Computer Networks 223 (2023) 109556

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Online training data acquisition for federated learning in cloud–edge networks☆

Konglin Zhu [a], Wentao Chen [a], Lei Jiao [b,*], Jiaxing Wang [a], Yuyang Peng [c], Lin Zhang [a]

[a] School of Artificial Intelligence, Beijing University of Posts and Telecommunications, Beijing, China
[b] Department of Computer Science, University of Oregon, Eugene, OR, USA
[c] School of Computer Science and Engineering, Macau University of Science and Technology, Macao Special Administrative Region of China

ARTICLE INFO

Keywords:
Data acquisition
Federated Learning
Cloud–edge networks
Lyapunov control
Online scheduling

ABSTRACT

Federated learning (FL) is an effective approach to exploiting different data sources for collaborative model training while maintaining the privacy of users. Adequate data is necessary to improve the accuracy of the federated learning. However, it is difficult for all data sources to acquire sufficient data for each iteration of federated training. In this paper, we study the data acquisition problem in cloud–edge networks, where edges acquire data from users and conduct the local training while the cloud aggregates the local models. Since data acquired by users with uncertainty, it is not easy to make the data acquisition decision for the FL online. Even worse, due to the unknown cost of data, it is difficult to make a macro-timescale decision. We propose a two-timescale online scheduling for federated learning to confront the uncertainty of the data acquisition. By learning empirical state information of the system with a carefully designed Lyapunov virtual queue and coordinating the data acquisition in different timescales in an online manner, the proposed approach minimizes the data acquisition cost of federated learning, reduces the data transmission delay and accelerates the convergence speed of federated learning. Rigorous theoretical analysis shows strong performance guarantees of the proposed two-timescale Lyapunov optimization algorithm and extensive trace-driven experimental results suggests that the algorithm achieves outstanding performance gains over existing benchmarks.

1. Introduction

Federated learning (FL) [1–3] is a kind of distributed machine learning, which receives increasing interest in the forthcoming era of 5G, Internet of Things, and Artificial. Intelligence. In the paradigm of federated learning, the local model is trained distributively and then aggregated to a global model. The global model is shared distributively for model updates. The procedure iterates until the desired accuracy is achieved. During the FL training procedure, it always assumes that the data is adequate for model convergence. However, it is not realistic in many scenarios, especially in the case that the data needs to be purchased from users. For example, the perception function of autonomous vehicles needs to collect images from local vehicles for training the local model by the edge. The data needed may be underestimated by the edge at the beginning of the FL training, which may lead the edge to purchase more data during the FL training. Therefore, it is a huge challenge to acquire sufficient data for distributed FL systems.

Recent studies [4–6] focusing on data acquisition for machine learning usually use the cloud data warehouse for data collecting, organizing and storing. These data acquisition decisions are one-time by the central server. They cannot be easily adapted to the FL systems because the data acquisition decision is distributed and online. In this paper, we study a two-timescale data acquisition approach for the FL system. As shown in Fig. 1, in the macro-timescale, the system estimates the need for data and purchases the data from the users. The edge uses these data for local training to achieve the goal. In the case that the data purchased in the macro-timescale is not sufficient for the FL, the edge needs to make the data acquisition decision again in the micro-timescale to obtain more data. Such a two-timescale data acquisition

☆ This work was supported in part by the National Key Research and Development Program of China under Grant 2022YFB2503202, in part by the Beijing Natural Science Foundation under Grant 4222033, in part by the Hunan Province Science and Technology Project Funds under Grant 2018TP1036, in part by the Key Laboratory of Modern Measurement and Control Technology, Ministry of Education, Beijing Information Science and Technology University under Grant KF20211123202, in part by the Beijing Advanced Innovation Center for Future Blockchain and Privacy Computing and in part by the U.S. National Science Foundation under Grant CNS-2047719 and CNS-2225949.

* Corresponding author.
E-mail addresses: klzhu@bupt.edu.cn (K. Zhu), chenwentao@bupt.edu.cn (W. Chen), jiao@cs.uoregon.edu (L. Jiao), wjiaxing@bupt.edu.cn (J. Wang), yypeng@must.edu.mo (Y. Peng), zhanglin@bupt.edu.cn (L. Zhang).

https://doi.org/10.1016/j.comnet.2023.109556
Received 20 September 2022; Received in revised form 24 November 2022; Accepted 1 January 2023
Available online 7 January 2023

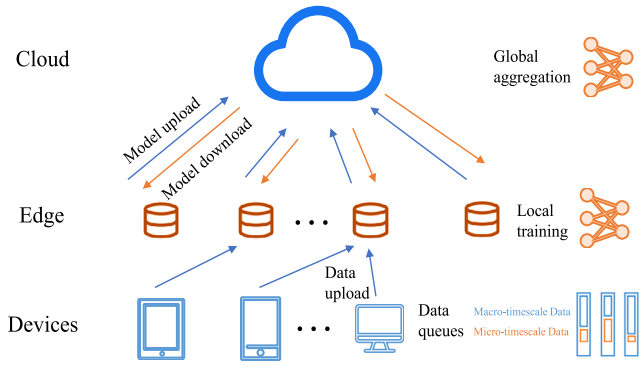

Fig. 1. System architecture.

in the FL system is not an easy task and faces multiple challenges as follows.

First of all, the two timescales are time-coupled. Under the time-varying system state information, the data waiting for training is determined not only by the accumulated data queue but also by the data arrival and departure in the macro-timescale and micro-timescale. Therefore, the two-timescale data acquisition problem is an online problem that needs an online solution to decouple the different timescales.

Secondly, the data queue is unstable and often congested. To maintain the stability of the queue, the Lyapunov control approach [7] is usually applied. However, existing Lyapunov control methods [8] react only to the instantaneous system condition thus resulting in slow convergence of data queue and large delay. We need to use the empirical system state for the Lyapunov control approach to enhance the convergence speed and meanwhile reduce the delay for data acquisition.

Thirdly, pursuing the cost optimization of the edges should not sacrifice the quality of the models being trained [9]. The cost of data acquisition, data transmission as well as model iterations [10,11] is usually nonlinearly correlated with the quality of the model. Intertwined with Lyapunov integer decision variables control, the problem can often be NP-hard. Even in the offline setting, it is difficult to strike a balance between data acquisition and model training not to mention the online setting.

Although existing studies also pay the effort to solve the two-timescale problems, they cannot fully address all of the challenges mentioned above same time. Some studies [12–20] apply Lyapunov technique to deal with different timescales resources problems. However, they only use the instantaneous system condition to optimize the system cost, resulting in slow convergence speed and large delay, especially $[O(1/V), O(V)]$ cost-delay tradeoff. Some studies use Lyapunov optimization directly [13,14,16,21–25] for convex problems or relaxed convex problems, which are not applicable to our problem. Others [12,26–29] face the NP-hard problem but only use the heuristic algorithms, which have no proof of approximation ratio and may not produce optimal results. Zhou et al. [30] tend to minimize the total resource cost while keeping the model's quality. But they take the model's parameters as input, so they cannot control the model's quality online according to the dynamic and time-varying resource price.

In this paper, we formulate the online optimization problem of minimizing the cost of data acquisition for the edge–cloud FL system as a mixed-integer problem. The goal of the formulation is to strike to minimize the total cost of the system and meanwhile maintain the quality of the FL models being trained by controlling the data acquisition rhythm and FL convergence parameters. The problem characterizes the cost of data acquisition, data transmission, and model iteration, the accuracy of models and data queue stability over the time-varying data acquisition of the local training data.

We propose a novel algorithm named Two-timescale Online schEduling for Federated Learning ($TOEFL$), to acquire data and minimize the cost while maintaining the accuracy of the model being trained using the two-timescale Lyapunov approach. First, we use the two-timescale Lyapunov approach to make data acquisition and transmission decisions without a priori knowledge. Next, utilizing the characteristics of Lyapunov queues as the Lagrange multipliers, we design the empirical virtual queue which learns empirical state information for optimization, yielding lower delay and higher convergence speed. Finally, we carry out a novel Randomized Rounding with Weight (RRW) algorithm which solves the NP-hard problem in the Lyapunov function. It converts fractional solutions from our relaxed problem into integers by rounding pairs of fractions in opposite directions for compensation using weights information, without violating constraints.

We provide rigorous performance analysis and extensive evaluation of our proposed algorithms. We prove that our algorithm achieves closer to optimal performance compared with the other algorithms, i.e., the algorithm can achieve $[O(1/V), O([log(V)]^2)]$ cost-delay tradeoff, which significantly improves the $[O(1/V), O(V)]$ cost-delay tradeoff of the traditional Lyapunov technique. We evaluate the practical performance of our approach extensively, via training different models with federated learning settings. Our proposed algorithm achieves faster queue length convergence speed and near-optimal average cost while producing machine learning models with satisfiable inference accuracy across different training tasks.

The remainder of this paper proceeds as follows. In Section 2, we survey the related works. In Section 3, we present the basic model and the problem formulation of our system. In Section 4, the $TOEFL$ algorithm is developed to minimize the total cost while keeping queues stable and models' quality in the Lyapunov NP-hard problem. In Section 5, we present the performance analysis. In Section 6, we provide the simulation results. Section 7 concludes this paper.

2. Related works

We categorize and discuss related work in different groups, and then point out how our work differs from each group.

Resources Consumption of FL Systems: Yang et al. [31] investigate the problem of energy-efficient transmission and computation resource allocation for FL over wireless communication networks. Dinh et al. [32] employ FL in wireless networks as a resource allocation optimization problem that captures the trade-off between FL convergence wall clock time and energy consumption of UEs with heterogeneous computing and power resources. Pu et al. [33] propose Cocktail, a cost-efficient and data skew-aware online in-network distributed machine learning framework and build a comprehensive model and formulate an online data scheduling problem to optimize the framework cost. Abad et al. [34] optimize the resource allocation among mobile users to reduce the communication latency in learning iterations. Trans et al. [35] study the trade-offs between computation and communication latencies determined by learning accuracy level, and thus between the federated Learning time and UE energy consumption. Ye et al. [36] propose a selective model aggregation approach, where "fine" local DNN models are selected and sent to the central server by evaluating the local image quality and computation capability. Li et al. [37] propose a novel optimization objective inspired by fair resource allocation in wireless networks that encourages a fairer (specifically, more uniform) accuracy distribution across devices in federated networks.

Two-timescale Lyapunov Algorithm: Zhang et al. [12] determine the harvested and purchased energy on a large time scale and the channel allocation and data collection on a small time scale based on the two-timescale Lyapunov algorithm. Li et al. [13] propose a two-timescale Lyapunov optimization algorithm to overcome the uncertainty of the system's future information and make optimal decisions only based on the system's current states. Yao et al. [14] focus on a stochastic optimization-based approach to make distributed routing

and server management decisions in the context of large-scale, geographically distributed data centers using the two-timescale Lyapunov algorithm. Feng et al. [15] investigate the dynamic resource allocation for content transmission in wireless-powered device-to-device communications with self-interested nodes including device-to-device transmitters and power stations. Hu et al. [16] apply two-timescale Lyapunov optimization to transform the long-term optimization problem into the drift-plus-penalty and also conduct a theoretic analysis of the proposed energy management strategy. Deng et al. [17] propose an online control algorithm for the data centers with a power supply system based on the two-timescale Lyapunov optimization techniques. Ma et al. [18] propose a T-slot predictive service placement algorithm to incorporate the prediction of user mobility based on the two-timescale Lyapunov optimization method. Ren et al. [19] study the problem of scheduling batch jobs, which originate from multiple organizations/users and are scheduled to multiple geographically distributed data centers. Zhang et al. [20] formulate the problem into an optimization programming to achieve optimal decisions for energy scheduling and sleep control and implement the power scheduling and data transmission in time slots by adopting a two-timescale approach. For solving Lyapunov problems combined with NP-hard problem, Joshi et al. [26] simply relaxes the problem into a convex problem. Zhang et al. [12] falls into the category of bipartite matching with conflict pairs, which has been proven to be NP-hard. They apply the Cross Entropy algorithm to solve it. Feng [15] makes some linear approximations by employing the Taylor expansion, which makes the optimization terms become a convex problem. Peng et al. [27] use C-additive approximation algorithm to get the local optimal solution. Li et al. [28] deals with a mixed-integer problem using a distributed optimized algorithm, while Chang et al. [29] deals with a similar mixed-integer problem but using semi-smooth Newton method.

Our research in this paper differs from both of the above two groups of research. Those [31–37] focus on optimizing the federated learning systems in wireless networks, edge computing environments, and vehicular or other platforms while ignoring the data acquisition process of FL and assuming that all the data exists in the local clients in the beginning. Those [12–20] use Lyapunov optimization techniques to deal with different timescales' problem. But they only use the instantaneous system condition to optimize the system cost, resulting in slow convergence speed and large delay. What is more, those [12,15,27–29] face the NP-hard problem in Lyapunov techniques. They only apply heuristic algorithms to solve it, which have no proof of approximation ratio and may not produce optimal results. To sum up, none of the existing research, to the best of our knowledge, have studied online NP-hard optimization problem of minimizing the cost of data acquisition for edge–cloud FL system with guaranteed model quality and data queue stability in faster convergence speed.

3. Model and problem formulation

3.1. System model

We study online training data acquisition problem for FL in cloud–edge networks, which includes a cloud, a crowd of edges and devices. During the whole process, a set of devices collect data, indexed by $i \in \mathrm{M} = \{1, 2, \ldots, M\}$. Also, a set of edges are used for devices' data acquirement and iterative updates of local models, denoted by $j \in \mathrm{N} = \{1, 2, \ldots, N\}$. All edges are connected to a cloud which is for the aggregation of global model parameters.

Two-timescale Manner: We consider our system to operate in a two-timescale manner. As shown in Fig. 2, time is divided into K ($K \in N+$) time frames named as macro-timescale. Each time frame is further divided into T ($T \in N+$) time slots named as micro-timescale. For acquiring data from devices in different timescales, the system makes decisions as follows: At macro-timeslot $t \in kT(k = 0, 1, \ldots, K-1)$,

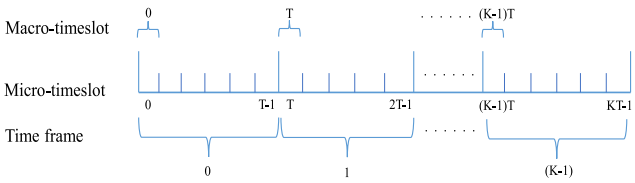

Fig. 2. Two-timescale manner.

the system observes the current data demanded $R^{(t)}$ and the macro-timescale data acquisition price $a_i^{(t)}$. Then it decides on how much amount of macro timescale data $u_i(t)$ to be acquired for macro-timescale use. At micro-timeslot $\tau \in [t+1, t+T-1]$, according to the current data demand for federated learning $R^{(\tau)}$, current macro-timescale data $u_i(t)$ and the micro-timescale data acquisition price $r_i^{(t)}$, the system decides how much amount of additional data acquisition $z_i(t)$ in a micro-timescale manner.

Federated Learning: In the setup of a typical machine learning problem, for a sample data $\{x_n, y_n\}$ with a multi-dimension input feature x_n and a corresponding labeled output y_n, the objective is to find a model parameter vector w that maps x_n to y_n via a loss function $f_n(w)$.

For an edge server dataset $j \in N$ with a number of D_j data samples, the loss function on this dataset is defined as:

$$F_j(\boldsymbol{w}) = \frac{1}{D_j} \sum_{n=1}^{D_j} f_n(\boldsymbol{w}). \tag{1}$$

Then, the learning problem can be formulated as finding an optimal model parameter vector $\boldsymbol{w}^*$ to minimize the following global loss function $F(\boldsymbol{w})$ over the number of N local datasets:

$$\boldsymbol{w}^* = \underset{\boldsymbol{w}}{\operatorname{argmin}} F(\boldsymbol{w}) = \underset{\boldsymbol{w}}{\operatorname{argmin}} \frac{\sum_{j=1}^{N} D_j F_j(\boldsymbol{w})}{\sum_{j=1}^{N} D_j}. \tag{2}$$

In this paper, we consider the federated learning procedure as follows. In each time slot $\tau \in [t, t+T-1]$, there are multiple rounds of local iteration followed by one global aggregation. We use $\eta_j(\tau)$ as the actual local error rate and $\epsilon(\tau)$ as the actual global error rate at each time slot τ. Additionally, we use $v log(1/\eta_j(\tau))$ in [31] to approximate the number of local iterations in each global aggregation, where v is a constant. And the global error rate ϵ can be bounded by the following equation, where l represents the demanded error between $\eta_j(\tau)$ and η,

$$\begin{aligned} \epsilon \le \epsilon(\tau) &= \exp\left(\frac{-\tau(1-\eta_j(\tau))\gamma^2\xi}{2L^2}\right) \\ &\le \exp\left(\frac{-\tau(1-\eta-l)\gamma^2\xi}{2L^2}\right). \end{aligned} \tag{3}$$

Then the global loss function $F(\boldsymbol{w}^{(\tau)})$ at time slot $\tau \in [t, t+T-1]$ is bounded by the following equation:

$$\begin{aligned} F(\boldsymbol{w}^{(\tau)}) - F\left(\boldsymbol{w}^*\right) &\le \epsilon\left(F(\boldsymbol{w}^{(0)}) - F(\boldsymbol{w}^*)\right) \\ &\le \epsilon(\tau)\left(F(\boldsymbol{w}^{(0)}) - F(\boldsymbol{w}^*)\right), \end{aligned} \tag{4}$$

where we assume that $F_j(\boldsymbol{w})$ is L-Lipschitz continuous and γ-strongly convex, and $0 \le \xi \le \frac{\gamma}{L}$.

3.2. Problem formulation

As mentioned above, we need to make decisions on macro-timescale data acquisition $u_i(t)$ at macro-timeslot $t \in kT(k = 0, 1, \ldots, K-1)$ and micro-timescale data acquisition $z_i(\tau)$ at micro-timeslot $\tau \in [t+1, t+T-1]$ according to the current data demand $R^{(\tau)}$. Than the data acquisition cost is $u_i(t)a_i^{(t)} + z_i(\tau)r_i^{(\tau)}$, where $a_i^{(t)}$ denotes the data acquisition price in macro-timescale manner and $r_i^{(\tau)}$ denotes the data acquisition price in micro-timescale manner. What is more, we need to make decisions

on the number of data sets transmitted from device i to edge server j, $x_{ij}(\tau)$ at $\tau \in [t+1, t+T-1]$ at data transmission price of $m_{ij}^{(\tau)}$ and at aggregated model iteration price $b_j^{(\tau)}$. The transmission cost is $x_{ij}(\tau)m_{ij}^{(\tau)}$. And the computation cost is $x_{ij}(\tau)b_j^{(\tau)}vlog(1/\eta_j(\tau))$, where $vlog(1/\eta_j(\tau))$ is the number of local iterations in each global aggregation. Therefore, the cost of multiple tasks in each time slot τ is:

$$Cost(\tau) = \sum_{i\in M}\sum_{j\in N}[u_i(t)a_i^{(t)} + z_i(\tau)r_i^{(\tau)} + x_{ij}(\tau)m_{ij}^{(\tau)} + x_{ij}(\tau)b_j^{(\tau)}vlog(1/\eta_j(\tau))]. \tag{5}$$

In addition, when $\eta_j(\tau)$ increases, the number of local training iterations decreases, leading to the quality of local and global models becoming worse. When $\eta_j(\tau)$ decreases, the total cost will increase as the number of local training iterations increases. Therefore, we need to use a given η to control $\eta_j(\tau)$ so that it can be near our expectations. Using the design of deep learning regularization term, we use the following formula to control $\eta_j(\tau)$:

$$\sum_{j\in N}\lambda(\eta_j(\tau)-\eta)^2. \tag{6}$$

Overall, the total cost in each time slot τ is:

$$Cost(\tau) = \sum_{i\in M}\sum_{j\in N}[u_i(t)a_i^{(t)} + z_i(\tau)r_i^{(\tau)} + x_{ij}(\tau)m_{ij}^{(\tau)} + x_{ij}(\tau)b_j^{(\tau)}vlog(1/\eta_j(\tau)) + \lambda(\eta_j(\tau)-\eta)^2]. \tag{7}$$

Our objective is to minimize the total cost while keeping the stability of the data queues. We formulate the objective function:

$$\min \lim_{t\to\infty}\frac{1}{t}\sum_{\tau=0}^{t-1}\mathrm{Cost}(\tau). \tag{8}$$

s.t.

$$R^{(\tau)} \le \sum_{i\in M}[u_i(t)+z_i(\tau)], \quad \forall t, \forall\tau; \tag{9}$$

$$0 \le u_i(t) \le u^{max}, 0 \le z_i(\tau) \le z^{max}, \quad \forall i, \forall t, \forall\tau; \tag{10}$$

$$0 \le x_i(\tau) \le S^{max}, \quad \forall i, \forall\tau; \tag{11}$$

$$\sum_{j\in N} x_{ij}(\tau) = x_i(\tau), \quad \forall i, \forall\tau, \quad 0 \le x_{ij}(\tau) \le x_{ij}^{max}, \quad \forall i, \forall j, \forall\tau; \tag{12}$$

$$|\eta_j(\tau)-\eta| \le l, 0 \le \eta_j(\tau) \le 1, \quad \forall j; \tag{13}$$

$$Data \quad queues \quad are \quad stable, \quad \forall i, \forall\tau. \tag{14}$$

Constraint (9) indicates that the overall devices' data uploaded should satisfy the current data demand while devices' ability is limited according to Constraint (10). Constraint (11) shows that the ability of data provisioned for edge is limited. Constraint (12) ensures all the data can be uploaded to the edge. Constraint (13) indicates the local error rate's error cannot overrun a limit which is requested by accuracy demand l. Constraint (14) ensures the queue is stable all the time.

In the next section, we will apply the Lyapunov optimization technique to minimize (8) subject to (9)–(14) without knowing any system statistics information. In order to coordinate the diverse cost terms at different timescales, we will propose the $TOEFL$ algorithm. The main symbols used in this paper and their meanings are summarized in Table 1.

4. Two-timescale online scheduling algorithm

In this section, we illustrate the proposed $TOEFL$ algorithm and explain how the algorithm optimizes the total cost in an online manner.

Table 1
Main symbols and their meanings.

Inputs	Meaning
V	An importance weight on the system cost minimization
$a_i^{(t)}$	Data acquisition price in macro-timescale manner
$r_i^{(\tau)}$	Data acquisition price in micro-timescale manner
$m_{ij}^{(\tau)}$	Data transmission price from device i to edge j at time τ
$b_j^{(\tau)}$	Model iteration price at edge j at time τ
$R^{(\tau)}$	Data demanded for federated learning at time τ
λ	Key control parameter for error rate penalty
η	Desired local error rate
l	Demanded error between $\eta_j(\tau)$ and η
M	Set of devices
N	Set of edges
Variables	**Meaning**
$\eta_j(\tau)$	Local error rate at edge j at time τ
$Q_i(\tau)$	Queue backlog at device i at time τ
$H_i(\tau)$	Backlog of delay-aware virtual queue at device i at time τ
$x_i(\tau)$	Data transmission rate from device i at time τ
$x_{ij}(\tau)$	Data transmission rate from device i to edge j at time τ
$u_i(t)$	Data acquisition rate in macro-timescale manner
$z_i(\tau)$	Data acquisition rate in micro-timescale manner

4.1. Lyapunov queue design

Actual Data Queue: As shown in Fig. 1, we construct a queuing system in devices. Since delay-tolerant data from the device i can be acquired before the maximum delay $\overline{Z_i^{\max}}$, we defer data in a waiting queue $Q_i(\tau)$. At each time slot τ, the system receives the data transmission price from device i to edge j $m_{ij}^{(\tau)}$, the model iteration price on the edge $b_j^{(\tau)}$, the local error rate $\eta_j(\tau)$, the actual queue $Q_i(\tau)$ and the delay-aware virtual queue $H_i(\tau)$ as system states. Then it needs to determine how much amount of data should be acquired from device i, which is denoted by $x_i(\tau)$. We assume that $x_i(\tau)$ is bounded by the maximum acquiring rate S^{max} and the current waiting queue $Q_i(\tau)$. Therefore, the amount $Q_i(\tau)$ of device i is given by:

$$Q_i(\tau+1) = Q_i(\tau) - x_i(\tau) + d_i(\tau), \tag{15}$$

where $d_i(\tau) = u_i(t) + z_i(\tau)$. $u_i(t)$ refers to the data acquisition rate in macro-timescale manner and $z_i(\tau)$ refers to the data acquisition rate in micro-timescale manner.

Because of the limitation of the actual data queue, we need to add a constraint:

$$x_i(\tau) \le Q_i(\tau), \quad \forall i, \forall\tau. \tag{16}$$

Delay-Aware Virtual Queue: In order to guarantee the maximum delay $\overline{Z_i^{\max}}$ for data acquisition, we introduce a delay-aware virtual queue $H_i(\tau)$ based on the ϵ_i persistent queue whose update equation is:

$$H_i(\tau+1) = \begin{cases} \max\{H_i(\tau) - x_i(\tau) + \epsilon_i, 0\} & \text{if } Q_i(\tau) > x_i(\tau), \\ 0 \text{ if } Q_i(\tau) = x_i(\tau), \end{cases} \tag{17}$$

where $\epsilon_i > 0$. $H_i(\tau)$ has the same data transmission rate $x_i(\tau)$ as $Q_i(\tau)$, but has an extra arrival constant ϵ_i whenever the actual queue $Q_i(\tau)$ is larger than $x_i(\tau)$. This ensures that $H_i(\tau)$ only grows when there exists data in queue $Q_i(\tau)$ that have not been acquired. Therefore, if there exists data staying in the waiting queue $Q_i(\tau)$ for a long time, the queue length of $H_i(\tau)$ will continue to grow, which makes it easier for the optimizer to focus on the current congestion queues. In any feasible algorithm, we should ensure that the deferred delay-tolerant tasks can be served within a worst-case delay $Z_i^{\max}$ given in Lemma 1:

Lemma 1. *For any time slot τ, if the system can be controlled to ensure that $Q_i(\tau)$ is less than its certain maximum value Q_i^{max} and $H_i(\tau)$ is less*

than its certain maximum value $H_i^{\max}$, then any delay-tolerant request is fulfilled with a maximum delay Z_i^{max} defined as follows:

$$Z_i^{\max} \triangleq \left\lceil \left(Q_i^{\max} + H_i^{\max}\right)/\epsilon_i \right\rceil. \tag{18}$$

Proof. See Appendix A. □

Given the above property, we can choose the appropriate ϵ_i for data to ensure that it cannot exceed its maximum delay $Z_i^{\max}$ $\left(\text{e.g., } Z_i^{\max} \le \overline{Z_i^{\max}}\right)$ while waiting in a queue to be processed.

Empirical Virtual Queue: $Q_i(\tau)$ plays the role of the Lagrange dual multiplier which iterates at each time slot according to [38]. However, $Q_i(\tau)$ can only update its value according to the actual arrivals and departures at each time slot which takes a lot of time to reach its optimal value. To overcome this problem, we design a virtual empirical queue $\alpha_i(\tau)$ which uses the empirical model state to get its optimal value to accelerate the algorithm. First, we denote that $S(t) = (s_1(t), .., s_M(t))$ represents the input state information which affects the actual queue model $Q_i(\tau)$, while $s_1(t) = (m_{1j}^{(\tau)}, b_j^{(\tau)})$, $s_2(t) = (m_{2j}^{(\tau)}, b_j^{(\tau)})$, etc. We denote that $\boldsymbol{\pi}(t) = (\pi_{s_1}(t), \dots, \pi_{s_N}(t))$, where $\pi_{s_i}(t) = N_{s_i}(t)/t$ and $N_{s_i}(t)$ is the number of slots where $S(t) = s_i$ in $(0, 1, \dots, t-1)$. Note that $\lim_{t\to\infty} \pi_{s_i}(t) = \pi_{s_i}\, w.p.1$.

Using the empirical distribution, we then define the following empirical dual problem as follows:

$$\max: q(\boldsymbol{\alpha}(t), t) \triangleq \sum_{s_i} \pi_{s_i}(t) q_{s_i}(\boldsymbol{\alpha}(t)), \quad \text{s.t.} \quad \boldsymbol{\alpha}(t) \succeq \mathbf{0}. \tag{19}$$

We denote $\boldsymbol{\alpha}(t) = (\alpha_1^*(t), \dots, \alpha_M^*(t))^T$ as an optimal solution vector of (19), and we call this step of obtaining $\boldsymbol{\alpha}(t)$ via (19) dual learning. On the other hand, by its definition, $\boldsymbol{\alpha}(t)$ is time-varying and error-prone, which introduces significant challenges in the analysis of the proposed algorithms. Additionally, $q_{s_i}(\boldsymbol{\alpha})$ is the dual function of the original problem.

4.2. Problem transformation via two-timescale lyapunov optimization

We use Lyapunov function to analyze the $Q_i(\tau)$ and $H_i(\tau)$ first. We concatenate the actual queue $Q_i(\tau)$ and delay-aware virtual queue $H_i(\tau)$ as a vector:

$$\boldsymbol{\Theta}(t) = \left[Q_1(t), \dots, Q_M(t), H_1(t), \dots, H_M(t)\right], \tag{20}$$

and introduce the quadratic Lyapunov function:

$$L(\boldsymbol{\Theta}(t)) \triangleq \frac{1}{2} \sum_{i\in M} \left[Q_i^2(t) + H_i^2(t)\right], \tag{21}$$

as a scalar metric of the congestion level of the system. For example, a small value of $L(\boldsymbol{\Theta}(t))$ implies that all the queue backlogs are small. When any one of the queue backlogs is large, the $L(\boldsymbol{\Theta}(t))$ would be large. Obviously, by pushing the Lyapunov function towards a lower value, we can keep the system stable (all the queue backlogs are bounded to a certain value). Then, we define the T-slot conditional Lyapunov drift as:

$$\Delta_T(\boldsymbol{\Theta}(t)) \triangleq \mathbb{E}[L(\boldsymbol{\Theta}(t+T)) - L(\boldsymbol{\Theta}(t)) \mid \boldsymbol{\Theta}(t)], \tag{22}$$

which measures the difference in the Lyapunov function between two consecutive time frames.

4.3. Joint queue stability and system cost minimization

Intuitively, by minimizing the Lyapunov drift, we can prevent the queue backlogs from unbounded growth and thus preserves system stability. Following the drift-plus-penalty algorithm of the Lyapunov framework, our control algorithm is designed to make decisions on $u_i(t), z_i(\tau), x_{ij}(\tau)$, $x_i(\tau)$ and $\eta_j(\tau)$ at each time slot to minimize the upper bound on the following drift-plus-penalty term every T time slots:

$$\Delta_T(\boldsymbol{\Theta}(t)) + V\mathbb{E}\left\{\sum_{\tau=t}^{t+T-1} \text{Cost}(\tau) \mid \Theta(t)\right\}, \tag{23}$$

where the control parameter $V \ge 0$ is a parameter that represents an importance weight on how much we emphasize the system cost minimization. Such a control parameter can be motivated as follows: We want to make $\Delta_T(\boldsymbol{\Theta}(t))$ small to push queue backlogs towards a lower congestion state. It would force us to consume more resources to transport and process the data, which consequently leads to a high cost. We also want to make the system cost $\mathbb{E}\left\{\sum_{\tau=t}^{t+T-1} \text{Cost}(\tau) \mid \Theta(t)\right\}$ small so that we do not incur a large cost expenditure. Using the Lyapunov optimization technique, we decompose the macro-timescale optimization problem (7) into T-time-slot optimization problems which are much easier to solve. The analytical bound on the drift-plus-penalty term is given in the following Theorem 1.

Theorem 1 (*Drift-Plus-Penalty Bound*)**.** *Let $V > 0, \epsilon_i > 0$ $T \ge 1$ and $t \in kT(k = 0, 1, 2, \dots, K-1), \tau \in [t, t+T-1]$ Given that $x_i(\tau) \le S^{max}$, $Q_i(\tau) \le Q_i^{\max}$ and $H_i(\tau) \le H_i^{\max}$, the drift-plus-penalty is bounded under any possible actions where $B_1 \triangleq [(S^{max})^2 + \frac{1}{2}(u^{max} + z^{max})^2 + \frac{1}{2}\sum_{i\in M} \epsilon_i^2]$:*

$$\begin{aligned}
&\Delta_T(\boldsymbol{\Theta}(t)) + V\mathbb{E}\left\{\sum_{\tau=t}^{t+T-1} \text{Cost}(\tau) \mid \boldsymbol{\Theta}(t)\right\} \\
&\le B_1 T + V\mathbb{E}\left\{\sum_{\tau=t}^{t+T-1} \text{Cost}(\tau) \mid \boldsymbol{\Theta}(t)\right\} \\
&+ \sum_{i\in M} \mathbb{E}\left\{\sum_{\tau=t}^{t+T-1} Q_i(\tau)\left[d_i(\tau) - x_i(\tau)\right] \mid \boldsymbol{\Theta}(t)\right\} \\
&+ \sum_{i\in M} \mathbb{E}\left\{\sum_{\tau=t}^{t+T-1} H_i(\tau)\left[\epsilon_i - x_i(\tau)\right] \mid \boldsymbol{\Theta}(t)\right\}.
\end{aligned} \tag{24}$$

Proof. See Appendix B. □

Theorem 1 shows that the T time slot drift-plus-penalty term is deterministically upper bound. Rather than directly minimize the (23), our strategy actually seeks to minimize the bound given on the right-hand side of the inequality (24). By minimizing every time frame drift plus penalty term, we can find the suboptimal minimizer of the original objective function (8) subject to (9)–(14). Then, we substitute the $\text{Cost}(\tau)$ into the inequality above and try to minimize the right-hand-side of (24), then we can get our optimization objective P1:

$$\begin{aligned}
\min_{\substack{x_i(\tau), x_{ij}(\tau),\\ u_i(t), z_i(\tau), \eta_j}} \; &\sum_{i\in M} \mathbb{E}\left\{\sum_{\tau=t}^{t+T-1} \sum_{j\in N} x_{ij}(\tau)\left[V b_j^{(\tau)} \upsilon log(1/\eta_j(\tau))\right.\right. \\
&\left.\left. + V m_{ij}^{(\tau)} - Q_i(\tau) - H_i(\tau)\right] \mid \boldsymbol{\Theta}(t)\right\} \\
&+ \mathbb{E}\left\{\sum_{\tau=t}^{t+T-1} \sum_{i\in M} V[u_i(t) a_i^{(t)} + z_i(\tau) r_i^{(\tau)}\right. \\
&\left. + \sum_{j\in N} \lambda(\eta_j(\tau) - \eta)^2] \mid \boldsymbol{\Theta}(t)\right\} \\
&s.t. (9)(10)(11)(12)(13)(14)(16).
\end{aligned} \tag{25}$$

4.4. Optimization via queue approximation and empirical virtual queue

Let us look deeper into the problem P1: the future queue backlogs $\boldsymbol{\Theta}(t)$ over the time frame $[t, t+T-1]$ depends on the arrival data $d_i(\tau)$ and the acquired data $x_i(\tau)$. Therefore, we can address the issue of unavailable information by approximating future queue backlog values as the current value, i.e., $Q_i(\tau) = Q_i(t)$ and $H_i(\tau) = H_i(t)$ for all $\tau \in [t, t+T-1]$. However, such approximation loosens the upper bound on the drift-plus-penalty term in (24), Theorem 2 shows the "loosened drift-plus-penalty bound".

Theorem 2 (*Loosened Drift-Plus-Penalty Bound*)**.**: *Let $V > 0, \epsilon_j > 0$ and $T > 1$. Considering Theorem 1 under approximation, the drift-plus-penalty*

term satisfies where $B_2 \triangleq [B_1 + \frac{T-1}{2}(u^{max} + z^{max})^2 + \frac{T-1}{2}\sum_{i\in M}\epsilon_i^2]$:

$$\begin{aligned}
&\Delta_T(\boldsymbol{\Theta}(t)) + V\mathbb{E}\left\{\sum_{\tau=t}^{t+T-1}\text{Cost}(\tau) \mid \boldsymbol{\Theta}(t)\right\} \\
&\quad \le B_2 T + V\mathbb{E}\left\{\sum_{\tau=t}^{t+T-1}\text{Cost}(\tau) \mid \boldsymbol{\Theta}(t)\right\} \\
&\quad + \sum_{i\in M}\mathbb{E}\left\{\sum_{\tau=t}^{t+T-1} Q_i(t)\left[d_i(\tau) - x_i(\tau)\right] \mid \boldsymbol{\Theta}(t)\right\} \\
&\quad + \sum_{i\in M}\mathbb{E}\left\{\sum_{\tau=t}^{t+T-1} H_i(t)\left[\epsilon_i - x_i(\tau)\right] \mid \boldsymbol{\Theta}(t)\right\}.
\end{aligned} \tag{26}$$

Proof. See Appendix C. □

We can find although the above minimize problem is a T-time-slot optimization problem, there is no influence between the decisions of the two consecutive time slots since the queue backlogs $\boldsymbol{\Theta}(t)$ are unchanged over $\tau \in [t, t+T-1]$ in the problem (26). Therefore, we can decompose the T-time-slot problem into multiple simple time-slot optimization problems: at each time slot. We just need to solve a linear program with the variable tuple $[x_i(\tau), x_{ij}(\tau), u_i(t), z_i(\tau), \eta_j(\tau)]$ with constraints (9)–(14),(16) without knowing the future information about the data demanded and the resource prices. So the simple time slot optimization problem is as follows:

$$\begin{aligned}
\min_{\substack{x_i(\tau),x_{ij}(\tau),\\ u_i(t),z_i(\tau),\eta_j(\tau)}} \; & V\,\text{Cost}(\tau) + \sum_{i\in M} Q_i(t)\left[d_i(\tau) - x_i(\tau)\right] \\
& + \sum_{i\in M} H_i(t)\left[\epsilon_i - x_i(\tau)\right] \\
& s.t.(9)(10)(11)(12)(13)(14)(16).
\end{aligned} \tag{27}$$

Next, we add the empirical virtual queue $\alpha_i(t)$ and its bias θ_i into the time slot optimization problem. Bias θ_i are used to adjust the total size of the new queue $Q_i'(t)$. We define that:

$$Q_i'(t) = Q_i(t) + \alpha_i(t) - \theta_i, \tag{28}$$

And the new problem is as follows:

$$\begin{aligned}
\min_{\substack{x_i(\tau),x_{ij}(\tau),\\ u_i(t),z_i(\tau),\eta_j(\tau)}} \; & V\,\text{Cost}(\tau) + \sum_{i\in M} Q_i'(t)\left[d_i(\tau) - x_i(\tau)\right] \\
& + \sum_{i\in M} H_i(t)\left[\epsilon_i - x_i(\tau)\right] \\
& s.t.(9)(10)(11)(12)(13)(14)(16).
\end{aligned} \tag{29}$$

We obtain $\boldsymbol{\alpha}(t)$ through the following equation $q_{s_i}(\boldsymbol{\alpha})$ and (19) which is similar to (23) when exists the actual queue $\boldsymbol{Q}(t)$ because $\boldsymbol{\alpha}(t)$ serves as the empirical virtual queue:

$$\begin{aligned}
q_{s_i}(\boldsymbol{\alpha}) = \max_{\substack{x_i(s_i,\tau),x_{ij}(s_i,\tau),\\ u_i(t),z_i(\tau),\eta_j(\tau)}} \; & V\,\text{Cost}(s_i,\tau) \\
& + \sum_{i\in M} \alpha_i(t)\left[d_i(s_i,\tau) - x_i(s_i,\tau)\right] \\
& + \sum_{i\in M} H_i(t)\left[\epsilon_i - x_i(s_i,\tau)\right] \\
& s.t.(9)(10)(11)(12)(13)(14)(16).
\end{aligned} \tag{30}$$

Finally, we substitute the $\text{Cost}(\tau)$ into the inequality (26), then we can get our optimization objective P2:

$$\begin{aligned}
\min_{\substack{x_i(\tau),x_{ij}(\tau),\\ u_i(t),z_i(\tau),\eta_j}} \; & \sum_{i\in M}\mathbb{E}\left\{\sum_{\tau=t}^{t+T-1}\sum_{j\in N} x_{ij}(\tau)\left[V b_j^{(\tau)} v log(1/\eta_j(\tau))\right.\right. \\
& \left.\left. + V m_{ij}^{(\tau)} - Q_i'(\tau) - H_i(\tau)\right] \mid \boldsymbol{\Theta}(t)\right\} \\
& + \mathbb{E}\left\{\sum_{\tau=t}^{t+T-1}\sum_{i\in M} V[u_i(t)a_i^{(t)} + z_i(\tau)r_i^{(\tau)} \right. \\
& \left. + \sum_{j\in N}\lambda(\eta_j(\tau) - \eta)^2] \mid \boldsymbol{\Theta}(t)\right\} \\
& s.t.(9)(10)(11)(12)(13)(14)(16).
\end{aligned} \tag{31}$$

Noticing that $u_i(t)$ only be determined in $t \in kT(k = 0, 1, \ldots, K-1)$ and for each time slot $\tau \in [t+1, t+T-1]$ we make decisions on $z_i(\tau)$ to meet the current demand. Then we can get our optimization objectives P3 and P4:

$$\begin{aligned}
\min_{\substack{x_i(t),x_{ij}(t),\\ u_i(t),\eta_j(t)}} \; & \sum_{i\in M}\sum_{j\in N} x_{ij}(t)\left[V b_j^{(\tau)} v log(1/\eta_j(t))\right. \\
& \left. + V m_{ij}^{(\tau)} - Q_i'(t) - H_i(t)\right] \\
& + \sum_{i\in M} V\left[u_i(t)a_i^{(t)} + \sum_{j\in N}\lambda(\eta_j(\tau) - \eta)^2\right] \\
& s.t.(9)(10)(11)(12)(13)(14)(16)
\end{aligned} \tag{32}$$

$$\begin{aligned}
\min_{\substack{x_i(\tau),x_{ij}(\tau),\\ z_i(\tau),\eta_j(\tau)}} \; & \sum_{i\in M}\sum_{j\in N} x_{ij}(\tau)\left[V b_j^{(\tau)} v log(1/\eta_j(\tau))\right. \\
& \left. + V m_{ij}^{(\tau)} - Q_i'(t) - H_i(t)\right] \\
& + \sum_{i\in M} V\left[z_i(\tau)r_i^{(\tau)} + \sum_{j\in N}\lambda(\eta_j(\tau) - \eta)^2\right] \\
& s.t.(9)(10)(11)(12)(13)(14)(16).
\end{aligned} \tag{33}$$

Then our proposed $TOEFL$ algorithm is summarized in Algorithm 1.

Algorithm 1: The $TOEFL$ Algorithm

```
// macro-timescale data acquisition decisions.
for k = 0; k < K; k++ do
    Obtain α(t) by solving (19).
    minimize the following problem (32) to obtain
    (x̃_i(t), x̃_ij(t), ũ_i(t), η̃_j(t)).
    Invoke Algorithm 2 to round (x̃_ij(t), x̃_i(t), ũ_i(t), η̃_j(t)) to
    (x̄_ij(t), x̄_i(t), ū_i(t), η̃_j(t)).
    Fix (x̄_i(t), ū_i(t)) and minimize (32) to obtain
    (x*_ij(t), x̄_i(t), ū_i(t), η*_j(t)).
    Invoke Algorithm 2 to round (x*_ij(t), x̄_i(t), ū_i(t), η*_j(t)) to
    (x̄_ij(t), x̄_i(t), ū_i(t), η*_j(t)).
    Update the queue Q_i(t) and H_i(t).
    // micro-timescale data acquisition decisions.
    for τ = kT + 1; τ < (k+1)T; τ++ do
        minimize the following problem (33) to obtain
        (x̃_i(τ), x̃_ij(τ), z̃_i(τ), η̃_j(τ)).
        Invoke Algorithm 2 to round (x̃_ij(τ), x̃_i(τ), z̃_i(τ), η̃_j(τ)) to
        (x̄_ij(τ), x̄_i(τ), z̄_i(τ), η̃_j(τ)).
        Fix (x̄_i(τ), z̄_i(τ)) and minimize (33) to obtain
        (x*_ij(τ), x̄_i(τ), z̄_i(τ), η*_j(τ)).
        Invoke Algorithm 2 to round (x*_ij(τ), x̄_i(τ), z̄_i(τ), η*_j(τ)) to
        (x̄_ij(τ), x̄_i(τ), z̄_i(τ), η*_j(τ)).
        Update the queue Q_i(τ) and H_i(τ).
    end for
end for
```

Algorithm 1 is executed in the following steps. At the macro-timeslot $t \in kT(k = 0, 1, \ldots, K-1)$, the system makes macro-timescale data acquisition decisions: First, the optimal value of Lagrange multiplier of the empirical virtual queue obtained in problem (19) is used as the input value and substituted into the minimization problem (32) for the solution. Second, in order to convert the fractional decisions such as $x_{ij}(\tau), x_i(\tau), z_i(\tau), u_i(t)$ into integers, we propose Algorithm 2, which is described later. Note that after we get a set of rounding solutions,

we need to fix the value of the set of solutions before continuing to solve the original minimization problem. Finally, the solution to the optimization problem is used to update the values of $Q_i(t)$ and $H_i(t)$. At micro-timeslot $\tau \in [t+1, t+T-1]$, the system makes micro-timescale data acquisition decisions. The remaining steps are similar to the above.

4.5. Randomized rounding with weight algorithm

From Algorithm 1 we can see that since some variables such as $x_{ij}(\tau), x_i(\tau), z_i(\tau), u_i(t)$ are integer while the first step of Algorithm 1 only provides the non-integer solutions for them. Therefore, We propose the *RRW* algorithm to round the non-integer results into integers while achieving a better expectation for the minimized formulation and violating none of our problem's constraints. This algorithm is summarized in Algorithm 2.

Algorithm 2: The *RRW* Algorithm

1: INPUT: Fractional decision such as $\widetilde{x}_{ij}(\tau), \widetilde{x}_i(\tau), \widetilde{z}_i(\tau), \widetilde{u}_i(t)$. We use I_t to present the variables.
2: // For any variables, we define the integer part as $\boldsymbol{X}_t$ and the non integer part as $\boldsymbol{U}_t$. Ensure the sum of all columns in $\boldsymbol{U}_t$ is an integer.
3: $\boldsymbol{U}_t = [\widetilde{\boldsymbol{I}}_{\boldsymbol{1},t}, ...\widetilde{\boldsymbol{I}}_{\boldsymbol{N},t}]^{\boldsymbol{T}}$.
4: $k = \mathbf{1}^{\top}\mathbf{U}_t, \gamma_1 = 1-(k-\lfloor k \rfloor)/k$, $\gamma_2 = 1+(\lceil k \rceil - k)/k$.
5:
$$\mathbf{V}_t^{\top} = \begin{cases} [\gamma_1 U_{1,t}, \ldots, \gamma_1 U_{N,t}] & \text{with prob. } \lceil k \rceil - k \\ [\gamma_2 U_{1,t}, \ldots, \gamma_2 U_{N,t}] & \text{with prob. } k - \lfloor k \rfloor \end{cases}$$
6: // Ensure each column of $\boldsymbol{V}_t$ is an integer.
7: **while** $V_{i,t}$ and $V_{j,t}$ are not integer **do**
8: Assume a as the weight of $V_{i,t}$ and b as the weight of $V_{j,t}$.
9: $\theta_1 = \min\left\{\lceil V_{i,t} \rceil - V_{i,t}, V_{j,t} - \lfloor V_{j,t} \rfloor\right\}, \theta_2 = \min\left\{V_{i,t} - \lfloor V_{i,t} \rfloor, \lceil V_{j,t} \rceil - V_{j,t}\right\}$
10:
$$\left(V_{i,t}, V_{j,t}\right) = \begin{cases} \left(V_{i,t}+\theta_1, V_{j,t}-\theta_1\right) & \text{with prob. } \frac{b\theta_2}{a\theta_1+b\theta_2} \\ \left(V_{i,t}-\theta_2, V_{j,t}+\theta_2\right) & \text{with prob. } \frac{a\theta_1}{a\theta_1+b\theta_2} \end{cases}$$
11: **end while**
12: Return $I_t = [V_{1,t}, ..., V_{N,t}, X_t]$.

Algorithm 2 is executed in the following steps. First, we adjust $\boldsymbol{U}_t$ in a randomized manner, ensuring that the sum of the adjusted values equals an integer. It increases each column of $\boldsymbol{U}_t$ by multiplying $\gamma_2 \geq 0$ or decreases each column of $\boldsymbol{U}_t$ by multiplying $\gamma_1 \leq 0$. The probabilities of taking these two decisions are $k - \lfloor k \rfloor$ and $\lceil k \rceil - k$ respectively, which can ensure the expectation of each adjusted value equals its corresponding value before such adjustment. Second, we round $\boldsymbol{V}_t$ into integers in a randomized manner. We use the values of $\boldsymbol{V}_t$ and its weight as the probability of rounding the columns in pairs into integers while letting the two fractions compensate each other. In this way, we can guarantee that the sum of all the values stays unchanged after rounding and each value is an integer. The complexity of the algorithm reaches $O(N^2)$.

5. Performance analysis

In this section, we present the detailed performance results and some preliminaries for our proposed *TOEFL* algorithm.

Theorem 3 (*The Maximum Acceptable Value of V*)**.** *Given any fixed control parameter $V, \epsilon_i > 0$, our algorithm achieves:*

(1) While $S^{max} \geq u^{max} + z^{max}$, the delay-tolerant queue $Q_i(\tau)$ and delay-aware queue $H_i(\tau)$ is bounded as:

$$\begin{aligned} Q_i^{\max} &\triangleq VCost_{comp}^{\max} + u^{max} + z^{max} \\ H_i^{\max} &\triangleq VCost_{comp}^{\max} + \epsilon^{\max}. \end{aligned} \tag{34}$$

(2) The maximum acceptable value of V is:

$$V^{\max} = \frac{\epsilon_i \overline{Z_i^{\max}} - \epsilon^{\max} - u^{max} + z^{max}}{2Cost_{comp}^{\max}}. \tag{35}$$

Proof. See Appendix E. □

Assumption 1. There exists a constant $\epsilon_s = \Theta(1) > 0$ such that for any valid state distribution $\boldsymbol{\pi}' = \left(\pi'_{s_1}, \ldots, \pi'_{s_M}\right)$ with $\left\|\boldsymbol{\pi}' - \boldsymbol{\pi}\right\| \leq \epsilon_s$ for all s_i (possibly depending on π') such that:

$$\sum_{s_i} \pi_{s_i} \left[d_i\left(s_i, \tau\right) - x_i\left(s_i, \tau\right)\right] \leq -\eta_0, \tag{36}$$

where $\eta_0 = \Theta(1) > 0$ is independent of $\boldsymbol{\pi}'$.

Assumption 2. γ_0^* is the unique optimal solution of $q_0(\gamma)$ in $\mathbb{R}^r$.

Lemma 2. *There exists an $O(1)$ time $T_{\epsilon_s} < \infty$, such that with probability 1, for all $t > T_{\epsilon_s}$,*

$$\sum_i \alpha_i(t) \leq \xi \triangleq \frac{VCost_{max}}{\eta_0}. \tag{37}$$

With the above bound, we have the following corollary, which shows the convergence of $q(\boldsymbol{\alpha}, t)$ to $q(\boldsymbol{\alpha})$.

Proof. See Appendix F. □

Corollary 1. *With probability 1, for all $t > T_{\epsilon_s}$ (here T_{ϵ_s} is defined in Lemma 2), the function $q(\boldsymbol{\alpha}, t)$ satisfies:*

$$|q(\boldsymbol{\alpha}, t) - q(\boldsymbol{\alpha})| \leq max|\delta_{s_i}(t)| M(VCost_{max} + r\xi B). \tag{38}$$

Proof. See Appendix G. □

Lemma 3. $\lim_{t\to\infty} \boldsymbol{\alpha}(t) = \boldsymbol{\gamma}^* w.p.1.$

Proof. See Appendix H. □

For easy understanding and not to confuse, we use $\boldsymbol{\gamma}(t)$ to represent the $Q(t)$ as the Lagrange dual multiplier of the problem (19) and its optimal value is $\boldsymbol{\alpha}^*$. Lemma 3 thus shows that although such a condition can appear, it is simply a transient phenomenon and $\boldsymbol{\gamma}(t)$ will eventually converge to $\boldsymbol{\gamma}^*$.

Lemma 4. *There exists a constant $D_p \triangleq \frac{B-\eta^2}{2(\rho-\eta)}$ with $\eta < \rho$. If $\|\boldsymbol{Q}(t) - \tilde{\boldsymbol{\gamma}}^*(t)\| \geq D_p$, then $\|\boldsymbol{Q}(t+1) - \tilde{\boldsymbol{\gamma}}^*(t)\| \leq \|\boldsymbol{Q}(t) - \tilde{\boldsymbol{\gamma}}^*(t)\| - \eta$.*

Proof. See Appendix I. □

Theorem 4 (*Performance of the Delay*)**.**

$$\bar{Q}_{avg} = \sum_i \bar{Q}_i = \sum_i \theta_i + \Theta(1). \tag{39}$$

Theorem 5 shows that by using the empirical virtual queue $\alpha(t)$, one can guarantee the average actual queue size is roughly $\sum_i \theta_i$. Hence, by choosing $\theta_i = O([log(V)]^2)$, we can get the better delay performance.

Proof. See Appendix J. □

Theorem 5 (*Performance of the Cost*)**.** *Suppose the conditions in Theorem 5 hold. Then, with a sufficiently large V and $\theta_i = O([log(V)]^2)$ for all i, we*

have:

$$Cost_{av}^{RRW} \leq Cost^* + \frac{B_2}{V} + O(\frac{1}{V}) = Cost^* + O(\frac{1}{V}). \quad (40)$$

Theorems 4 and 5 together show that our proposed algorithm can achieve the $[O(1/V), O([log(V)]^2)]$ cost-delay tradeoff for our problem. We use the FIFO queueing discipline and do not require any pre-learning phase, which is suitable for practical implementations.

Proof. See Appendix K. □

Theorem 6 (*Performance of the RRW Algorithm*)**.** *Assume the formulation as a simple function $f(V_1, V_2) = aV_1 + bV_2$, while a, b are the weight of variables. V_1' and V_2' are the results after using the RRW algorithm. Our algorithm achieves:*

(1) while $a \geq b$,

$$\mathbb{E}[aV_1' + bV_2'] \leq aV_1 + bV_2. \quad (41)$$

It implies $\mathbb{E}[Cost^{RRW}] \leq Cost^*$ which shows the performance of the original function after using the RRW algorithm.

(2)

$$\mathbb{E}[V_1' + V_2'] = V_1 + V_2. \quad (42)$$

It means after using the RRW algorithm, our original conditions remain unchanged.

Proof. See Appendix D. □

6. Experimental study

6.1. Experimental settings

Control Algorithms: We implement and compare multiple approaches that minimize the total cost and control federated learning over time: (1) $TOEFL$, our online approach in this paper; (2) Traditional two-timescale Lyapunov algorithm (TL), the general approach to deal with different time scale problems; (3) Traditional two-timescale Lyapunov algorithm with RRW ($TL + RRW$). We use RRW to deal with the NP-hard problems in two-timescale Lyapunov techniques and improve its performance; (4) $Random$. The device system decides the data acquisition and which edge server to upload the data randomly; (5) $Greedy$. The device system compares the data acquisition price in the macro-timescale manner and the micro-timescale manner, and then chooses the cheaper one to acquire; (6) $Offline$. We assume that the offline algorithm can obtain all the system information and then make the best decision for the data acquisition in macro-timescale manner and micro-timescale manner.

Training Data and Tasks: We utilize a federated version of MINIST [39] that has a version of the original NIST dataset that has been re-processed using Leaf so that the data is keyed by the original writer of the digits. Since each writer has a unique style, the dataset shows the kind of non-i.i.d behavior expected of federated datasets, which is more suitable for our experiment. Also, we utilize an i.i.d version of Minist for comparison, in order to verify the validity of our algorithms under i.i.d dataset settings.

FL-CNN: This task trains Convolution Neural Network (CNN) federated model. It consists of the following structure: two $5 * 5$ convolution layers (the first with 32 channels, the second with 64 channels, each followed with $2 * 2$ max pooling), a fully connected layer with 512 units and ReLu activation, and a final softmax output layer (1,663,370 total parameters)

FL-MLP: This task trains Multi-Layer Perception (MLP) federated model. It consists of the following structure: two fully connected layers with 128, 64 units respectively and ReLu activation, and a final softmax output layer.

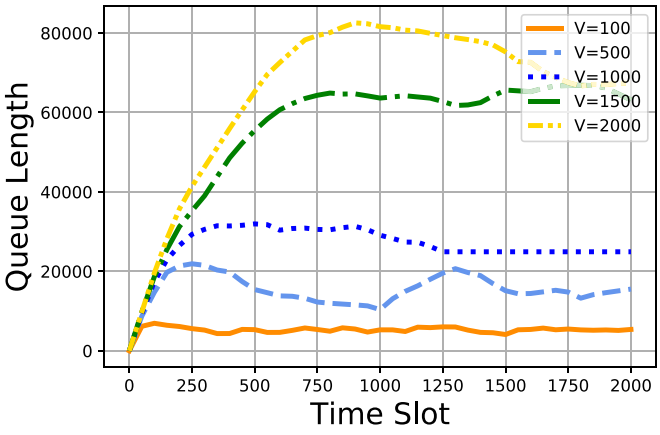

Fig. 3. Queue length convergence under different values of V.

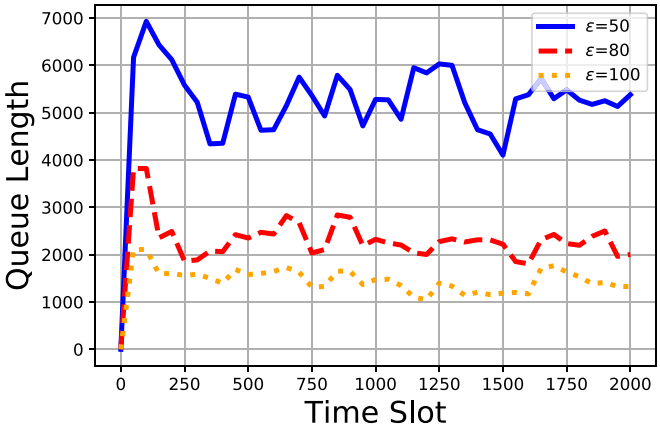

Fig. 4. Queue length convergence under different values of ϵ.

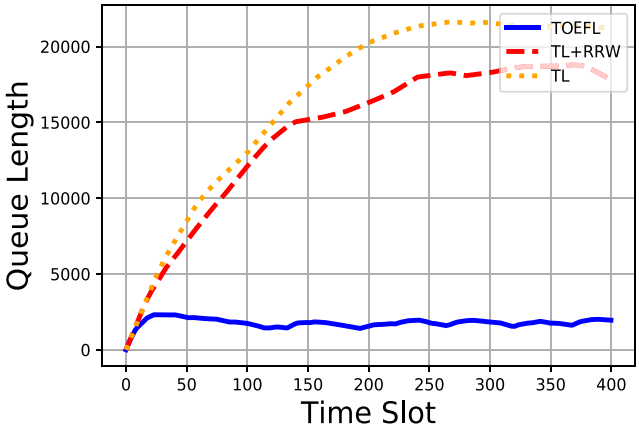

Fig. 5. Queue length convergence under different control algorithms.

FL-MLR: This task trains the Multinomial Logistic Regression (MLP) federated model. It consists of a fully connected layer with 64 units and a sigmoid output layer.

Here we note that we are not doing any fine tuning of the hyperparameters of the tasks, so the inference accuracy and loss can achieve better performance if one continues to tune such hyperparameters.

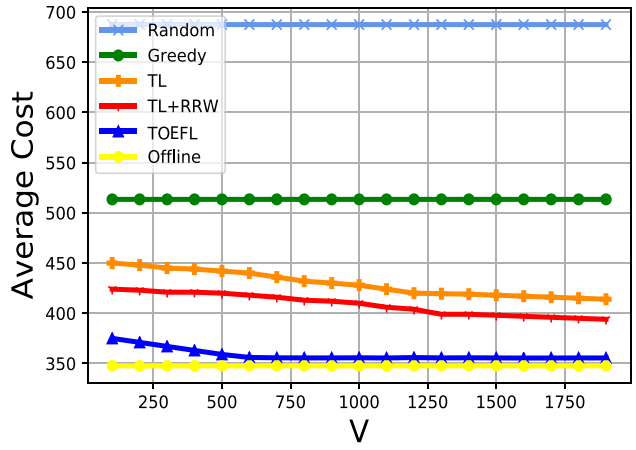

Fig. 6. Average Cost under different values of V.

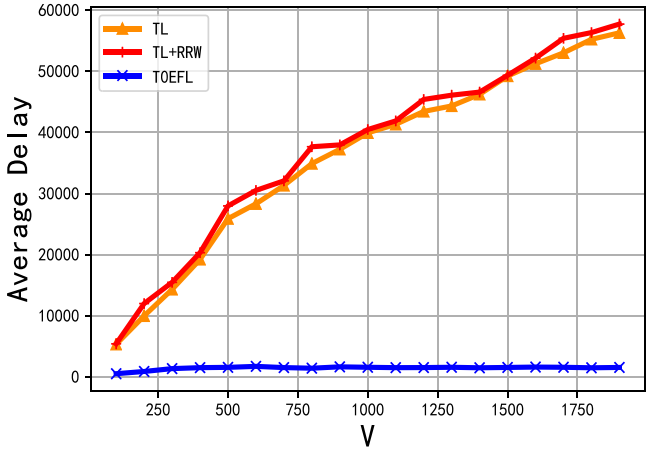

Fig. 7. Average delay under different values of V.

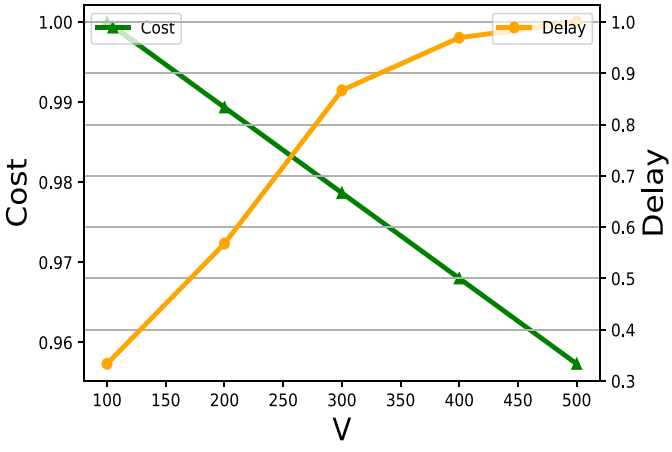

Fig. 8. Cost-delay tradeoff under different values of V.

Simulation Setup: We assume that $a_i^{(t)}$ and $r_i^{(\tau)}$ are uniformly and randomly distributed with an order $\mathbb{E}[r_i^{(\tau)}] > \mathbb{E}[a_i^{(t)}]$ where $\mathbb{E}[r_i^{(\tau)}] = 0.5$ and $\mathbb{E}[a_i^{(t)}] = 0.2$. We set the 60 micro timescales in a macro timescale in these experiments. Also we set the inputs' value as follows: $M = 5, N = 5, \lambda = 100, \eta = 0.1, l = 0.1, \mathbb{E}[R^{(\tau)}] = 170, \mathbb{E}[m_{ij}^{(\tau)}] = 0.55, \mathbb{E}[b_j^{(\tau)}] = 0.55$.

6.2. Experimental results

Queue Length Convergence: Fig. 3 shows that the impact of different control parameters V on queue length convergence under TL. With the increase of V, the maximum queue length increases and the convergence time of the algorithm becomes longer, which is in line with the control purpose of parameter V.

Fig. 4 shows the impact of different ϵ on queue length convergence. ϵ is the delay-aware parameter used in the delay-aware virtual queue equation [(17)]. The higher value of ϵ makes the maximum queue length become smaller and converge faster.

Fig. 5 presents the impact of different control algorithms on queue length convergence. We set the ideal state of $V = 250, \epsilon = 100$, which is actually consistent with the previous analysis, and setting other values can get similar results. In terms of queue length, $TOEFL$ beats TL and $TL + RRW$, reducing by 89.1% and 87.9% average queue length respectively. In terms of convergence time, $TOEFL$ beats TL and $TL+RRW$, reducing by 88.5% and 86.9% convergence time respectively. Under our proposed $TOEFL$, the actual queue size converges very quickly and has a lower delay and stable convergence situation with the help of an empirical virtual queue.

The Impact of Control Parameter V: Fig. 6 shows that our proposed $TOEFL$ significantly outperforms the other control algorithms in terms of the average cost, and it is close to the optimal performance. Especially, our proposed $TOEFL$ can save nearly 0.83 times, 0.61 times, 0.34 times and 0.23 times average cost compared with $Random$, $Greedy$, TL and $TL + RRW$, respectively. What is more, with the help of RRW algorithm, $TL + RRW$ has a lower average cost compared to TL. $TL + RRW$ is up to 6.2% lower than TL in terms of average cost, which actually proves the performance of the RRW algorithm. Fig. 7 also validates the performance of our proposed $TOEFL$ in terms of delay. It can have a lower average cost while keeping a lower average delay.

In conclusion, Fig. 8 shows the impact of control Parameter V and its tradeoff between cost and delay. As V increases, the delay becomes larger and the cost becomes smaller. Such change trend is consistent with the $[O(1/V), O([log(V)]^2)]$ cost-delay tradeoff.

Quality of Federated Model: Fig. 9 shows the model's accuracy and loss of different control algorithms. We can observe that $TOEFL$ converges in 18 rounds, which is earlier than 20 rounds of TL. And $TOEFL$ can achieve 95.3% accuracy, which is higher than 86.3% accuracy of TL. Our proposed $TOEFL$ can achieve higher accuracy and lower loss in fewer communication rounds, which is similar to the offline optimal algorithm results. The offline optimal algorithm here means the data exists in the edge servers in the beginning. Because of the empirical virtual queue and the delay-aware virtual queue, our data can reach the edge servers with lower delay and can be trained as fast as possible.

For non-i.i.d data settings, Fig. 10 validates the robustness and applicability of $TOEFL$ in the heterogeneous federated learning environment for the training tasks of FL-CNN, FL-MLP, and FL-MLR. The results show a good convergence trend, even in the non-IID setting. It indicates that $TOEFL$ can always control federated learning to achieve satisfiable inference accuracy and loss. Fig. 11 also shows the validity of our $TOEFL$ under i.i.d data settings.

7. Conclusion

In this paper, in order to coordinate time-varying data acquisition in macro-timescale and micro-timescale manners with minimal cost in federated learning, we propose the $TOEFL$ algorithm, which has the empirical virtual queue that learns empirical state information of the system in order to accelerate the convergence speed while

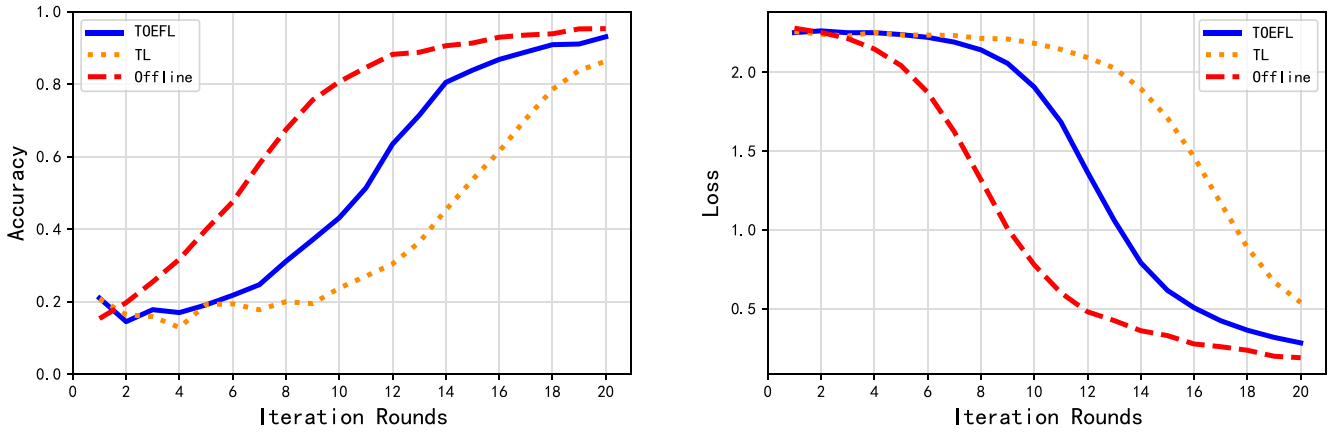

Fig. 9. Accuracy and loss of different control algorithms.

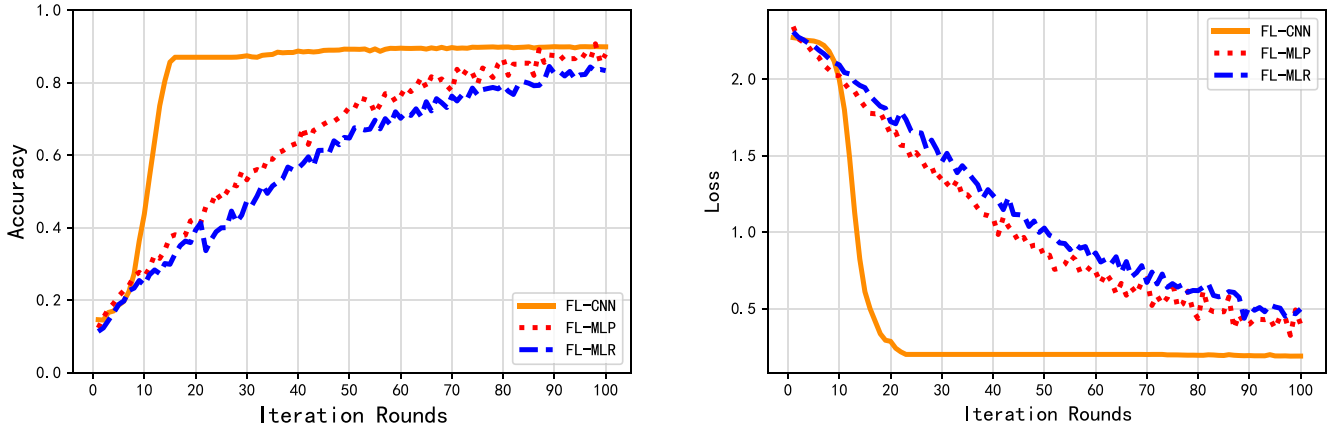

Fig. 10. Accuracy and loss in the non-i.i.d setting.

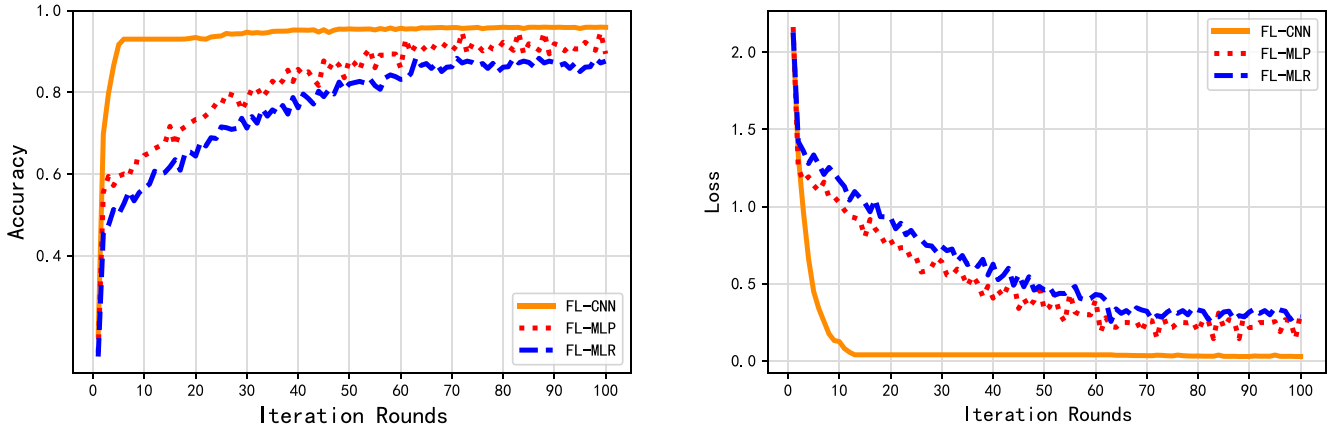

Fig. 11. Accuracy and loss in the i.i.d setting.

keeping the low cost and high federated models' quality. We prove the performance guarantees for the proposed algorithm which can achieve $[O(1/V), O([log(V)]^2)]$ cost-delay tradeoff, which significantly improves the $[O(1/V), O(V)]$ cost-delay tradeoff of the traditional Lyapunov technique. We also propose a novel *RRW* algorithm that solves the NP-hard problem in the Lyapunov function with low time complexity and competitive performance and proves its performance. We have carried out extensive experiments and demonstrated the results to validate the practical performance of our proposed approach.

CRediT authorship contribution statement

Konglin Zhu: Conceptualization, Methodology, Writing – original draft, Software. **Wentao Chen:** Formal analysis, Data curation, Writing – original draft, Investigation. **Lei Jiao:** Conceptualization, Methodology, Writing – review & editing. **Jiaxing Wang:** Investigation, Validation, Writing – review & editing. **Yuyang Peng:** Validation, Writing – review & editing. **Lin Zhang:** Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Appendix A. Proof of Lemma 1

Fix a slot t. We show that all the data $d_i(t)$ can be acquired on or before slot t + Z_i^{max}. Suppose this is not true. We reach a contradiction. Note by (15) that arrivals $d_i(t)$ are added to the queue backlog $Q_i(t+1)$ and are first available for service on slot t + 1. Therefore, by (17), we have for all slots $\tau \in [t+1, t+Z_i^{max}]$:

$$H_i(\tau+1) \le H_i(\tau) - x_i(\tau) + \epsilon_i. \tag{A.1}$$

Summing the above $\tau \in [t+1, t+Z_i^{max}]$ gets:

$$H_i\left(t+Z_i^{\max}+1\right) - H_i(t+1) \ge -\sum_{\tau=t+1}^{t+Z_i^{\max}} x_i(\tau) + Z_i^{\max}.\epsilon_i. \tag{A.2}$$

Rearranging terms above and using the fact that $H_i\left(t+Z_i^{\max}+1\right) \le H_i^{max}$ and $H_i(t+1) \ge 0$ gets:

$$Z_i^{\max}\epsilon_i \le \sum_{\tau=t+1}^{t+Z_i^{\max}} x_i(\tau) + H_i^{\max}. \tag{A.3}$$

On the other hand, the sum of $x_i(\tau)$ over the interval $\tau = [t+1, t+Z_j^{max}]$ must be strictly less than $Q_i(t+1)$ (else, by the first-in-first-out (FIFO) rule, any arrival $d_i(t)$, which is included at the end of the backlog $Q_i(t+1)$, would have been cleared during this interval). Thus:

$$\sum_{\tau=t+1}^{t+Z_i^{\max}} x_j(\tau) < Q_i(t+1) \le Q_i^{\max}. \tag{A.4}$$

Combining the above two equations together:

$$\begin{aligned} Z_i^{max}\epsilon_i &< Q_i^{max} + H_i^{max} \\ Z_i^{max} &< (Q_i^{max} + H_i^{max})/\epsilon_i. \end{aligned} \tag{A.5}$$

This contradicts the definition of Z_i^{max}. So the arrival $d_i(t)$ can be served on or before $t + Z_i^{max}$.

Appendix B. Proof of Theorem 1

Squaring the queue update Eq. (15) :

$$\begin{aligned} Q_i^2(\tau+1) &\le [Q_i(\tau) - x_i(\tau)]^2 + d_i(\tau)^2 \\ &\quad + 2[Q_i(\tau) - x_i(\tau)]d_i(\tau) \\ &\le [Q_i(\tau) - x_i(\tau)]^2 + d_i(\tau)^2 \\ &\quad + 2Q_i(\tau)d_i(\tau). \end{aligned} \tag{B.1}$$

Therefore:

$$\begin{aligned} Q_i^2(\tau+1) - Q_i^2(\tau) &\le x_i(\tau)^2 + d_i(\tau)^2 \\ &\quad + 2Q_i(\tau)[d_i(\tau) - x_i(\tau)]. \end{aligned} \tag{B.2}$$

Because $x_i(\tau) \le S^{max}$, $d_i(\tau) = u_i(t) + z_i(\tau)$, $u_i(t) \le u^{max}$, $z_i(\tau) \le z^{max}$, then

$$\begin{aligned} Q_i^2(\tau+1) - Q_i^2(\tau) &\le (S^{max})^2 + (u^{max} + z^{max})^2 \\ &\quad + 2Q_i(\tau)[d_i(\tau) - x_i(\tau)]. \end{aligned} \tag{B.3}$$

Similarly, for virtual queues $H_i(\tau)$, we have:

$$H_i^2(\tau+1) - H_i^2(\tau) \le \left(S^{\max}\right)^2 + \epsilon_i^2 + 2H_i(\tau)[\epsilon_i - x_i(\tau)]. \tag{B.4}$$

Applying (B.3) and (B.4) to (23) we get:

$$\begin{aligned} &\Delta_T(\Theta(t)) + V\mathbb{E}\left\{\sum_{\tau=t}^{t+T-1} \mathrm{Cost}(\tau) \mid \boldsymbol{\Theta}(t)\right\} \\ &= \mathbb{E}[L(\boldsymbol{\Theta}(t+T)) - L(\boldsymbol{\Theta}(t)) \mid \Theta(t)] \\ &\quad + V\mathbb{E}\left\{\sum_{\tau=t}^{t+T-1} \mathrm{Cost}(\tau) \mid \boldsymbol{\Theta}(t)\right\} \\ &= \frac{1}{2}\sum_{\tau=t}^{t+T-1}\sum_{j\in N}\left[Q_i^2(t+1) - Q_i^2(t) + H_i^2(t+1) - H_i^2(t)\right] \\ &\quad + V\mathbb{E}\left\{\sum_{\tau=t}^{t+T-1} \mathrm{Cost}(\tau) \mid \boldsymbol{\Theta}(t)\right\} \\ &\le V\mathbb{E}\left\{\sum_{\tau=t}^{t+T-1} \mathrm{Cost}(\tau) \mid \boldsymbol{\Theta}(t)\right\} \\ &\quad + \frac{1}{2}T[2(S^{max})^2 + (u^{max} + z^{max})^2 + \sum_{i\in M}\epsilon_i^2] \\ &\quad + \sum_{\tau=t}^{t+T-1}\sum_{i\in M} Q_i(\tau)[d_i(\tau) - x_i(\tau)] \\ &\quad + \sum_{\tau=t}^{t+T-1}\sum_{i\in M} H_i(\tau)[\epsilon_i - x_i(\tau)]. \end{aligned} \tag{B.5}$$

Simplifying the above inequality leads to Theorem 1:

$$\begin{aligned} &\Delta_T(\Theta(t)) + V\mathbb{E}\left\{\sum_{\tau=t}^{t+T-1} \mathrm{Cost}(\tau) \mid \boldsymbol{\Theta}(t)\right\} \\ &\quad \le B_1 T + V\mathbb{E}\left\{\sum_{\tau=t}^{t+T-1} \mathrm{Cost}(\tau) \mid \boldsymbol{\Theta}(t)\right\} \\ &\quad + \sum_{i\in M}\mathbb{E}\left\{\sum_{\tau=t}^{t+T-1} Q_i(\tau)\left[d_i(\tau) - x_i(\tau)\right] \mid \boldsymbol{\Theta}(t)\right\} \\ &\quad + \sum_{i\in M}\mathbb{E}\left\{\sum_{\tau=t}^{t+T-1} H_i(\tau)\left[\epsilon_i - x_i(\tau)\right] \mid \boldsymbol{\Theta}(t)\right\}, \end{aligned} \tag{B.6}$$

where $B_1 \triangleq [(S^{max})^2 + \frac{1}{2}(u^{max} + z^{max})^2 + \frac{1}{2}\sum_{i\in M}\epsilon_i^2]$.

Appendix C. Proof of Theorem 2

For any $\tau \in [t, t+T-1]$, we can get:

$$\begin{aligned} Q_i(\tau) &\le Q_i(t) + (\tau - t)(u^{\max} + z^{\max}) \\ H_i(\tau) &\le H_i(t) + (\tau - t)\epsilon_i. \end{aligned} \tag{C.1}$$

Then, applying the first inequality to $Q_i(\tau)$, we get:

$$\begin{aligned} &\sum_{\tau=t}^{t+T-1} Q_i(\tau)[d_i(\tau) - x_i(\tau)] \\ &\le \sum_{\tau=t}^{t+T-1} [Q_i(t) + (\tau - t)(u^{\max} + z^{\max})][d_i(\tau) - x_i(\tau)] \\ &\le \sum_{\tau=t}^{t+T-1} Q_i(t)[d_i(\tau) - x_i(\tau)] + (\tau - t)(u^{\max} + z^{\max})^2. \end{aligned} \tag{C.2}$$

Similarly, we apply the second inequality to $H_i(\tau)$, we get:

$$\begin{aligned} &\sum_{\tau=t}^{t+T-1} H_i(\tau)[\epsilon_i - x_i(\tau)] \\ &\le \sum_{\tau=t}^{t+T-1} H_i(t)[\epsilon_i - x_i(\tau)] + (\tau - t){\epsilon_i}^2. \end{aligned} \tag{C.3}$$

Substituting the above inequalities into (24) and then summing up through time slot $\tau \in [t, t+T-1]$ leads to (26).

Appendix D. Proof of Theorem 3

Assume the formulation as a simple way $f(V_1, V_2) = aV_1 + bV_2$, while a, b are the weight of variables. V_1' and V_2' are the results after using the RRW Algorithm. Because our problem is a minimized problem, for the variables with bigger weight, we hope its randomized rounding results have a higher probability of becoming smaller and vice versa. So we design the algorithm with weight that can achieve better performance.

(1)

$$\begin{aligned}
&\mathbb{E}[aV_1' + bV_2'] \\
&= a\mathbb{E}(V_1') + b\mathbb{E}(V_2') \\
&= aV_1 + bV_2 + a[\frac{\theta_1\theta_2 b}{\theta_1 a + \theta_2 b} - \frac{\theta_1\theta_2 a}{\theta_1 a + \theta_2 b}] \\
&+ b[\frac{\theta_1\theta_2 a}{\theta_1 a + \theta_2 b} - \frac{\theta_1\theta_2 b}{\theta_1 a + \theta_2 b}] \\
&= aV_1 + bV_2 + \frac{\theta_1\theta_2(2ab - a^2 - b^2)}{\theta_1 a + \theta_2 b} \\
&= aV_1 + bV_2 - \frac{\theta_1\theta_2(a-b)^2}{\theta_1 a + \theta_2 b} \\
&\le aV_1 + bV_2.
\end{aligned} \tag{D.1}$$

And so on, while using the RRW Algorithm, we can achieve $E[Cost^{RRW}] \le Cost^*$. Here $Cost^{RRW}$ denotes the cost under Randomized Rounding with weight Algorithm, and $Cost^*$ denotes the optimal cost while relaxing the original problem.

(2)

$$\begin{aligned}
&\mathbb{E}[V_1' + V_2'] \\
&= \frac{\theta_2 b}{\theta_1 a + \theta_2 b}[(V_1 + \theta_1) + (V_2 - \theta_1)] \\
&+ \frac{\theta_1 a}{\theta_1 a + \theta_2 b}[(V_1 - \theta_2) + (V_2 + \theta_2)] \\
&= V_1 + V_2.
\end{aligned} \tag{D.2}$$

Appendix E. Proof of Theorem 4

(1) We use mathematical induction to prove this conclusion. Obviously, $Q_i(0) = 0 < Q_i^{max}$. Now assume that $Q_i(\tau) \le Q_i^{max} = VCost_{comp}^{max} + d_l^{max} + d_r^{max}$, then we need to prove $Q_i(\tau+1) \le Q_i^{max}$.

In the first case $Q_i(\tau) \le VCost_{comp}^{max}$:

$$\begin{aligned}
Q_i(\tau+1) &= Q_i(\tau) - x_i(\tau) + d_i(\tau) \\
&\le Q_i(\tau) + u^{max} + z^{max} \\
&\le VCost_{comp}^{max} + u^{max} + z^{max} = Q_i^{max}.
\end{aligned} \tag{E.1}$$

In the second case $VCost_{comp}^{max} \le Q_i(\tau) \le VCost_{comp}^{max} + u^{max} + z^{max}$:

Observe the optimization term: $x_{ij}(\tau)\left[Vb_j^{(\tau)}vlog(1/\eta_j(\tau)) + Vm_{ij}^{(\tau)} - Q_i(\tau) - H_i(\tau)\right]$. In this case, the term $Vb_j^{(\tau)}vlog(1/\eta_j(\tau)) + Vm_{ij}^{(\tau)} - Q_i(\tau) - H_i(\tau) \le 0$. Consequently, in order to minimize the cost, the variable $x_i(t)$ may will the maximum value S^{max} or $Q_i(\tau)$.

If $Q_i(\tau) > S^{max}$, then,

$$\begin{aligned}
Q_i(\tau+1) &= Q_i(\tau) - x_i(\tau) + d_i(\tau) \\
&\le Q_i(\tau) - S^{max} + d_l^{max} + d_r^{max}.
\end{aligned} \tag{E.2}$$

While $S^{max} \ge u^{max} + z^{max}$, $Q_i(\tau+1) \le Q_i(\tau)$. That means the queue will not increase in this situation. So $Q_i(\tau+1) \le Q_i(\tau) \le Q_i^{max}$.

If $Q_i(\tau) \le S^{max}$, because of the constraint $x_i(\tau) \le Q_i(\tau)$, $x_i(\tau) = Q_i(\tau)$, then $Q_i(\tau+1) \le d_i(\tau) \le u^{max} + z^{max} \le Q_j^{max}$.

In conclusion, $Q_i^{max} \triangleq VCost_{comp}^{max} + u^{max} + z^{max}$.

Using the same way we can prove $H_i^{\max} \triangleq VCost_{comp}^{\max} + \epsilon^{\max}$.

(2) From Theorems 5 and 6, we can know our algorithm has the $[O(1/V), O([log(V)]^2)]$ cost-delay trade-off. While getting lower overall cost by increasing V, we can only get larger queue backlogs which is unreasonable. So we use constraint (13) to guarantee the data should be served within the maximum delay $\overline{Z_i^{\max}}$, which results in the limitation of V as below:

$$\begin{aligned}
Z_i^{max} &\le \frac{Q_i^{max} + H_i^{max}}{\epsilon_i} \\
&= \frac{2VCost_{comp}^{max} + u^{max} + z^{max} + \epsilon^{max}}{\epsilon_i} \le \overline{Z_i^{\max}}.
\end{aligned} \tag{E.3}$$

Consequently, we can get:

$$V^{\max} = \frac{\epsilon_i \overline{Z_i^{\max}} - \epsilon^{\max} - u^{max} - z^{max}}{2Cost_{comp}^{\max}}. \tag{E.4}$$

Appendix F. Proof of Lemma 2

First we see that $\pi_{s_i}(t)$ converges to π_{s_i} as t goes to infinity with probability 1 . Consequently, there exists a time $T_{\epsilon_s} < \infty$, such that $\|\pi(t) - \pi\| \le \epsilon_s$ for all $t \ge T_{\epsilon_s}$ with probability 1. This implies that $\alpha(t) < \infty$ for all $t \ge T_{\epsilon_s}$.

We now construct a fictitious system, which is exactly the same as our system, except that we replace the input state distribution π by $\boldsymbol{\pi}(t)$. From Assumption 1, we know that for any $t \ge T_{\epsilon_s}, w.p.1$, this system admits an optimal control policy that achieves the optimal cost (with the state distribution being $\boldsymbol{\pi}(t)$) and ensures the queue stability. We denote $Cost_{\text{av}}^*(\boldsymbol{\pi}(t))$ the optimal average cost of the fictitious system subject to stability. We see then $Cost_{\text{av}}^*(\boldsymbol{\pi}(t)) \ge 0$.

Then we obtain:

$$\begin{aligned}
0 &\le Cost_{\text{av}}^*(\boldsymbol{\pi}(t)) \\
&\overset{(a)}{=} q(\boldsymbol{\alpha}(t), t) \\
&\overset{(b)}{\le} \sum_{s_i} \pi_{s_i}(t) \left[VCost_{\max} - \eta_0 \sum_j \alpha_i(t) \right].
\end{aligned} \tag{F.1}$$

Here (a) follows from Theorem 1 in [40] and (b) follows from the definition of $q(\boldsymbol{\alpha}(t), t)$, Assumption 1 and $\sum_{s_i} \pi_{s_i}(t)[\epsilon_i - x_i(s_i, x_k^{(s_i)})] \le 0$. This shows that $w.p.1$

$$\sum_i \alpha_i(t) \le \xi \triangleq \frac{VCost_{\max}}{\eta_0}, \forall t \ge T_{\epsilon_s}. \tag{F.2}$$

Appendix G. Proof of Corollary 1

From Lemma 2 we know that $w.p.1$ after T_{ϵ_s} time, $\|\boldsymbol{\alpha}(t)\| \le \xi$ which implies that for all $t \ge T_{\epsilon_s}$ and all s_i, we have $w.p.1$ that where $\|d_i(t) - x_i(t)\| \le B$:

$$q_{s_i}(\boldsymbol{\alpha}(t)) \le VCost_{\max} + N\xi B, \quad \forall s_i. \tag{G.1}$$

Now note that $q(\boldsymbol{\alpha}(t), t)$ can be expressed as:

$$q(\boldsymbol{\alpha}(t), t) = \sum_{s_i} \left[\pi_{s_i} + \delta_{s_i}(t)\right] q_{s_i}(\boldsymbol{\alpha}(t)), \tag{G.2}$$

where $\delta_{s_i}(t) = \pi_{s_i}(t) - \pi_{s_i}$ denotes the error of the empirical distribution. Therefore,

$$\begin{aligned}
|q(\boldsymbol{\alpha}(t), t) - q(\boldsymbol{\alpha}(t))| &\le \max_{s_i} \left|\delta_{s_i}(t)\right| \sum_{s_i} \left|q_{s_i}(\boldsymbol{\alpha}(t))\right| \\
&\le \max_{s_i} \left|\delta_{s_i}(t)\right| M \left(VCost_{\max} + N\xi B\right).
\end{aligned} \tag{G.3}$$

Appendix H. Proof of Lemma 3

For easy understanding and not to confuse, we use $\boldsymbol{\gamma}(t)$ to represent the $\boldsymbol{Q}(t)$ as the Lagrange dual multiplier of the problem (8) and its optimal value is $\boldsymbol{\gamma}^*$. Note that for all t, the empirical dual function $q(\boldsymbol{\alpha}, t)$ is always concave [41] and hence continuous. Suppose $\boldsymbol{\alpha}(t)$ does not converge to γ^*. Then, there exists a $\delta > 0$ such that we can find an infinite sequence $\{\boldsymbol{\alpha}(t_k)\}_{k=1}^{\infty}$ with $t_k \to \infty$ such that $\left\|\boldsymbol{\alpha}(t_k) - \gamma^*\right\| \ge \delta$ for all k. Since $q(\boldsymbol{\gamma})$ has a unique optimal, this implies that there exists a constant $\epsilon_\delta > 0$ such that for all k,

$$q(\boldsymbol{\gamma}^*) - q(\boldsymbol{\alpha}(t_k)) \ge \epsilon_\delta. \tag{H.1}$$

To see why this holds, suppose this is not the case. Then, for any small $\epsilon > 0$, we can find some $\boldsymbol{\alpha}(t_k)$ such that (H.1) is violated. This implies that there is a sequence of $\{(\boldsymbol{\alpha}(t_m))\}_{m=1}^{\infty}$ which converges to $q(\boldsymbol{\gamma}^*)$. Denote $\mathcal{M} = \{\boldsymbol{\alpha} : \|\boldsymbol{\alpha}\| \le \xi\}$. We see that $\mathcal{M}$ is compact. Also, by Lemma 1, we see that there exists a finite m^*, such that $\{\boldsymbol{\alpha}(t_m)\}_{m=m^*}^{\infty}$ is an infinite sequence in the compact set $\mathcal{M}$ $w.p.1$. Hence, it has a converging subsequence [27]. Denote the limit point of this subsequence by $\boldsymbol{\alpha}'$. The above thus means that $q(\boldsymbol{\alpha}') = q(\boldsymbol{\gamma}^*)$. However, in this case $\|\boldsymbol{\alpha}' - \boldsymbol{\gamma}^*\| \ge \delta$, which contradicts the fact that $\boldsymbol{\gamma}^*$ is the unique optimal of $q(\boldsymbol{\gamma})$. Now let us choose a time T such that $w.p.1$, for all $t \ge T$,

$$|q(\boldsymbol{\alpha}, t) - q(\boldsymbol{\alpha})| \le \epsilon_\delta/3, \tag{H.2}$$

for all $\boldsymbol{\alpha} \in \mathcal{M}$. This is always possible using Corollary 1 and the fact that $\max_{s_i} |\delta_{s_i}(t)| \to 0$ as $t \to \infty$. Then, choose a point $\boldsymbol{\alpha}(t_k)$ from $\{\boldsymbol{\alpha}(t_k)\}_{k=1}^{\infty}$ with $t_k \ge T$. We see that the optimal solution $\boldsymbol{\alpha}(t_k)$ at time t_k satisfies:

$$q(\boldsymbol{\alpha}(t_k), t_k) \ge q(\boldsymbol{\gamma}^*, t_k). \tag{H.3}$$

Using (H.2), (H.3) implies that:

$$q(\boldsymbol{\alpha}(t_k)) + \frac{2}{3}\epsilon_\delta \ge q(\gamma^*), \tag{H.4}$$

which contradicts. This shows that $\boldsymbol{\alpha}(t)$ converge to $\boldsymbol{\gamma}^*$ $w.p.1$.

Appendix I. Proof of Lemma 4

We know that $\tilde{\boldsymbol{\gamma}}^*(t) = \boldsymbol{\gamma}^* - \boldsymbol{\alpha}(t) + \boldsymbol{\theta}$. Next we need to calculate the distance between $\boldsymbol{Q}(t)$ and $\tilde{\boldsymbol{\gamma}}^*(t)$.

$$\begin{aligned}
&\|\boldsymbol{Q}(t+1) - \tilde{\boldsymbol{\gamma}}^*(t)\|^2 \\
&\le \|\boldsymbol{Q}(t) - \boldsymbol{x}(t) + \boldsymbol{d}(t) - \tilde{\boldsymbol{\gamma}}^*(t)\|^2 \\
&\le \|\boldsymbol{Q}(t) - \tilde{\boldsymbol{\gamma}}^*(t)\|^2 + \|\boldsymbol{x}(t) - \boldsymbol{d}(t)\|^2 \\
&\quad - 2(\tilde{\boldsymbol{\gamma}}^*(t) - \boldsymbol{Q}(t))^T (\boldsymbol{d}(t) - \boldsymbol{x}(t)).
\end{aligned} \tag{I.1}$$

Here we notice that the term $(\boldsymbol{d}(t) - \boldsymbol{x}(t))$ is a subgradient of the dual problem $\tilde{q}$, so we can get:

$$\begin{aligned}
&\|\boldsymbol{Q}(t+1) - \tilde{\boldsymbol{\gamma}}^*(t)\|^2 \\
&\le \|\boldsymbol{Q}(t) - \tilde{\boldsymbol{\gamma}}^*(t)\|^2 + B - 2(\tilde{q}(\boldsymbol{\gamma}^*) - \tilde{q}(\boldsymbol{Q}(t))) \\
&\le \|\boldsymbol{Q}(t) - \tilde{\boldsymbol{\gamma}}^*(t)\|^2 + B - 2(q(\boldsymbol{\gamma}^*) - q(\boldsymbol{Q}(t) + \boldsymbol{\alpha}(t) - \boldsymbol{\theta})).
\end{aligned} \tag{I.2}$$

Now for a given $\eta > 0$, if :

$$\begin{aligned}
&\eta^2 - 2\eta \|\boldsymbol{Q}(t) - \tilde{\boldsymbol{\gamma}}^*(t)\| \\
&\quad \ge B - 2(q(\boldsymbol{\gamma}^*) - q(\boldsymbol{Q}(t) + \boldsymbol{\alpha}(t) - \boldsymbol{\theta})).
\end{aligned} \tag{I.3}$$

Then we can plug this into (I.2) and obtain:

$$\begin{aligned}
&\|\boldsymbol{Q}(t+1) - \tilde{\boldsymbol{\gamma}}^*(t)\|^2 \\
&\le \|\boldsymbol{Q}(t) - \tilde{\boldsymbol{\gamma}}^*(t)\|^2 - 2\eta \|\boldsymbol{Q}(t) - \tilde{\boldsymbol{\gamma}}^*(t)\| + \eta^2 \\
&= (\|\boldsymbol{Q}(t) - \tilde{\boldsymbol{\gamma}}^*(t)\| - \eta)^2.
\end{aligned} \tag{I.4}$$

This implies that:

$$\|\boldsymbol{Q}(t+1) - \tilde{\gamma}^*(t)\| \le \|\boldsymbol{Q}(t) - \tilde{\gamma}^*(t)\| - \eta. \tag{I.5}$$

So we need to prove (I.5) hold. Rearranging the terms in (I.5) it becomes:

$$\begin{aligned}
&2(q(\boldsymbol{\gamma}^*) - q(\boldsymbol{Q}(t) + \boldsymbol{\alpha}(t) - \boldsymbol{\theta})) \\
&\ge 2\eta \|\boldsymbol{Q}(t) + \boldsymbol{\alpha}(t) - \boldsymbol{\theta} - \boldsymbol{\gamma}^*\| + B - \eta^2.
\end{aligned} \tag{I.6}$$

This holds whenever:

$$\begin{aligned}
&\rho\|\boldsymbol{\gamma}^* - \boldsymbol{Q}(t) - \boldsymbol{\alpha}(t) + \boldsymbol{\theta}\| \\
&\quad \ge \eta \|\boldsymbol{Q}(t) + \boldsymbol{\alpha}(t) - \boldsymbol{\theta} - \boldsymbol{\gamma}^*\| + \frac{B - \eta^2}{2}.
\end{aligned} \tag{I.7}$$

By choosing $0 < \eta < \rho$ and using (I.7), we see that (I.5) holds whenever:

$$\|\boldsymbol{Q}(t) - \tilde{\boldsymbol{\gamma}}^*(t)\| = \|\boldsymbol{\gamma}^* - \boldsymbol{Q}(t) - \boldsymbol{\alpha}(t) + \boldsymbol{\theta}\| \ge D_p \triangleq \frac{B - \eta^2}{2(\rho - \eta)}. \tag{I.8}$$

In conclusion, if $\|\boldsymbol{Q}(t) - \tilde{\boldsymbol{\gamma}}^*(t)\| \ge D_p$, then $\|\boldsymbol{Q}(t+1) - \tilde{\boldsymbol{\gamma}}^*(t)\| \le \|\boldsymbol{Q}(t) - \tilde{\boldsymbol{\gamma}}^*(t)\| - \eta$.

Appendix J. Proof of Theorem 5

Using Lemma 4, we know that while $\|\boldsymbol{Q}(t) - \tilde{\boldsymbol{\gamma}}^*(t)\| \ge D_p$, $\boldsymbol{Q}(t)$ will decrease until $\|\boldsymbol{Q}(t) - \tilde{\boldsymbol{\gamma}}^*(t)\| \le D_p$. Obviously that means the value of $\boldsymbol{Q}(t)$ will falls into $[\theta - D_p, \theta + D_p]$ when $t \ge T_{\epsilon_s}$

We see that for any ϵ, with probability 1, there exists a time $T_{\epsilon_s} \le \infty$ such that $\|\boldsymbol{\alpha}(t) - \boldsymbol{\gamma}^*\| \le \epsilon$ for all $t \ge T_{\epsilon_s}$. Using this fact in Lemma 4, we see that $w.p.1$, when $t \ge T_{\epsilon_s}$ and $\|\boldsymbol{Q}(t) - \boldsymbol{\theta}\| \ge \tilde{D}_p = D_p + \epsilon$:

$$\|\boldsymbol{Q}(t+1) - \boldsymbol{\theta}\| \le \|\boldsymbol{Q}(t) - \boldsymbol{\theta}\| - \eta + 2\epsilon. \tag{J.1}$$

We can now use an argument as in [38] and show that there exist constants $c_p = \Theta(1), K_p = \Theta(1)$ such that $w.p.1$,

$$\mathcal{P}_p(\tilde{D}_p, m) \le c_p e^{-K_p m}, \tag{J.2}$$

where $\mathcal{P}_p(\tilde{D}_p, m)$ is defined as:

$$\mathcal{P}_p(\tilde{D}_p, m) \triangleq \limsup_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \Pr\{\|\boldsymbol{Q}(\tau) - \boldsymbol{\theta}\| > \tilde{D}_p + m\}. \tag{J.3}$$

This implies that for any $Q_i(t)$, the probability that $Q_i(t) > \theta_i + (\tilde{D}_p + m)$ and $Q_i(t) < \theta_i - (\tilde{D}_p + m)$ is exponentially decreasing in $K_p m$ with $K_p = \Theta(1)$. Hence, if we define $dis_i(t) = \max[Q_i(t) - \theta_j, 0]$, it can be shown that $\bar{dis}_i(t) = \Theta(1)$. Thus, we have:

$$\bar{Q}_{avg} = \sum_i \bar{Q}_i = \sum_i \theta_i + \Theta(1). \tag{J.4}$$

By choosing $\theta_i = O([log(V)]^2)$, we can get Theorem 5.

Appendix K. Proof of Theorem 6

$$\begin{aligned}
&\Delta_T(\Theta(t)) + V\mathbb{E}\left\{\sum_{\tau=t}^{t+T-1} \text{Cost}^{RRW}(\tau) \mid \Theta(t)\right\} \\
&+ \sum_{i\in M} \mathbb{E}\left\{\sum_{\tau=t}^{t+T-1} (\alpha_i(t) - \theta_i)[d_i(\tau) - x_i(\tau)] \mid \Theta(t)\right\} \\
&\le B_2 T + V\mathbb{E}\left\{\sum_{\tau=t}^{t+T-1} \text{Cost}^{RRW}(\tau) \mid \Theta(t)\right\} \\
&+ \sum_{i\in M} \mathbb{E}\left\{\sum_{\tau=t}^{t+T-1} Q_i'(t)[d_i(\tau) - x_i(\tau)] \mid \Theta(t)\right\} \\
&+ \sum_{i\in M} \mathbb{E}\left\{\sum_{\tau=t}^{t+T-1} H_i(t)[\epsilon_i - x_i(\tau)] \mid \Theta(t)\right\}.
\end{aligned} \tag{K.1}$$

Because $\mathbb{E}\{d_i(\tau) - x_i(\tau)\} \le 0$ and $\mathbb{E}\{\epsilon_i - x_i(\tau)\} \le 0$, then where $Cost^*$ is the optimal value of (8):

$$\begin{aligned}
&\Delta_T(\Theta(t)) + V\mathbb{E}\left\{\sum_{\tau=t}^{t+T-1} \text{Cost}^{RRW}(\tau) \mid \boldsymbol{\Theta}(t)\right\} \\
&+ \sum_{i\in M} \mathbb{E}\left\{\sum_{\tau=t}^{t+T-1} (\alpha_i(t) - \theta_i)[d_i(\tau) - x_i(\tau)] \mid \boldsymbol{\Theta}(t)\right\} \\
&\le B_2 T + V Cost^*.
\end{aligned} \tag{K.2}$$

Taking expectations on both sides of it, taking a telescoping sum over $t = 0, \ldots, T-1$, rearranging the terms, and dividing both sides by TV, we have:

$$\begin{aligned}
Cost_{av}^{RRW} &\triangleq \inf \lim_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}[\text{Cost}^{RRW}(\tau)] \\
&\triangleq \inf \lim_{K\to\infty} \frac{1}{(K-1)T} \sum_{\tau=0}^{(K-1)T-1} \mathbb{E}[\text{Cost}^{RRW}(\tau)] \\
&\le \frac{\Delta_T(\boldsymbol{\Theta}(t))}{VT}
\end{aligned}$$

$$
\begin{aligned}
&+ \inf \lim_{K\to\infty} \frac{1}{(K-1)T} \sum_{\tau=0}^{(K-1)T-1} \mathbb{E}[\mathrm{Cost}^{RRW}(\tau)] \\
&\le \frac{B_2}{V} + \frac{1}{T}\mathbb{E}\left\{ \sum_{\tau=t}^{t+T-1} \mathrm{Cost}^{RRW} \mid \boldsymbol{\Theta}(t) \right\} \\
&- \lim_{t\to\infty} \sum_{i\in M} \frac{1}{T}\mathbb{E}\left\{ \sum_{t=0}^{T-1} \frac{(\alpha_i(t)-\theta_i)}{V} \left[d_i(\tau) - x_i(\tau)\right] \right\} \\
&\le \frac{B_2}{V} + Cost^* \\
&+ \lim_{t\to\infty} \sum_{i\in M} \frac{1}{t}\mathbb{E}\left\{ \sum_{t=0}^{T-1} \frac{(\alpha_i(t)-\theta_i)}{V} \left[x_i(\tau) - d_i(\tau)\right] \right\}.
\end{aligned} \tag{K.3}
$$

We let d_i^{av} and x_i^{av} be the average value and analyze the last term separately:

$$
\begin{aligned}
&\lim_{t\to\infty} \sum_{i\in M} \frac{1}{t}\mathbb{E}\left\{ \sum_{t=0}^{T-1} \frac{(\alpha_i(t)-\theta_i)}{V} \left[x_i(\tau) - d_i(\tau)\right] \right\} \\
&\le \frac{\gamma_i^* - \theta_i}{V}(x_i^{av} - d_i^{av}) + \frac{2\epsilon\delta_{max}}{V}.
\end{aligned} \tag{K.4}
$$

Assume that every queue i can only serve $\delta_{\max}$ data in a timeslot. From (J.2), we know that $w.p.1$, if $\theta_i > \tilde{D}_p + \delta_{\max}$, that means m in (J.2)$= \theta_i - \tilde{D}_p - \delta_{\max}$ while $Q_i(t) < \theta_i - (\tilde{D}_p + m)$, then:

$$
\limsup_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \Pr\left\{ Q_i(\tau) < \delta_{\max} \right\} \le c_p e^{-K_p\left(\theta_i - \tilde{D}_p - \delta_{\max}\right)}. \tag{K.5}
$$

This shows that the fraction of time that $Q_i(t)$ is smaller than $\delta_{\max}$ is at most $c_p e^{-K_p\left(\theta_i - \tilde{D}_p - \delta_{\max}\right)}$. So since d_i^{avg} and x_i^{avg} are the average value of $Q_i(t)$, then:

$$
x_i^{av} - d_i^{av} \le \delta_{\max} c_p e^{-K_p\left(\theta_i - \tilde{D}_p - \delta_{max}\right)}, w.p.1. \tag{K.6}
$$

Using $\theta_i = O([log(V)]^2)$ and a sufficiently large V such that $K_p([log(V)]^2 - \tilde{D}_p - \delta_{max}) \ge 2log(V)$, we have:

$$
x_i^{av} - d_i^{av} \le \frac{\delta_{\max} c_p e^{K_p\left(\tilde{D}_p + \delta\max\right)}}{e^{K_p[log(V)]^2}} = O(\frac{1}{V^2}). \tag{K.7}
$$

While $\gamma_i^* = O(V)$, then:

$$
(\gamma_i - \theta_i)(x_i^{av} - d_i^{av}) = O(\frac{1}{V}). \tag{K.8}
$$

In conclusion, $w.p.1$,

$$
Cost_{av}^{RRW} \le Cost^* + \frac{B_2}{V} + O(\frac{1}{V}) = Cost^* + O(\frac{1}{V}). \tag{K.9}
$$

This completes the proof of Theorem 6.

References

[1] H.B. Mcmahan, E. Moore, D. Ramage, S. Hampson, B.A.y. Arcas, Communication-efficient learning of deep networks from decentralized data, 2016, http://dx.doi.org/10.48550/ARXIV.1602.05629, URL https://arxiv.org/abs/1602.05629.

[2] J. Konečnỳ, H.B. McMahan, D. Ramage, P. Richtárik, Federated optimization: Distributed machine learning for on-device intelligence, 2016, arXiv preprint arXiv:1610.02527.

[3] J. Konečnỳ, H.B. McMahan, F.X. Yu, P. Richtárik, A.T. Suresh, D. Bacon, Federated learning: Strategies for improving communication efficiency, 2016, arXiv preprint arXiv:1610.05492.

[4] H.J. Uhlemann, C. Schock, C. Lehmann, S. Freiberger, R. Steinhilper, The digital twin: Demonstrating the potential of real time data acquisition in production systems, Procedia Manuf. 9 (2017) 113–120.

[5] N. Mounir, Y. Guo, Y. Panchal, I. Mohamed, A. Abou-Sayed, O. Abou-Sayed, Integrating big data: simulation, predictive analytics, real time monitoring, and data warehousing in a single cloud application, in: Offshore Technology Conference, OnePetro, 2018.

[6] M. Bastwadkar, C. McGregor, S. Balaji, A cloud based big data health-analytics-as-a-service framework to support low resource setting neonatal intensive care unit, in: Proceedings of the 4th International Conference on Medical and Health Informatics, 2020, pp. 30–36.

[7] S. Morgan, Claypool, Required Texts: M.J. Neely. Stochastic Network Optimization with Application to Communication and Queueing.

[8] L. Huang, M.J. Neely, Delay reduction via Lagrange multipliers in stochastic network optimization, IEEE Trans. Automat. Control 56 (4) (2011) p.842–857.

[9] Y. Jin, L. Jiao, Z. Qian, S. Zhang, S. Lu, X. Wang, Resource-efficient and convergence-preserving online participant selection in federated learning, in: IEEE International Conference on Distributed Computing Systems, ICDCS, 2020.

[10] Y. Chen, S. He, F. Hou, Z. Shi, J. Chen, An efficient incentive mechanism for device-to-device multicast communication in cellular networks, IEEE Trans. Wireless Commun. 17 (12) (2018) 7922–7935.

[11] T.H.T. Le, N.H. Tran, Y.K. Tun, M.N. Nguyen, S.R. Pandey, Z. Han, C.S. Hong, An incentive mechanism for federated learning in wireless cellular network: An auction approach, 2020, arXiv preprint arXiv:2009.10269.

[12] D. Zhang, Y. Qiao, L. She, R. Shen, J. Ren, Y. Zhang, Two time-scale resource management for green internet of things networks, IEEE Internet Things J. 6 (1) (2018) 545–556.

[13] R. Li, Z. Zhou, X. Chen, Q. Ling, Resource price-aware offloading for edge-cloud collaboration: A two-timescale online control approach, IEEE Trans. Cloud Comput. 10 (1) (2022) 648–661, http://dx.doi.org/10.1109/TCC.2019.2937928.

[14] Y. Yao, L. Huang, A. Sharma, L. Golubchik, M. Neely, Data centers power reduction: A two time scale approach for delay tolerant workloads, in: 2012 Proceedings IEEE INFOCOM, IEEE, 2012, pp. 1431–1439.

[15] L. Feng, Q. Yang, K. Kim, K.S. Kwak, Two-timescale resource allocation for wireless powered D2D communications with self-interested nodes, IEEE Access 7 (2019) 10857–10869.

[16] Y. Hu, C. Qiu, Y. Chen, Lyapunov-optimized two-way relay networks with stochastic energy harvesting, IEEE Trans. Wireless Commun. 17 (9) (2018) 6280–6292.

[17] W. Deng, F. Liu, H. Jin, C. Wu, Smartdpss: Cost-minimizing multi-source power supply for datacenters with arbitrary demand, in: 2013 IEEE 33rd International Conference on Distributed Computing Systems, IEEE, 2013, pp. 420–429.

[18] H. Ma, Z. Zhou, X. Chen, Leveraging the power of prediction: Predictive service placement for latency-sensitive mobile edge computing, IEEE Trans. Wireless Commun. 19 (10) (2020) 6454–6468.

[19] S. Ren, Y. He, F. Xu, Provably-efficient job scheduling for energy and fairness in geographically distributed data centers, in: 2012 IEEE 32nd International Conference on Distributed Computing Systems, IEEE, 2012, pp. 22–31.

[20] G. Zhang, Y. Cao, L. Wang, D. Li, Operation cost minimization for base stations with heterogenous energy supplies and sleep-awake mode: A two-timescale approach, IEEE Trans. Cogn. Commun. Netw. 4 (4) (2018) 908–918.

[21] Y. Yao, L. Huang, A.B. Sharma, L. Golubchik, M.J. Neely, Power cost reduction in distributed data centers: A two-time-scale approach for delay tolerant workloads, IEEE Trans. Parallel Distrib. Syst. 25 (1) (2012) 200–211.

[22] Y. Li, S. Xia, M. Zheng, B. Cao, Q. Liu, Lyapunov optimization-based trade-off policy for mobile cloud offloading in heterogeneous wireless networks, IEEE Trans. Cloud Comput. 10 (1) (2022) 491–505, http://dx.doi.org/10.1109/TCC.2019.2938504.

[23] N. Liu, X. Yu, W. Fan, C. Hu, T. Rui, Q. Chen, J. Zhang, Online energy sharing for nanogrid clusters: A Lyapunov optimization approach, IEEE Trans. Smart Grid 9 (5) (2017) 4624–4636.

[24] A. Asheralieva, D. Niyato, Combining contract theory and Lyapunov optimization for content sharing with edge caching and device-to-device communications, IEEE/ACM Trans. Netw. 28 (3) (2020) 1213–1226, http://dx.doi.org/10.1109/TNET.2020.2978117.

[25] Y. Mao, J. Zhang, S. Song, K.B. Letaief, Power-delay tradeoff in multi-user mobile-edge computing systems, in: 2016 IEEE Global Communications Conference, GLOBECOM, IEEE, 2016, pp. 1–6.

[26] S.K. Joshi, K.S. Manosha, M. Codreanu, M. Latva-aho, Dynamic inter-operator spectrum sharing via Lyapunov optimization, IEEE Trans. Wireless Commun. 16 (10) (2017) 6365–6381.

[27] M. Peng, Y. Yu, H. Xiang, H.V. Poor, Energy-efficient resource allocation optimization for multimedia heterogeneous cloud radio access networks, IEEE Trans. Multimed. 18 (5) (2016) 879–892.

[28] L. Li, H. Zhang, Delay optimization strategy for service cache and task offloading in three-tier architecture mobile edge computing system, IEEE Access 8 (2020) 170211–170224.

[29] Z. Chang, L. Liu, X. Guo, Q. Sheng, Dynamic resource allocation and computation offloading for IoT fog computing system, IEEE Trans. Ind. Inform. 17 (5) (2021) 3348–3357, http://dx.doi.org/10.1109/TII.2020.2978946.

[30] Z. Zhou, S. Yang, L. Pu, S. Yu, CEFL: online admission control, data scheduling, and accuracy tuning for cost-efficient federated learning across edge nodes, IEEE Internet Things J. 7 (10) (2020) 9341–9356.

[31] Z. Yang, M. Chen, W. Saad, C.S. Hong, M. Shikh-Bahaei, Energy efficient federated learning over wireless communication networks, IEEE Trans. Wireless Commun. 20 (3) (2021) 1935–1949, http://dx.doi.org/10.1109/TWC.2020.3037554.

[32] C.T. Dinh, N.H. Tran, M.N.H. Nguyen, C.S. Hong, W. Bao, A.Y. Zomaya, V. Gramoli, Federated learning over wireless networks: Convergence analysis and resource allocation, IEEE/ACM Trans. Netw. 29 (1) (2021) 398–409, http://dx.doi.org/10.1109/TNET.2020.3035770.

[33] L. Pu, X. Chen, R. Yun, X. Yuan, P. Zhou, J. Xu, Cocktail: Cost-efficient and data skew-aware online in-network distributed machine learning for intelligent 5G and beyond, in: arXiv preprint arXiv:2004.00799, 2020.
[34] M.S.H. Abad, E. Ozfatura, D. Gunduz, O. Ercetin, Hierarchical federated learning ACROSS heterogeneous cellular networks, in: ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, 2020.
[35] N.H. Tran, W. Bao, A. Zomaya, M.N.H. Nguyen, C.S. Hong, Federated learning over wireless networks: Optimization model design and analysis, in: IEEE INFOCOM 2019 - IEEE Conference on Computer Communications, 2019.
[36] D. Ye, R. Yu, M. Pan, Z. Han, Federated learning in vehicular edge computing: A selective model aggregation approach, IEEE Access 8 (2020) 23920–23935, http://dx.doi.org/10.1109/ACCESS.2020.2968399.
[37] T. Li, M. Sanjabi, A. Beirami, V. Smith, Fair resource allocation in federated learning, 2019, arXiv preprint arXiv:1905.10497.
[38] L. Huang, M.J. Neely, Delay reduction via Lagrange multipliers in stochastic network optimization, in: 2009 7th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks, IEEE, 2009, pp. 1–10.
[39] Y. Lecun, L. Bottou, Gradient-based learning applied to document recognition, Proc. IEEE 86 (11) (1998) 2278–2324.
[40] Huang, M.J. Neely, Max-weight achieves the exact $[O(1/V), O(V)]$ utility-delay tradeoff under Markov dynamics, 2010, arXiv preprint arXiv:1008.0200.
[41] D. Bertsekas, A. Nedic, Convex Analysis and Optimization (Conservative), Athena Scientific, 2003.

Konglin Zhu received the master's degree in computer Science from the University of California, Los Angeles, CA, USA, and the Ph.D. degree from the University of Göttingen, Germany, in 2009 and 2014, respectively. He is now an Associate Professor with the Beijing University of Posts and Telecommunications, Beijing, China. His research interests include Internet of Vehicles, Edge Computing and Distributed Learning.

Wentao Chen received the BEng degree in Communication Engineering from Beijing University of Posts and Telecommunications, China, in 2019. He is now a master student in School of Artificial Intelligence in Beijing University of Posts and Telecommunications. His research interests are in the areas of Online Optimization and Federated Learning.

Lei Jiao received the Ph.D. degree in computer science from the University of Göttingen, Germany. He is currently an assistant professor at the Department of Computer Science, University of Oregon, USA. Previously he worked as a member of technical staff at Nokia Bell Labs in Dublin, Ireland and as a researcher at IBM Research in Beijing, China. He is interested in the mathematics of optimization, control, learning, and economics, applied to computer and telecommunication systems, networks, and services. He publishes papers in journals such as IEEE/ACM Transactions on Networking, IEEE Transactions on Parallel and Distributed Systems, IEEE Transactions on Mobile Computing, and IEEE Journal on Selected Areas in Communications, and in conferences such as INFOCOM, MOBIHOC, ICNP, and ICDCS. He is a recipient of the NSF CAREER Award and the Best Paper Awards of IEEE LANMAN 2013 and IEEE CNS 2019. He has been on the program committees of conferences including INFOCOM, MOBIHOC, ICDCS, and IWQoS, and was the program chair of multiple workshops with INFOCOM and ICDCS.

Jiaxing Wang received the BEng degree in Electronic Science and Technology from Beijing University of Posts and Telecommunications, China, in 2021. She is now a master student in School of Artificial Intelligence in Beijing University of Posts and Telecommunications. Her research interests are in the areas of network optimization, On-orbit satellite of computing and safety certification.

Yuyang Peng received the M.S. and Ph.D. degrees in electrical and electronic engineering from Chonbuk National University, Jeonju, South Korea, in 2011 and 2014, respectively. He worked as a Postdoctoral Research Fellow with the Korea Advanced Institute of Science and Technology, Daejeon, South Korea, from 2014 to 2018. He is currently an Assistant Professor with the School of Computer Science and Engineering, Macau University of Science and Technology, Macau, China. His research activities lie in the broad area of digital communications, wireless sensor networks, and computing. In particular, his current research interests include cooperative communications, spatial modulation, and energy optimization. Dr. Peng currently serves as an Associate Editor for IEEE Access.

Lin Zhang received the B.S. and the Ph.D. degrees in 1996 and 2001, both from the Beijing University of Posts and Telecommunications, Beijing, China. From 2000 to 2004, he was a Postdoctoral Researcher with Information and Communications University, Daejeon, South Korea, and Nanyang Technological University, Singapore, respectively. He joined Beijing University of Posts and Telecommunications in 2004, where he has been a Professor since 2011. He is also the director of Beijing Big Data Center. His current research interests include mobile cloud computing and Internet of Things.