i]} + \frac{(1-\alpha_i(t))D_i}{F_i}. \end{aligned} \quad (5)$$

According to (5), if there is no computation workload offloaded from device i at time slot t, the entire computation workload is processed locally at the IoT devices, which gives the entire delay to be equal to

$$\frac{\lambda_i(t)(\sigma_i^2 + D_i^2)}{2(F_i^2 - \lambda_i(t)D_iF_i)} + \frac{D_i}{F_i},$$

and if all computation workload is offloaded to the edge node, the delay becomes

$$\frac{1}{\frac{C_i(t)}{L_i} - \lambda_i(t)} + \frac{\lambda_i(t)(\sigma_i^2 + D_i^2)}{2(f_i^2(t) - \lambda_i(t)D_if_i(t))} + \frac{D_i}{f_i(t)}.$$

Here, the optimization problem of minimizing long-term end-to-end response time delay for the IoT devices is investigated, and long-term average constraints on computation and power resources are considered. The formulation of the problem is as follows

$$\textbf{P1:} \quad min \quad \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \sum_{i=1}^{N} R_i(t), \quad (6)$$

$$s.t. \quad \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \sum_{i=1}^{N} f_i(t) \le F_e, \quad (7)$$

$$\lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} (p_i(t) + \nu_i I_{\{\alpha_i(t)<1\}}) \le P_i \quad \forall i \in \mathcal{N}, \quad (8)$$

$$0 \le \alpha_i(t) \le 1 \quad \forall i \in \mathcal{N}, \quad (9)$$

$$0 \le p_i(t) \le P_i \quad \forall i \in \mathcal{N}, \quad (10)$$

$$0 \le f_i(t) \le F_e \quad \forall i \in \mathcal{N}. \quad (11)$$

The objective is to minimize the total long-term average of response time delay of all IoT devices. The decision variables are: offloading portion $\alpha_i(t)$, computation resource allocation $f_i(t)$, and transmission power $p_i(t)$ at each time slot, where $i \in \mathcal{N}$. In Problem **P1**, decision variables are based on the arrival rate of computation requests during the time slot as (5), if arrival rate changes, new decision variables should be made for the new arrival rate. In this paper, we follow Neely's framework [14] and assume that all time slots are of equal length.

The constraints in Problem **P1** limit the resources allocated to long-term average computation (7) and to long-term average power (8). In (7), the long-term average computation resource allocation is limited by the edge node computation capacity F_e, which implies that more than resource capacity can be allocated at a time slot.It also means that the workload peak can be buffered to be processed in a later time slot. Note that this long-term computation constraint allows the summation of computation consumptions to be larger than the computation capacity at an instant time slot, but from long-term perspective, the average computation consumption must be less than the computation capacity for feasibility. Similarly, with power amplifier and battery life cycle considerations, each IoT device has a long-term power limitation P_i, $i \in \mathcal{N}$ at (8), where the power consumption consists of two parts: from uplink data transmission and computation processing. If there is no computation workload offloading, the transmission power $p_i(t)$ should be zero; otherwise, $p_i(t)$ will decide the transmission rate of the uplink according to (1). If IoT device i has a portion or all of its computation workload processed locally, i.e. $\alpha_i(t) < 1$, then, the indicator function $I_{\{\alpha_i(t)<1\}}$ takes the value of 1. This makes the IoT device consume a constant power ν_i to provide its computation resource F_i, and ν_i is given to be no larger than the power limitation P_i.

We note that our long-term focus in this paper implies that our optimization problem considers the aggregation of system behavior in various network states over a long period of time. These network states include: high traffic load states and low traffic load states. This is different from an optimization problem over a limited time interval, because the former takes account of correlation between measures such as queue length of tasks at the buffer at consecutive time slots. An optimization that focuses only on a limited time interval, e.g. a single time slot, such consideration of correlated measures cannot be taken account of. The global optimization over a longer time horizon is able to consider long-term effects of processes and aggregation of all possible system states in the system to achieve overall better performance. For example, optimal decisions on buffering or offloading traffic during peak traffic on certain devices will have long-term implications that cannot be considered if the optimization is done over a limited time period.

IV. Lyapunov Optimization Based Problem Upper Bound Analysis

To solve the optimization problem **P1** with a long-term objective function and constraints, we will use the Lyapunov optimization to convert the original optimization problem into a new optimization problem, and then efficiently and distributively solve the new one instead.

A. Virtual queues

In Lyapunov optimization, the satisfaction of a long-term average constraint is equal to the rate stability of a virtual queue. Specifically, in this paper, a virtual queue is provided to replace the computation resource constraint (7) of the edge node, and $A(t)$ denotes the stochastic process of the length of the virtual queue A at time t. The value of $A(t)$ indicates how much computation resources are allocated beyond the capacity at t. Similarly, to replace power constraint (8), another virtual queue is provided for the IoT device i, $\forall i \in \mathcal{N}$. $B_i(t)$ denotes the stochastic process of this virtual queue and is equal to the length of queue at t. The value of $B_i(t)$ indicates how much the computation power usage is beyond the capacity at t. For convenience, we use the notations A and B_i, $\forall i \in \mathcal{N}$ to name virtual queues, i.e. virtual queue A and virtual queue B_i, $\forall i \in \mathcal{N}$, and $A(t)$ and $B_i(t)$ are the size of virtual queue A and virtual B_i at time t, respectively. The updates of virtual queues are as follows

$$A(t+1) = \max\Big[A(t) + \sum_{i=1}^{N} f_i(t) - F_e, 0\Big], \qquad (12)$$

$$B_i(t{+}1) = \max\big[B_i(t){+}p_i(t){+}\nu_i I_{\{\alpha_i(t)<1\}}{-}P_i, 0\big] \quad \forall i \in \mathcal{N}. \qquad (13)$$

From (12), we can see that if the total amount of allocated computation resources exceeds the resources capacity, the virtual queue length $A(t)$ will increase, otherwise, the virtual queue length will decrease. Similar behaviors can also be observed in virtual queues B_i, $\forall i \in \mathcal{N}$ from (13) with respect to the power consumptions and the power budget.

Lemma 1: If virtual queues A and B_i, $\forall i \in \mathcal{N}$ are rate stable, i.e., $\lim\limits_{T\to\infty} \frac{A(T)}{T} = 0$ and $\lim\limits_{T\to\infty} \frac{B_i(T)}{T} = 0, \forall i \in \mathcal{N}$, then the long-term constraints (7) and (8) are satisfied.

Proof: From (12), we have

$$A(t{+}1) = \begin{cases} A(t) + \sum_{i=1}^{N} f_i(t) - F_e, \\ \qquad\qquad\qquad \text{if } A(t) + \sum_{i=1}^{N} f_i(t) - F_e \geq 0 \\ 0, \qquad\qquad\quad \text{if } A(t) + \sum_{i=1}^{N} f_i(t) - F_e < 0. \end{cases}$$

Then, we obtain

$$\begin{aligned} &A(t+1) - A(t) \\ &= \begin{cases} \sum_{i=1}^{N} f_i(t) - F_e, & \text{if } A(t) + \sum_{i=1}^{N} f_i(t) - F_e \geq 0 \\ -A(t), & \text{if } A(t) + \sum_{i=1}^{N} f_i(t) - F_e < 0, \end{cases} \\ &= \max\left\{\sum_{i=1}^{N} f_i(t) - F_e, -A(t)\right\} \geq \sum_{i=1}^{N} f_i(t) - F_e. \end{aligned}$$

For $t = 0, 1, 2, ..., T-1$, summing up both sides, we obtain

$$\lim_{T\to\infty} \frac{A(T) - A(0)}{T} \geq \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \sum_{i=1}^{N} f_i(t) - F_e.$$

Assuming $A(0) = 0$, and if the virtual queue A is rate stable, i.e. $\lim\limits_{T\to\infty} \frac{A(T)}{T} = 0$, we have

$$\lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \sum_{i=1}^{N} f_i(t) \leq F_e.$$

Similarly, if the virtual queues B_i, $i \in \mathcal{N}$ are rate stable, we can obtain

$$\lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} (p_i(t) + \nu_i I_{\{\alpha_i(t)<1\}}) \leq P_i \quad \forall i \in \mathcal{N}.$$

This completes the proof. ■

Following Lemma 1, the optimization problem **P1** is equivalently converted to become the following problem.

$$\textbf{P2:} \quad min \quad \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \sum_{i=1}^{N} R_i(t), \qquad (14)$$

$$s.t. \quad A(t) \text{ is rate stable}, \qquad (15)$$

$$B_i(t) \text{ is rate stable}, \quad \forall i \in \mathcal{N}, \qquad (16)$$

$$(9), (10), (11), \qquad (17)$$

where (15) and (16) are equality constraints of (7) and (8) based on the virtual queues defined in (12) and (13), respectively.

B. Drift-plus-Penalty

We use the drift-plus-penalty [14] of the Lyapunov optimization to approximately solve **P2** with an upper bound. According to the Lyapunov optimization theory, we first define the vector $\Theta(t)$ as $\Theta(t) = [A(t), B_1(t), B_2(t), ..., B_N(t)]$. Then, the Lyapunov function can be written as $L(\Theta(t)) = \frac{1}{2}(A(t)^2 + \sum_{i=1}^{N} B_i(t)^2)$. The drift $\Delta\Theta(t)$ can be obtained as

$$\Delta\Theta(t) = \mathbb{E}\{L(\Theta(t+1)) - L(\Theta(t))|\Theta(t)\}.$$

The original problem with long-term average objective and constraints can be approximately converted into a problem with the drift-plus-penalty as follows

$$\begin{aligned} \textbf{P3:} \quad & min \quad \Delta\Theta(t) + V\mathbb{E}\Big\{\sum_{i=1}^{N} R_i(t)|\Theta(t)\Big\} \\ & s.t. \quad (9), (10), (11), \end{aligned} \tag{18}$$

where minimizing the drift $\Delta\Theta(t)$ enforces rate stability of the virtual queues A and B_i, $i \in \mathcal{N}$, and the parameter $V \geq 0$ represents the penalty weight of the objective function to the drift.

At the following, we will provide an upper bound for $\Delta\Theta(t)$. Next, we solve the optimization problem **P3** with its upper bound. We start by finding an upper bound of $A(t+1)^2 - A(t)^2$ and of $\sum_{i=1}^{N}[B_i(t+1)^2 - B_i(t)^2]$.

$$\begin{aligned} & A(t+1)^2 - A(t)^2 \\ & = \Big\{\max\Big[A(t) + \sum_{i=1}^{N} f_i(t) - F_e, 0\Big]\Big\}^2 - A(t)^2 \\ & \leq \Big[A(t) + \sum_{i=1}^{N} f_i(t) - F_e\Big]^2 - A(t)^2 \\ & = 2A(t)\Big(\sum_{i=1}^{N} f_i(t) - F_e\Big) + \Big[\sum_{i=1}^{N} f_i(t) - F_e\Big]^2 \\ & \leq 2A(t)\sum_{i=1}^{N} f_i(t) + (\sum_{i=1}^{N} f_i(t))^2 - 2F_e\sum_{i=1}^{N} f_i(t) + F_e^2 \\ & \leq (N^2+1)F_e^2 + 2(A(t) - F_e)\sum_{i=1}^{N} f_i(t) \\ & = G_1 + 2(A(t) - F_e)\sum_{i=1}^{N} f_i(t), \end{aligned}$$

where G_1 is a constant which equals to $(N^2+1)F_e^2$, and the first inequality is due to $\{max(a, 0)\}^2 \leq a^2$, the third inequality is due to $0 \leq f_i(t) \leq F_e$. Similarly, an upper bound of $\sum_{i=1}^{N}[B_i(t+1) - B_i(t)]$ can be obtained as follows

$$\begin{aligned} & B_i(t+1)^2 - B_i(t)^2 \\ & = \big\{\max[B_i(t) + p_i(t) + \nu_i I_{\{\alpha_i(t)<1\}} - P_i, 0]\big\}^2 - B_i(t)^2 \\ & \leq 2B_i(t)(p_i(t) + \nu_i I_{\{\alpha_i(t)<1\}} - P_i) \\ & \quad + (p_i(t) + \nu_i I_{\{\alpha_i(t)<1\}} - P_i)^2 \leq 2B_i(t)(p_i(t) + \nu_i - P_i) \\ & \quad + p_i(t)^2 + 2p_i(t)(\nu_i I_{\{\alpha_i(t)<1\}} - P_i) + (\nu_i I_{\{\alpha_i(t)<1\}} - P_i)^2 \\ & \leq 2B_i(t)(p_i(t) + \nu_i - P_i) + p_i(t)^2 + 2p_i(t)(\nu_i - P_i) \\ & \qquad\qquad + \nu_i^2 + P_i^2 \\ & = P_i^2 + \nu_i^2 + p_i(t)^2 + 2p_i(t)(B_i(t) + \nu_i - P_i) \\ & \qquad\qquad + 2B_i(t)(\nu_i - P_i), \end{aligned}$$

and

$$\begin{aligned} & \sum_{i=1}^{N}[B_i(t+1)^2 - B_i(t)^2] = \sum_{i=1}^{N}\big\{P_i^2 + \nu_i^2 + p_i(t)^2 \\ & \qquad + 2p_i(t)(B_i(t) + \nu_i - P_i) + 2B_i(t)(\nu_i - P_i)\big\} \\ & = G_2 + 2\sum_{i=1}^{N}\Big\{\frac{1}{2}p_i(t)^2 + p_i(t)(B_i(t) + \nu_i - P_i) \\ & \qquad + B_i(t)(\nu_i - P_i)\Big\}, \end{aligned}$$

where G_2 is a constant which is equal to $\sum_{i=1}^{N}(P_i^2 + \nu_i^2)$. Accordingly, the upper bound of the objective function (18) is expressed as

$$\begin{aligned} & \Delta\Theta(t) + V\mathbb{E}\Big\{\sum_{i=1}^{N} R_i(t)|\Theta(t)\Big\} \\ & = \mathbb{E}\big\{L(\Theta(t+1)) - L(\Theta(t))|\Theta(t)\big\} + V\mathbb{E}\Big\{\sum_{i=1}^{N} R_i(t)|\Theta(t)\Big\} \\ & = \frac{1}{2}\mathbb{E}\Big\{A(t+1)^2 - A(t)^2 + \sum_{i=1}^{N}[B_i(t+1)^2 - B_i(t)^2]|\Theta(t)\Big\} \\ & \qquad + V\mathbb{E}\Big\{\sum_{i=1}^{N} R_i(t)|\Theta(t)\Big\} \\ & \leq \frac{G_1 + G_2}{2} + \mathbb{E}\Big\{(A(t) - F_e)\sum_{i=1}^{N} f_i(t)|\Theta(t)\Big\} \\ & \quad + \mathbb{E}\Big\{\sum_{i=1}^{N}\Big(\frac{1}{2}p_i(t)^2 + p_i(t)(B_i(t) + \nu_i - P_i) \\ & \qquad + B_i(t)(\nu_i - P_i)\Big)|\Theta(t)\Big\} + V\mathbb{E}\Big\{\sum_{i=1}^{N} R_i(t)|\Theta(t)\Big\}. \end{aligned} \tag{19}$$

Based on the idea of opportunistically minimizing an expectation [14], we convert problem **P3** to the upper bound problem **P4** with the drift-plus-penalty as follows. In this way, the original problem **P1** can be approximated as problem **P4**, and we will show the relationship of the solutions between these two different problems.

$$\begin{aligned} \textbf{P4:} \quad & min \quad V\sum_{i=1}^{N} R_i(t) + (A(t) - F_e)\sum_{i=1}^{N} f_i(t) \\ & + \sum_{i=1}^{N}\Big(\frac{1}{2}p_i(t)^2 + p_i(t)(B_i(t) + \nu_i - P_i) + B_i(t)(\nu_i - P_i)\Big) \\ & \qquad s.t. \quad (9), (10), (11). \end{aligned} \tag{20}$$

Lemma 2: If problem **P4** is feasible and the objective value C is obtained, the virtual queues A and B_i, $\forall i \in \mathcal{N}$ are rate stable, i.e., $\lim_{T\to\infty}\frac{A(T)}{T} = 0$ and $\lim_{T\to\infty}\frac{B_i(T)}{T} = 0, \forall i \in \mathcal{N}$.

Proof: From (19) and the objective function of problem **P4**, we have

$$L(\Theta(t+1)) - L(\Theta(t)) + V\sum_{i=1}^{N} R_i(t) \leq C,$$

then, we obtain

$$L(\Theta(t+1)) - L(\Theta(t)) \leq C - V\sum_{i=1}^{N} R_i(t) \leq C,$$

where $V\sum_{i=1}^{N} R_i(t) \geq 0$. Given $t = 0, 1, 2, ..., T-1$, we do the summations at right-hand side and at left-hand side of the above inequalities, and we obtain $L(\Theta(T)) - L(\Theta(0)) \leq TC$. Considering $L(\Theta(0)) = 0$, we have $\frac{1}{2}(A(T)^2 + \sum_{i=1}^{N} B_i(T)^2) \leq TC$, and given $A(T) \geq 0$, we have $A(T) \leq \sqrt{2TC}$, and

$$\lim_{T\to\infty} \frac{A(T)}{T} \leq \lim_{T\to\infty} \frac{\sqrt{2TC}}{T} = 0.$$

Finally, we have

$$\lim_{T\to\infty} \frac{A(T)}{T} = 0.$$

Similarly, we can obtain $\lim\limits_{T\to\infty} \frac{B_i(T)}{T} = 0 \quad \forall i \in \mathcal{N}$.

Virtual queues A and B_i, $\forall i \in \mathcal{N}$ are rate stable. This completes the proof. ■

Lemma 3: If problem **P4** is feasible, then the long-term constraints (7) and (8) are satisfied.

Proof: From Lemma 2, the virtual queues are rate stable if problem **P4** is feasible, and from Lemma 1, the long-term constraints (7) and (8) are satisfied if the virtual queues are rate stable. This completes the proof. ■

C. Approximation Analysis

In this subsection, we provide a mathematical proof for the upper bound of the difference between the solutions derived by the approximation **P4** and by problem **P1**.

Theorem 1: The optimal long-term response time delay obtained by problem **P4** is limited by an upper bound that is the optimal value R^* of the original problem **P1** plus a constant ε, which is obtained as follows

$$\lim_{T\to\infty} \frac{1}{T}\sum_{t=0}^{T-1}\sum_{i=1}^{N} R_i'(t) \leq R^* + \varepsilon, \tag{21}$$

where $R_i'(t)$, $i \in \mathcal{N}$, $t = 1, 2, ..., T-1$ is the delay of IoT devices i in time slot t that is derived by problem **P4**, and

$$\begin{aligned}\varepsilon =& \frac{(N^2+1)F_e^2 + 2NA_{max}F_e}{2V} \\ &+ \frac{1}{2V}\sum_{i=1}^{N}(2P_i^2 + \nu_i^2 + 2B_{max}P_i + 2P_i\nu_i + 2B_{max}\nu_i).\end{aligned}$$

Proof: After solving problem **P4**, we obtain the optimal solution (expressed by $R_i'(t)$, $f_i'(t)$ and $p_i'(t)$) that minimizes the objective function value of **P4**. From (19), we obtain

$$\begin{aligned}&\Delta\Theta'(t) + V\mathbb{E}\Big\{\sum_{i=1}^{N} R_i'(t)|\Theta(t)\Big\} \\ &\leq \frac{G_1+G_2}{2} + V\sum_{i=1}^{N} R_i'(t) + (A(t)-F_e)\sum_{i=1}^{N} f_i'(t) \\ &+ \sum_{i=1}^{N}\Big(\frac{1}{2}p_i'(t)^2 + p_i'(t)(B_i(t)+\nu_i-P_i) + B_i(t)(\nu_i-P_i)\Big) \\ &\leq \frac{G_1+G_2}{2} + VR^* + (A(t)-F_e)\sum_{i=1}^{N} f_i(t) \\ &+ \sum_{i=1}^{N}\Big(\frac{1}{2}p_i(t)^2 + p_i(t)(B_i(t)+\nu_i-P_i) + B_i(t)(\nu_i-P_i)\Big) \\ &\leq \frac{G_1+G_2}{2} + VR^* + NA_{max}F_e \\ &\quad + \sum_{i=1}^{N}\Big(\frac{1}{2}P_i^2 + B_{max}P_i + P_i\nu_i + B_{max}\nu_i\Big),\end{aligned}$$

where the second inequality is due to the fact that the minimal of the left-hand side (the optimal objective function value of **P4**) must be no larger than any solution of the right-hand side. The formulas can be rearranged as follows

$$\begin{aligned}\mathbb{E}\Big\{\sum_{i=1}^{N} R_i'(t)|\Theta(t)\Big\} &\leq \frac{G_1+G_2}{2V} + R^* + \frac{NA_{max}F_e}{V} \\ &+ \frac{1}{V}\sum_{i=1}^{N}\Big(\frac{1}{2}P_i^2 + B_{max}P_i + P_i\nu_i + B_{max}\nu_i\Big) - \frac{1}{V}\Delta\Theta'(t).\end{aligned}$$

By taking the long-term average from time slot 0 to $T-1$ on both sides of the inequality, and let T go to infinity, we obtain

$$
\begin{aligned}
&\lim_{T\to\infty}\frac{1}{T}\sum_{t=0}^{T-1}\sum_{i=1}^{N}R_i'(t)\\
&\leq R^* - \lim_{T\to\infty}\frac{1}{T}\sum_{t=0}^{T-1}\frac{\Delta\Theta'(t)}{V} + \frac{G_1+G_2}{2V} + \frac{NA_{max}F_e}{V}\\
&\quad + \frac{1}{V}\sum_{i=1}^{N}\Big(\frac{1}{2}P_i^2 + B_{max}P_i + P_i\nu_i + B_{max}\nu_i\Big)\\
&= R^* - \lim_{T\to\infty}\frac{L(\Theta'(T-1)) - L(\Theta'(0))}{VT} + \frac{G_1+G_2}{2V}\\
&\quad + \frac{NA_{max}F_e}{V} + \frac{1}{V}\sum_{i=1}^{N}\Big(\frac{1}{2}P_i^2 + B_{max}P_i + P_i\nu_i + B_{max}\nu_i\Big)\\
&= R^* + \frac{G_1+G_2}{2V} + \frac{NA_{max}F_e}{V}\\
&\quad + \frac{1}{V}\sum_{i=1}^{N}\Big(\frac{1}{2}P_i^2 + B_{max}P_i + P_i\nu_i + B_{max}\nu_i\Big)\\
&= R^* + \frac{(N^2+1)F_e^2 + \sum_{i=1}^{N}(P_i^2+\nu_i^2)}{2V} + \frac{NA_{max}F_e}{V}\\
&\quad + \frac{1}{V}\sum_{i=1}^{N}\Big(\frac{1}{2}P_i^2 + B_{max}P_i + P_i\nu_i + B_{max}\nu_i\Big)\\
&= R^* + \frac{(N^2+1)F_e^2 + 2NA_{max}F_e}{2V}\\
&\quad + \frac{\sum_{i=1}^{N}(2P_i^2 + \nu_i^2 + 2B_{max}P_i + 2P_i\nu_i + 2B_{max}\nu_i)}{2V},
\end{aligned}
$$

where the third equality is due to the fact that the virtual queues are rate stable. Otherwise, we cannot obtain the optimal solution of problem **P4**. This completes the proof. ■

V. Distributed Algorithm Design

In the above section, the original long-term optimization problem has been converted into the upper bound problem **P4**. However, problem **P4** is not a convex optimization problem due to the non-convexity of the objective function. Meanwhile, many IoT devices may connect to the edge node, which introduces a large number of variables, making problem **P4** intractable. In this section, we provide a distributed algorithm to solve problem **P4**.

In the objective function of problem **P4**, there are three summation terms for N IoT devices, and we can equally convert the objective function into N objective functions for the N IoT devices, and each has the following objective function.

$$
\begin{aligned}
min \quad &\frac{1}{2}p_i(t)^2 + (B_i(t)+\nu_i-P_i)p_i(t) + (A(t)-F_e)f_i(t)\\
&+ B_i(t)(\nu_i - P_i) + VR_i(t).
\end{aligned}
\tag{22}
$$

Specifically, each IoT device has the optimization problem after substituting $R_i(t)$ with (5) as follows. The acronym DIP stands for distributed problem in this paper.

$$
\begin{aligned}
\textbf{DIP1: } &min\ \frac{1}{2}p_i(t)^2 + (B_i(t)+\nu_i-P_i)p_i(t) + B_i(t)(\nu_i-P_i)\\
&+ (A(t)-F_e)f_i(t) + \frac{V\alpha_i(t)}{\frac{w}{L_i}\log\big(1+\frac{p_i(t)H_i(t)}{N_0}\big) - \alpha_i(t)\lambda_i(t)}\\
&+ \frac{V\alpha_i^2(t)\lambda_i(t)(\sigma_i^2+D_i^2)}{2\big(f_i^2(t) - \alpha_i(t)\lambda_i(t)D_if_i(t)\big)} + \frac{V\alpha_i(t)D_i}{f_i(t)}\\
&+ \frac{V(1-\alpha_i(t))^2\lambda_i(t)(\sigma_i^2+D_i^2)}{2\big[F_i^2 - (1-\alpha_i(t))\lambda_i(t)D_iF_i\big]} + \frac{V(1-\alpha_i(t))D_i}{F_i},
\end{aligned}
\tag{23}
$$

$$
s.t. \quad 0 \leq \alpha_i(t) \leq 1, \tag{24}
$$

$$
0 \leq p_i(t) \leq P_i, \tag{25}
$$

$$
0 \leq f_i(t) \leq F_e, \tag{26}
$$

$$
\frac{w}{L_i}\log\Big(1+\frac{p_i(t)H_i(t)}{N_0}\Big) - \alpha_i(t)\lambda_i(t) > 0, \tag{27}
$$

$$
f_i(t) - \alpha_i(t)\lambda_i(t)D_i > 0, \tag{28}
$$

$$
F_i - (1-\alpha_i(t))\lambda_i(t)D_i > 0. \tag{29}
$$

There are three more constraints ((27) to (29)) added to ensure that the denominators in (5) are positive, where (28) and (29) are the denominators divided by $f_i(t)$ and F_i (both are positive), respectively. Obviously, the objective function is a non-convex function, so the problem is a non-convex optimization problem. For this problem, we use fractional programming [47] to convert the problem **DIP1** into an equivalent optimization problem. First, we introduce four artificial variables t_1, t_2, t_3, t_4 as follows

$$
\frac{V\alpha_i(t)}{\frac{w}{L_i}\log\big(1+\frac{p_i(t)H_i(t)}{N_0}\big) - \alpha_i(t)\lambda_i(t)} \leq t_1,
$$

$$
\frac{V\alpha_i^2(t)\lambda_i(t)(\sigma_i^2+D_i^2)}{2\big(f_i^2(t) - \alpha_i(t)\lambda_i(t)D_if_i(t)\big)} \leq t_2,
$$

$$
\frac{V\alpha_i(t)D_i}{f_i(t)} \leq t_3,
$$

$$
\frac{V(1-\alpha_i(t))^2\lambda_i(t)(\sigma_i^2+D_i^2)}{2\big[F_i^2 - (1-\alpha_i(t))\lambda_i(t)D_iF_i\big]} \leq t_4,
$$

then we obtain an equivalent optimization problem as follows

$$
\begin{aligned}
\textbf{DIP2:} \quad min \quad &\frac{1}{2}p_i(t)^2 + (B_i(t)+\nu_i-P_i)p_i(t)\\
&+ B_i(t)(\nu_i-P_i) + (A(t)-F_e)f_i(t) + \sum_{j=1}^{4}t_j,
\end{aligned}
\tag{30}
$$

$$
s.t.\ V\alpha_i(t) - t_1\Big[\frac{w\log\big(1+\frac{p_i(t)H_i(t)}{N_0}\big)}{L_i} - \alpha_i(t)\lambda_i(t)\Big] \leq 0,
\tag{31}
$$

$$V\alpha_i^2(t)\lambda_i(t)(\sigma_i^2+D_i^2)-2t_2\big[f_i^2(t)-\alpha_i(t)\lambda_i(t)D_if_i(t)\big]\le 0, \tag{32}$$

$$V\alpha_i(t)D_i-t_3f_i(t)\le 0, \tag{33}$$

$$\begin{aligned}&V(1-\alpha_i(t))^2\lambda_i(t)(\sigma_i^2+D_i^2)\\&\qquad-2t_4\big[F_i^2-(1-\alpha_i(t))\lambda_i(t)D_iF_i\big]\le 0,\end{aligned} \tag{34}$$

$$(24)-(29). \tag{35}$$

Problem **DIP2** is also a non-convex optimization problem because of the non-convex constraints. We convert this problem into a lower bound convex problem which can be solved efficiently. Constraints (31)-(34) are relaxed as shown in (36)-(39), which make the terms of t_i $i=1,2,3,4$ linear.

$$V\alpha_i(t)-t_1\Big[\frac{w\log(1+\frac{p_i^u(t)H_i(t)}{N_0})}{L_i}-\alpha_i^l(t)\lambda_i(t)\Big]\le 0, \tag{36}$$

$$V\alpha_i^2(t)\lambda_i(t)(\sigma_i^2+D_i^2)-2t_2\big[\hat{f}_i^2(t)-\alpha_i^l(t)\lambda_i(t)D_i\hat{f}_i(t)\big]\le 0, \tag{37}$$

$$V\alpha_i(t)D_i-t_3f_i^u(t)\le 0, \tag{38}$$

$$\begin{aligned}&V(1-\alpha_i(t))^2\lambda_i(t)(\sigma_i^2+D_i^2)\\&\qquad-2t_4\big[F_i^2-(1-\alpha_i^u(t))\lambda_i(t)D_iF_i\big]\le 0,\end{aligned} \tag{39}$$

where $\alpha_i^l(t)$ and $\alpha_i^u(t)$ are the minimal and the maximal values of $\alpha_i(t)$, respectively, $f_i^l(t)$ and $f_i^u(t)$ are the minimal and the maximal values of $f_i(t)$, respectively, and $p_i^l(t)$ and $p_i^u(t)$ are the minimal and the maximal values of $p_i(t)$, respectively. It is noted that in (32), the derivation of the term $f_i^2(t)-\alpha_i(t)\lambda_i(t)D_if_i(t)$ to $\alpha_i(t)$ is $-\lambda_i(t)D_if_i(t)$, which is non-positive. Then, the term has the largest value when $\alpha_i(t)$ is equal to $\alpha_i^l(t)$. After $\alpha_i(t)$ is set to $\alpha_i^l(t)$, the term is convex function, and the largest value can be chosen from $f_i(t)=f_i^l(t)$ and $f_i(t)=f_i^u(t)$, and $\hat{f}_i(t)=\arg\max\{f_i^2(t)-\alpha_i^l(t)\lambda_i(t)D_if_i(t)|f_i(t)\in\{f_i^l(t),f_i^u(t)\}\}$ denotes the selection.

The optimal solution of the new optimization problem becomes a lower bound of problem **DIP2** due to a larger search area that leads to a lower objective function value. After relaxation, all the constraints are linear except (27) which is convex, and the new convex problem **DIP3** is the lower bound of problem **DIP2**, which is as follows

$$\begin{aligned}\textbf{DIP3:}\quad min\quad&\frac{1}{2}p_i(t)^2+(B_i(t)+\nu_i-P_i)p_i(t)\\&+B_i(t)(\nu_i-P_i)+(A(t)-F_e)f_i(t)+\sum_{j=1}^{4}t_j,\end{aligned} \tag{40}$$

$$s.t.\quad \alpha_i^l(t)\le\alpha_i(t)\le\alpha_i^u(t), \tag{41}$$

$$p_i^l(t)\le p_i(t)\le p_i^u(t), \tag{42}$$

$$f_i^l(t)\le f_i(t)\le f_i^u(t), \tag{43}$$

$$(27)-(29),(36)-(39). \tag{44}$$

A distributed algorithm is proposed based on the branch-and-bound method to derive the solution of problem **DIP1**, where only the lower bound problem **DIP3** is solved. It is noted that constraints (41)-(43), (27)-(29), and (36)-(39) define the search area of problem **DIP3**. In the branch-and-bound procedure, we divide and bound the variable ranges according to the lengths of variable ranges expressed by constraints (41)-(43), and include remaining constraints ((27)-(29) and (36)-(39)) in the solution process of the convex optimization problem **DIP3**. The specific algorithm is shown as **Algorithm 1**.

In Algorithm 1, the objective function value S is obtained, and then the delay value Z is calculated by $Z=\frac{S-g(S)}{V}$, where function $g(\cdot)$ is as follows

$$\begin{aligned}g(\cdot)=&\frac{1}{2}p_i(t)^2+(B_i(t)+\nu_i-P_i)p_i(t)\\&+B_i(t)(\nu_i-P_i)+(A(t)-F_e)f_i(t),\end{aligned} \tag{45}$$

and $g(S)$ is the function value where S is the value of the objective function of **DIP1**. At line 3 of the algorithm, the convex optimization problem **DIP3** can be efficiently solved by optimization solvers. In line 7, there are three variables ($\alpha_i(t)$, $f_i(t)$ and $p_i(t)$) that can be selected and divided the range, the variable that has the largest range is selected, and its range is equally divided into two; then, together with other two variable ranges, there are two new search regions, H_1 and H_2. The reason for selecting the largest range is that a larger range may have a larger variation of the objective function value, which may speed the finding of a lower objective value. We note that in line 8, where the convex optimization problem is solved with two different search regions H_1 and H_2, the relaxations of the four constraints (36)-(39) change according to the ranges of variables. In lines 9 and 10, the two solutions ($X(H_1)$ and $X(H_2)$) of problem **DIP3** are put into the objective function of problem **DIP1**, where two solutions are feasible solutions of problem **DIP1** because the constraints of problem **DIP1** are contained by problem **DIP3**. From lines 12 to 14, some search regions are deleted from H, this is because these regions will not generate better solutions than the solution obtained already. Line 15 sets the LB value as the minimal lower bound in the H. Line 16 decides the search region H_0 in the next iteration, where the region with the minimal lower bound is selected, and will be equally divided into two smaller regions. Then, the two new regions can generate two new lower bounds, $LB(H_1)$ and $LB(H_2)$, and they must be no smaller than the $LB(H_0)$ which is also the minimal lower bound in H. This can increase the LB value and reduce the gap between S and LB in the next iteration. In the new iteration, the difference between S and LB is checked, if the difference is no larger than ξ, the algorithm

Algorithm 1: Distributed algorithm for computation offloading

```
Input: Computation request arrival rate $\lambda_i(t)$ of IoT device, power limitation $P_i$, IoT device computation capacity $F_i$, edge node computation capacity $F_e$, lengths of virtual queues $A(t)$ and $B_i(t)$, accuracy parameter $\xi$.
Output: Solution $X = \{\alpha_i(t), f_i(t), p_i(t)\}$, objective function value $S$, and delay value $Z = \frac{S-g(S)}{V}$.
$H \leftarrow \emptyset$.
$H_0 = \{(\alpha_i(t), f_i(t), p_i(t)|0 \leq \alpha_i(t) \leq 1, 0 \leq f_i(t) \leq F_e, 0 \leq p_i(t) \leq P_i\}$.
Solve the convex problem DIP3 with constraints (27)-(29), (36)-(38) and $H_0$. The obtained objective value of DIP3 is $LB(H_0)$, and the solution is $X(H_0)$.
Assign the values of $X(H_0)$ to the variables in the objective function DIP1; obtain the objective function value $S =$ DIP1$(X(H_0))$.
Set the minimal lower bound $LB = LB(H_0)$, and $X = X(H_0)$.
while $(S - LB) > \xi$ do
    Choose one of the three variables ($\alpha_i(t)$, $f_i(t)$, $p_i(t)$) that has the largest range in $H_0$, then equally divide the range of the selected variable into two to obtain two new search areas $H_1$ and $H_2$.
    Solve the convex optimization problem DIP3 twice: in the first time with constraints (27)-(29), (36)-(38) and $H_1$, and in the second time with constraints (27)-(29), (36)-(38) and $H_2$. The obtained objective values are $LB(H_1)$ and $LB(H_2)$, and the solutions are $X(H_1)$ and $X(H_2)$.
    $S = \min\{S,$ DIP1$(X(H_1))$, DIP1$(X(H_2))\}$.
    $X = \arg\min\{S,$ DIP1$(X(H_1))$, DIP1$(X(H_2))\}$.
    $H = H \cup H_1 \cup H_2$.
    For $\Omega \in H$
        If $S \leq LB(\Omega)$
            Delete $\Omega$ from $H$.
    $LB = \min\{LB(\Omega)|\Omega \in H\}$.
    $H_0 = \arg\min\{LB(\Omega)|\Omega \in H\}$.
    Delete $H_0$ from $H$.
Return.
```

stops, otherwise, the selected search region H_0 are further divided.

Theorem 2: Algorithm 1 is convergent, when $\xi \geq 0$.

Proof: For a search region $H_0 = \{\alpha_i(t), f_i(t), p_i(t)|\alpha_i^l(t) \leq \alpha_i(t) \leq \alpha_i^u(t), f_i^l(t) \leq f_i(t) \leq f_i^u(t), p_i^l(t) \leq p_i(t) \leq p_i^u(t)\}$, if all three variable ranges approach 0, that means $|\alpha_i^u(t) - \alpha_i^l(t)| \rightarrow 0$, $|f_i^u(t) - f_i^l(t)| \rightarrow 0$, and $|p_i^u(t) - p_i^l(t)| \rightarrow 0$, the four relaxed fractional terms (36)-(39) are the same as the original ones (31)-(34). Then, the objective value of problem **DIP3** is equal to the objective of problem **DIP2** (which is also equal to the objective of problem **DIP1** due to the equivalent problem), so $(S - LB) \rightarrow 0$ is obtained. In a word, in Algorithm 1, the branch-and-bound makes the variable ranges small enough, then $(S - LB) \leq \xi$ will be obtained, and Algorithm 1 converges. ■

Theorem 3: The long-term delay obtained by Algorithm 1 is limited by an upper bound that is the optimal value R^* of the original problem **P1** plus a constant δ, and limited by the lower bound of the optimal value R^*, which is as follows

$$R^* \leq \lim_{T\rightarrow\infty} \frac{1}{T} \sum_{t=0}^{T-1} \sum_{i=1}^{N} Z_i(t) \leq R^* + \delta, \tag{46}$$

where $Z_i(t)$, $i \in \mathcal{N}$ is the delay value Z obtained by Algorithm 1 for IoT device i at time slot t, and

$$\delta = \frac{(N+1)^2 F_e^2 + 4NA_{max}F_e + 2N\xi}{2V} + \frac{\sum_{i=1}^{N}(5P_i^2 + \nu_i^2 + 6B_{max}P_i + 4P_i\nu_i + 4B_{max}\nu_i)}{2V}.$$

Proof: The proof of the lower bound is quite obvious as R^* is the minimal value of the delay, and any other values with the same search area are larger than R^*, this proves the lower bound.

When Algorithm 1 stops, we have $S_i - LB_i \leq \xi$, where S_i and LB_i are the objective value and the minimal lower bound obtained by the algorithm at IoT device $i, i \in \mathcal{N}$, respectively. Suppose S_i^* is the optimal objective value of problem **DIP1** at IoT device $i, i \in \mathcal{N}$, we have $S_i - S_i^* \leq \xi$ because $LB_i \leq S_i^* \leq S_i$. Then we have

$$S_i^* \geq S_i - \xi. \tag{47}$$

From Theorem 1, we have $\lim_{T\rightarrow\infty} \frac{1}{T}\sum_{t=0}^{T-1}\sum_{i=1}^{N} R_i'(t) \leq R^* + \varepsilon$, and the term at the left-hand side of the inequality is the value obtained by problem **P4**. It is noted that problem **DIP1** at N IoT devices solve problem **DIP1** distributively, so

$$\lim_{T\rightarrow\infty} \frac{1}{T} \sum_{t=0}^{T-1} \sum_{i=1}^{N} R_i'(t) = \frac{1}{V} \sum_{i=1}^{N} (S_i^* - g(S_i^*)),$$

where $g(S_i^*)$ is the value of the function $g(\cdot)$ when the entire objective value of **DIP1** is S_i^*. Then from (47), we have

$$R^* + \varepsilon \geq \frac{1}{V} \sum_{i=1}^{N} [S_i - g(S_i^*) - \xi].$$

and we have

$$R^* + \varepsilon \geq \frac{1}{V} \sum_{i=1}^{N} [VZ_i + g(S_i) - g(S_i^*) - \xi],$$

where $Z_i, i \in \mathcal{N}$ is the delay obtained for IoT device i. Then, we have

$$\begin{aligned}
\sum_{i=1}^{N} Z_i &\leq R^* + \varepsilon + \frac{N\xi}{V} + \frac{1}{V} \sum_{i=1}^{N} [g(S_i^*) - g(S_i)] \\
&\leq R^* + \varepsilon + \frac{N\xi}{V} + \frac{1}{V} \sum_{i=1}^{N} \Big[\big(\frac{1}{2}P_i^2 + B_{max}P_i + P_i\nu_i \\
&\qquad + B_{max}\nu_i + A_{max}F_e\big) - (-P_i^2 - B_{max}P_i - F_e^2) \Big] \\
&\leq R^* + \frac{(N+1)^2F_e^2 + 4NA_{max}F_e + 2N\xi}{2V} \\
&\qquad + \frac{\sum_{i=1}^{N}(5P_i^2 + \nu_i^2 + 6B_{max}P_i + 4P_i\nu_i + 4B_{max}\nu_i)}{2V}.
\end{aligned}$$

Finally, after taking the long-term average from time slot 0 to $T-1$ on both sides of the inequality, and let T go to infinite, we have (46), and this proves the upper bound. ■

According to Theorem 3, the gap between the final result obtained by Algorithm 1 and the actual optimal result of the original problem is lower and upper bounded by zero and δ, respectively. However, the exact gap value cannot be obtained, because the actual optimal result cannot be calculated due to the non-convexity of the original problem. Nevertheless, because δ can be controlled, the upper bound of the gap can be limited.

VI. Numerical Results

In this section, we evaluate the performance of the proposed distributed algorithm by experimental runs of Algorithm 1. In the experimental scenarios, each IoT device executes Algorithm 1 to obtain solution $\{\alpha_i(t), f_i(t), p_i(t)\}$, and all IoT devices work asynchronously. The edge node collects $f_i(t), i \in \mathcal{N}$ of IoT devices and updates the virtual queue A according to (12), then the length of the updated $A(t)$ is sent back to all IoT devices. It is noted that only the values of $f_i(t), i \in \mathcal{N}$ and the length of $A(t)$ are exchanged between the edge node and IoT devices, which achieves a low overhead for the communication and the system management.

In the experimental scenarios, IoT devices are randomly scattered around the coverage area of the edge node. As in [36], we set the channel bandwidth $w = 5$ MHz, the background noise $N_0 = -100$ dBm. According to the wireless channel model for IoT environment, we set the channel gain equal to the path loss factor $H_i(t) = d_i^{-4}$, where d_i is the distance between IoT device i and the edge node [48]. The parameter settings in the experiments are listed in Table II.

TABLE II
PARAMETER SETTINGS IN THE EXPERIMENTS

Parameter	Value
N	5, 10, 20, 30, 40
w	5 MHz
N_0	-100 dBm
$\lambda_i(t)$	unif[1, 1.5], unif[1.5, 2], unif[2, 2.5]
d_i	unif[10, 100] m
$H_i(t)$	d_i^{-4}
L_i	unif[5, 8] Kbit
D_i	unif[1, 2] CPU
σ_i	D_i, $5D_i$, $7D_i$
F_i	unif[1, 8] CPU
F_e	30 CPU
P_i	unif[1, 5] W
v_i	$0.9P_i$ W
V	1, 10, 50

Figures 2 and 3 demonstrate the performance of the algorithm for three scenarios, where 10 IoT devices are considered, the computation request arrival rate is uniformly chosen from 1 to 1.5 (unif[1, 1.5] for short), the weight parameter $V = 1$ and computation requirements have the same average as Table II, but different standard deviations (i.e. $\sigma_i = D_i$, $\sigma_i = 5D_i$ and $\sigma_i = 7D_i$). In Fig. 2, the average response time delays per IoT device for the three scenarios are compared. It is observed, as expected, that and the delay value increases when the standard deviation is larger. It is well known from queueing theory that increasing the variability of the demand causes high queueing delay even if the average demand stays constant. This is because high computation demands at certain heavy traffic periods causes high congestion and delay that cannot be offset during light or idle traffic periods as the queue and delay are bounded below by zero. According to Theorem 2, Algorithm 1 converges, and Fig. 2 illustrates the distributed algorithm convergence. Specifically, stable response time delays are attained at three scenarios within 25 time slots, and in the figure, delay values within 30 time slots are shown for brevity. The average drift value per IoT device is shown in Fig. 3, and the average drift value has the same trend as the average response time delay in Fig. 2, where a larger standard deviation leads to a larger average drift value. The effect here is related to the effect of larger standard deviation on the response time delay shown in Fig. 2. That is, a larger standard deviation leads to a longer average queue length, and the drift value is increased. We choose the standard deviation as $\sigma_i = D_i$, $\forall i \in \mathcal{N}$ for the numerical experiments next for brevity.

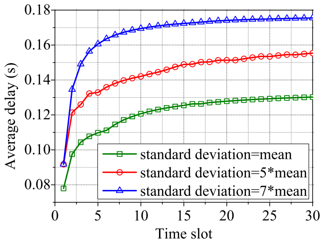


Fig. 2. Average response time delay per IoT device with different standard deviations.

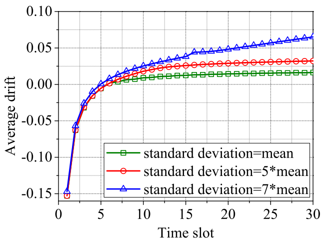


Fig. 3. Average drift per IoT device with different standard deviations.

Figures 4 to 7 demonstrate the performance of the algorithm given three different numbers of IoT devices N associated to the edge node, where the arrival rate is unif[1, 1.5], and the weight parameter $V = 1$. Figure 4 shows the average response time delay per IoT device for the three scenarios, and the average response time delays are stable. In Fig. 4, the scenario with more IoT devices has higher delay values, where the scenario with 30 IoT devices has the highest delay value, sequentially followed by the scenarios with 20 and with 10 IoT devices. This is because more IoT devices compete for a limited computation resource at the edge node, which leads to a higher queueing delay. Figure 5 shows the length of the virtual queue $A(t)$ during the experiments. In the figure, all three scenarios have stable length values, which means the virtual queue A are rate stable in three scenarios. The scenario with 30 IoT devices has the highest virtual queue length value, sequentially followed by the scenarios with 20 and with 10 IoT devices. This is because more IoT devices need more computation resources, and the computation capacity of the edge node is limited, which implies that more than the capacity computation resources are needed and makes the length of the virtual queue increase. It is noted that in the experiments, with

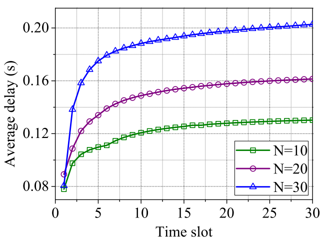


Fig. 4. Average response time delay with different numbers of IoT devices.

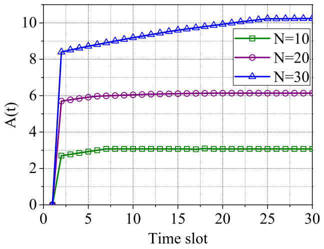


Fig. 5. Length of virtual queue A with different numbers of IoT devices.

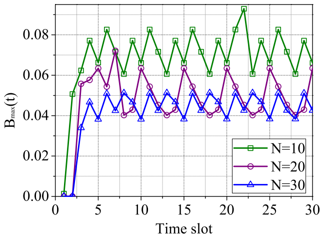


Fig. 6. Value of $B_{max}(t)$ with different numbers of IoT devices.

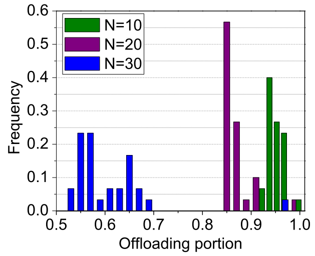


Fig. 7. Distribution of offloading portions with different numbers of IoT devices.

the limited computation resource, the number of IoT devices can be served is decided by the delay tolerance, and computation request arrival rate. For example, if a high response delay tolerance is granted, and computation request arrival rate is low and there are sufficient computation resources, the number of IoT devices served can be very large, otherwise, it is small. In Fig. 6, the largest value of the virtual queues $B_{max}(t) = max\{B_i(t)|i \in \mathcal{N}\}$ among all N IoT devices is compared in the three scenarios. In the figure, three scenarios have limited lengths of the longest virtual queue, which also means that all the virtual queues of the IoT devices are stable. The scenario with 10 IoT devices has the highest virtual queue length value, sequentially followed by the scenarios with 20 and with 30 IoT devices, and this is because more IoT devices compete for the limited computation resources at the edge node, which leads to the reduction of the workload offloaded to the edge node and the reduction of transmission power. Finally, it results in a lower power usage and a shorter length of the virtual queue $B_i(t)$, $i \in \mathcal{N}$ for IoT devices. In the experiments, we calculate the average workload offloading value of all IoT devices at each of the 30 time slots, then we show the distribution of these 30 values as histograms in Fig. 7. Specifically, for each scenario, we calculate the average workload offloading values of all IoT devices at each time slot, then we depict frequencies of the average workload offloading values in 30 average offloading values as histograms. We observe that there are three different distributions for the three scenarios. Specifically, in the scenario with 30 IoT devices, most average offloading values are from 0.52 to 0.7, and the mean value is 0.605. In the scenario with 20 IoT devices, most average offloading values are from 0.85 to 0.91, the mean value is 0.868. In the scenario with 10 IoT devices, most average offloading values are from 0.92 to 1, and the mean value is 0.95. This is because more IoT devices compete for a limited computation resource at the edge node, and a higher queueing delay may incur, which in turn implies reduction of the portion of the workload offloaded to the edge node and compute more workloads locally.

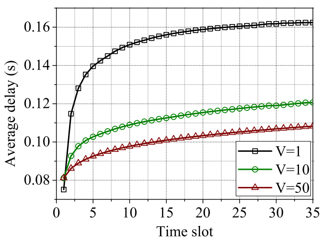


Fig. 8. Average response time delay with different V values.

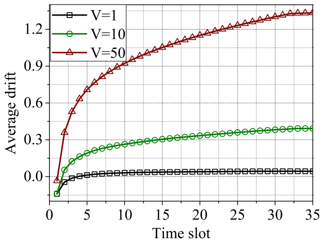


Fig. 9. Average drift per IoT device with different V values.

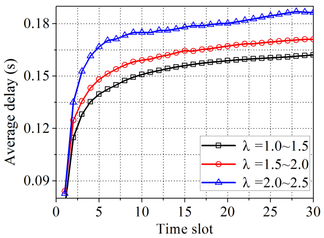


Fig. 10. Average response time delay with different arrival rates.

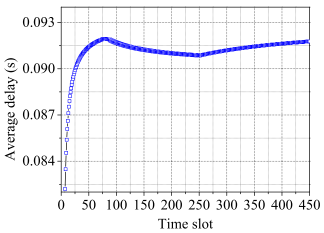


Fig. 11. Average response time delay with dynamic arrival rates.

We also investigate the performance of the algorithm given different values of V in Figs. 8 and 9, where V is the weight parameter of the response time delay and a larger value of V implies a higher impact of the response time delay value on the objective function value of problem **DIP1**. In the experiments, 20 IoT devices are considered, and the arrival rate is unif[1, 1.5]. In Fig. 8, the average response time delays per IoT device are compared for three different V values, 1, 10 and 50. In the figure, when V is set to 1, it has the largest average response time delay, followed by the scenarios of $V = 10$ and $V = 50$. This is because when the value of V increases, a small decrease of the delay incurs a large reduction of the objective function value, then the optimization procedure prefers to reduce the delay value leading to an even lower objective function value. The average drift value per IoT device is also shown in Fig. 9. The trend is just opposite to that of the average response time delay, where a small value of V means a high impact of the drift value to the objective function value, then the optimization procedure reduces the drift that brings more reductions of the objective function value than the delay.

TABLE III
AVERAGE OFFLOADING PORTION

Arrival rate	unif[1, 1.5]	unif[1.5, 2]	unif[2, 2.5]
Offloading portion	0.8718	0.8331	0.8234

Figure 10 shows the average response time delay for three different arrival rates of computation workloads, where 20 IoT devices are considered, and the weight parameter $V = 1$. In the figure, a higher arrival rate leads to a larger average response time delay, specifically, the scenario where the arrival rate of computation requests is uniformly distributed within [2.0, 2.5] has the highest average response time delay, followed by the scenarios with the arrival rates uniformly distributed within [1.5, 2.0] and within [1.0, 1.5]. As expected, the higher arrival rate leads to more computation tasks competing for the limited network and computation resources, which increases the queueing time of computation workload. The average workload offloading portions after convergence in three scenarios are also listed in Table III, where a higher arrival rate shows a lower average workload offloading portion. A similar reason as that of Fig. 7 is that more computation requests at the same time incur a large queueing delay at the edge node, which prefers more workload to be computed locally.

We investigate the performance of the algorithm when the arrival rate is dynamically changed. In Fig. 11, the average response time delay is shown when the arrival rate of the computation workload changes. In the figure, the arrival rate is firstly uniformly distributed within [1.5, 2.0], at the 80th time slot, the arrival rate is reduced by 0.5, so it is uniformly distributed [1.0, 1.5], and as expected, we find that the average response time delay is reduced. At the 230th time slot, the arrival rate is increased by 0.5, so it is again uniformly distributed within [1.5, 2.0], and the average response time delay returns back to its value at the beginning. This demonstrates that the algorithm can handle dynamic traffic scenarios well.

TABLE IV
NUMBER OF TIME SLOTS FOR ALGORITHM CONVERGENCE

Number of IoT devices	5	10	20	30	40
Time slots	13	18	22	24	28

We also show the number of time slots required by the distributed algorithm for convergence in Table IV. We observe this required number of time slots increases when the number of devices increases. This is because each device solves its local problem (executes Algorithm 1) and communicates with the edge node, and more devices incur more communications with the edge node, more rounds are needed to adjust and optimize resource allocations among devices.

VII. Conclusion

We have considered the combination of computation offloading and resource allocation optimization problem in an edge computing network. The objective is to minimize the long-term average delay under constraints of long-term averages of computation and power usage. We have converted the problem into an upper bound problem with the drift-plus-penalty of the Lyapunov optimization. Then a distributed algorithm has been proposed to solve the upper bound problem using the branch-and-bound method. In the branch-and-bound procedure, the problem is relaxed to be a convex optimization problem, which can be solved efficiently and derives the solution of the upper bound problem that has a limited gap to the original solution. Theoretical analysis of the algorithm has also been provided. Numerical results have demonstrated that the proposed distributed algorithm efficiently achieves the target long-term performance, balancing between the delay of computation workload and the drift of the virtual queues.

References

[1] IoT Analytics, “State of the IoT 2018: Number of IoT devices now at 7B-Market accelerating,” 2018, [Online]. Available: https://iot-analytics.com/state-of-the-iot-update-q1-q2-2018-number-of-iot-devices-now-7b/. [Accessed: 10-Feb-2020].

[2] IHS Markit, “IoT Trend Watch 2018,” 2018, [Online]. Available: https://ihsmarkit.com/industry/telecommunications.html. [Accessed: 10-Feb-2020].

[3] M. T. Beck, M. Werner, S. Feld, and T. Schimper, “Mobile edge computing: A taxonomy,” in *Proc. the Sixth International Conference on Advances in Future Internet*, 2014, pp. 48–54.

[4] M. Chiang and T. Zhang, “Fog and IoT: An overview of research opportunities,” *IEEE Internet Things J.*, vol. 3, no. 6, pp. 854–864, Dec 2016.

[5] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, “A survey on mobile edge computing: The communication perspective,” *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2322–2358, 4th quart. 2017.

[6] X. Hu, L. Wang, K. Wong, M. Tao, Y. Zhang, and Z. Zheng, “Edge and central cloud computing: A perfect pairing for high energy efficiency and low-latency,” *IEEE Trans. Wireless Commun.*, pp. 1–1, 2019.

[7] C. Wang, Y. He, F. R. Yu, Q. Chen, and L. Tang, “Integration of networking, caching, and computing in wireless systems: A survey, some research issues, and challenges,” *IEEE Commun. Surveys Tuts.*, vol. 20, no. 1, pp. 7–38, 2018.

[8] C. Wang, C. Liang, F. R. Yu, Q. Chen, and L. Tang, “Computation offloading and resource allocation in wireless cellular networks with mobile edge computing,” *IEEE Trans. Wireless Commun.*, vol. 16, no. 8, pp. 4924–4938, Aug 2017.

[9] Y. Mao, J. Zhang, S. H. Song, and K. B. Letaief, “Stochastic joint radio and computational resource management for multi-user mobile-edge computing systems,” *IEEE Trans. Wireless Commun.*, vol. 16, no. 9, pp. 5994–6009, Sep 2017.

[10] E. Li, L. Zeng, Z. Zhou, and X. Chen, “Edge AI: On-demand accelerating deep neural network inference via edge computing,” *IEEE Trans. Wireless Commun.*, vol. 19, no. 1, pp. 447–457, Jan 2020.

[11] C. Zhao, Y. Cai, A. Liu, M. Zhao, and L. Hanzo, “Mobile edge computing meets mmWave communications: Joint beamforming and resource allocation for system delay minimization,” *IEEE Trans. Wireless Commun.*, pp. 1–1, 2020.

[12] Z. Yang, C. Pan, K. Wang, and M. Shikh-Bahaei, “Energy efficient resource allocation in UAV-enabled mobile edge computing networks,” *IEEE Trans. Wireless Commun.*, vol. 18, no. 9, pp. 4576–4589, Sep 2019.

[13] S. N. Shirazi, A. Gouglidis, A. Farshad, and D. Hutchison, “The extended cloud: Review and analysis of mobile edge computing and fog from a security and resilience perspective,” *IEEE J. Sel. Areas Commun.*, vol. 35, no. 11, pp. 2586–2595, Nov 2017.

[14] M. J. Neely, “Stochastic network optimization with application to communication and queueing systems,” in Synthesis Lectures on Communication Networks, San Rafael, CA, USA: Morgan & Claypool, 2010, vol. 3, pp. 1-211, 1.

[15] Y. Cui, V. Lau, R. Wang, H. Huang, and S. Zhang, “A survey on delay-aware resource control for wireless systems - large deviation theory, stochastic Lyapunov drift, and distributed stochastic learning,” *IEEE Trans. Inf. Theory*, vol. 58, no. 3, pp. 1677–1701, Mar 2012.

[16] P. Blasco, D. Gunduz, and M. Dohler, “A learning theoretic approach to energy harvesting communication system optimization,” *IEEE Trans. Wireless Commun.*, vol. 12, no. 4, pp. 1872–1882, Nov 2013.

[17] A. Munir and A. Gordon-Ross, “An MDP-based dynamic optimization methodology for wireless sensor networks,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 23, no. 4, pp. 616–625, Apr 2012.

[18] A. Minasian, S. ShahbazPanahi, and R. S. Adve, “Energy harvesting cooperative communication systems,” *IEEE Trans. Wireless Commun.*, vol. 13, no. 11, pp. 6118–6131, Nov 2014.

[19] G. P. Fettweis, “The Tactile Internet: Applications and challenges,” *IEEE Veh. Technol. Mag.*, vol. 9, no. 1, pp. 64–70, Mar 2014.

[20] A. A. Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, “Internet of Things: A survey on enabling technologies, protocols, and applications,” *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 2347–2376, 4th Quart. 2015.

[21] Juniper, White Paper, "Smart wireless devices and the Internet of me," Mar 2015, [Online]. Available: http://itersnews.com/wp-content/uploads/experts/2015/03/96079Smart-Wireless-Devices-and-the-Internet-of-Me.pdf. [Accessed: 10-Feb-2020].

[22] B. Farahani, F. Firouzi, V. Chang, M. Badaroglu, N. Constant, and K. Mankodiya, "Towards fog-driven IoT eHealth: Promises and challenges of IoT in medicine and healthcare," *Future Generation Computer Systems*, vol. 78, pp. 659 – 676, 2018.

[23] AT&T Newsroom, "The cloud comes to you: AT&T to power self-driving cars, AR/VR and other future 5G applications through edge computing," Jul 2017, [Online]. Available: http://about.att.com/story/reinventing the cloud through edge computing.html. [Accessed: 10-Feb-2020].

[24] T. Dang and M. Peng, "Joint radio communication, caching, and computing design for mobile virtual reality delivery in fog radio access networks," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 7, pp. 1594–1607, Jul 2019.

[25] A. V. Dastjerdi, H. Gupta, R. N. Calheiros, S. K. Ghosh, and R. Buyya, "Fog computing: Principles, architectures, and applications," *arXiv e-prints*, Jan 2016.

[26] Y. Cao, S. Chen, P. Hou, and D. Brown, "FAST: A fog computing assisted distributed analytics system to monitor fall for stroke mitigation," in *Proc. IEEE International Conference on Networking, Architecture and Storage*, Aug 2015, pp. 2–11.

[27] J. K. Zao, T. T. Gan, C. K. You, S. J. R. Méndez, C. E. Chung, Y. T. Wang, T. Mullen, and T. P. Jung, "Augmented brain computer interaction based on fog computing and linked data," in *Proc. International Conference on Intelligent Environments*, Jun 2014, pp. 374–377.

[28] J. Zhu, D. S. Chan, M. S. Prabhu, P. Natarajan, H. Hu, and F. Bonomi, "Improving web sites performance using edge servers in fog computing architecture," in *Proc. IEEE Seventh International Symposium on Service-Oriented System Engineering*, Mar 2013, pp. 320–323.

[29] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, "Mobile edge computing: A survey," *IEEE Internet Things J.*, vol. 5, no. 1, pp. 450–465, Feb 2018.

[30] T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta, and D. Sabella, "On multi-access edge computing: A survey of the emerging 5G network edge cloud architecture and orchestration," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 3, pp. 1657–1681, 3rd quart. 2017.

[31] C. You, K. Huang, H. Chae, and B. Kim, "Energy-efficient resource allocation for mobile-edge computation offloading," *IEEE Trans. Wireless Commun.*, vol. 16, no. 3, pp. 1397–1411, Mar 2017.

[32] Y. Xiao and M. Krunz, "QoE and power efficiency tradeoff for fog computing networks with fog node cooperation," in *Proc. IEEE INFOCOM*, May 2017, pp. 1–9.

[33] Y. Xiao and M. Krunz, "Dynamic network slicing for scalable fog computing systems with energy harvesting," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 12, pp. 2640–2654, Dec 2018.

[34] G. Zhang, W. Zhang, Y. Cao, D. Li, and L. Wang, "Energy-delay tradeoff for dynamic offloading in mobile-edge computing system with energy harvesting devices," *IEEE Trans. Ind. Informat*, vol. 14, no. 10, pp. 4642–4655, Oct 2018.

[35] Y. Mao, J. Zhang, and K. B. Letaief, "Dynamic computation offloading for mobile-edge computing with energy harvesting devices," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 12, pp. 3590–3605, Dec 2016.

[36] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE/ACM Trans. Netw.*, vol. 24, no. 5, pp. 2795–2808, Oct 2016.

[37] S. Sardellitti, G. Scutari, and S. Barbarossa, "Joint optimization of radio and computational resources for multicell mobile-edge computing," *IEEE Trans. Signal Inf. Process. Over Netw.*, vol. 1, no. 2, pp. 89–103, Jun 2015.

[38] M. Chen and Y. Hao, "Task offloading for mobile edge computing in software defined ultra-dense network," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 3, pp. 587–597, Mar 2018.

[39] J. Zhu, J. Wang, Y. Huang, F. Fang, K. Navaie, and Z. Ding, "Resource allocation for hybrid NOMA MEC offloading," *IEEE Trans. Wireless Commun.*, pp. 1–1, 2020.

[40] F. Wang, J. Xu, and S. Cui, "Optimal energy allocation and task offloading policy for wireless powered mobile edge computing systems," *IEEE Trans. Wireless Commun.*, vol. 19, no. 4, pp. 2443–2459, Apr 2020.

[41] J. He, Z. Xue, D. Wu, D. Wu, and Y. Wen, "CBM: online strategies on cost-aware buffer management for mobile video streaming," *IEEE Trans. Multimedia*, vol. 16, no. 1, pp. 242–252, Jan 2014.

[42] C. Qiu, Y. Hu, Y. Chen, and B. Zeng, "Lyapunov optimization for energy harvesting wireless sensor communications," *IEEE Internet Things J.*, vol. 5, no. 3, pp. 1947–1956, Jun 2018.

[43] T. J. Ott, "Simple inequalities for the D/G/1 queue," *Operations Research*, vol. 35, no. 4, pp. 589–597, 1987.

[44] J. P. Champati, H. Al-Zubaidy, and J. Gross, "Statistical guarantee optimization for age of information for the D/G/1 queue," in *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications Workshops*, 2018, pp. 130–135.

[45] F. Metzger, T. Hofeld, A. Bauer, S. Kounev, and P. E. Heegaard, "Modeling of aggregated IoT traffic and its application to an IoT cloud," *Proceedings of the IEEE*, vol. 107, no. 4, pp. 679–694, 2019.

[46] M. Zukerman, "Introduction to queueing theory and stochastic teletraffic models," 2019, [Online]. Available: http://www.ee.cityu.edu.hk/∼zukerman/classnotes.pdf. [Accessed: 10-Feb-2020].

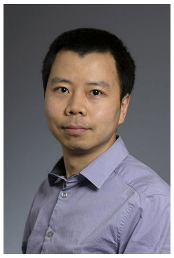
[47] P. Shen, T. Zhang, and C. Wang, "Solving a class of generalized fractional programming problems using the feasibility of linear programs," *J. Inequalities Appl.*, vol. 2017, no. 147, Jun 2017.

[48] T. S. Rappaport, *Wireless communications: Principles and practice*, 2nd ed. Prentice Hall, 2002.

Rongping Lin received the Ph.D. degree from the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, 2013. He is currently an Associate Professor with the School of Information and Communication Engineering, University of Electronic Science and Technology of China (UESTC), Chengdu, P.R. China. From 2013 to 2014, he was a Senior Research Assistant in City University of Hong Kong. His research interests include optimization, machine learning, and their applications in wire/wireless networks.

Zhijie Zhou is pursuing the M.S. degree with the School of Information and Communication Engineering, UESTC, P.R. China. His research interest includes optimization and edge computing.

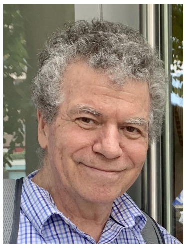
Shan Luo received the Ph.D. degree in information engineering from Nanyang Technological University in 2014. She is currently an Associate Professor with the School of Aeronautics and Astronautics, UESTC, P.R. China. Her research interests include wireless communications and optimization.



Yong Xiao (S'09-M'13-SM'15) is a professor in the School of Electronic Information and Communications at the Huazhong University of Science and Technology (HUST), Wuhan, China. Before he joins HUST, he was a research assistant professor in the Department of Electrical and Computer Engineering at the University of Arizona where he was also the center manager of the Broadband Wireless Access and Applications Center (BWAC), an NSF Industry/University Cooperative Research Center (I/UCRC) led by the University of Arizona. 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).

Xiong Wang is an Associate Professor with the School of Information and Communication Engineering, UESTC, P.R.China. His research interests include network measurement, modeling and optimization, algorithm analysis and design, network management in communication networks.

Sheng Wang is a Professor with the UESTC. His research interests include planning and optimization of wire and wireless networks, next generation of internet, and next-generation optical networks. He is a Senior Member of the Communication Society of China, a Member of the ACM, and a Member of the China Computer Federation.



Moshe Zukerman (M'87-SM'91-F'07-LF'20) received the B.Sc. degree in industrial engineering and management, the M.Sc. degree in operations research from the Technion-Israel Institute of Technology, Haifa, Israel, and the Ph.D. degree in engineering from University of California, Los Angeles, in 1985. He was an independent consultant with the IRI Corporation and a Postdoctoral Fellow with the University of California, Los Angeles, in 1985-1986. In 1986-1997, he was with Telstra Research Laboratories (TRL), first as a Research Engineer and, in 1988-1997, as a Project Leader. He also taught and supervised graduate students at Monash University in 1990-2001. During 1997-2008, he was with The University of Melbourne, Victoria, Australia. In 2008 he joined City University of Hong Kong as a Chair Professor of Information Engineering, and a team leader. He has over 300 publications in scientific journals and conference proceedings. He has served on various editorial boards such as Computer Networks, IEEE Communications Magazine, IEEE Journal of Selected Areas in Communications, IEEE/ACM Transactions on Networking and Computer Communications.