Research Article

Dynamic Transmission Rate Control for Multi-Interface IoT Devices: A Stochastic Optimization Framework

Yuming Zhang¹, Bohao Feng¹, Aleteng Tian¹, Chengxiao Yu¹, Zhiruo Liu², and Hongke Zhang¹

¹School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing 100044, China

²China Academy of Information and Communication Technology, China

Correspondence should be addressed to Bohao Feng; bhfeng@bjtu.edu.cn

Received 5 March 2021; Accepted 21 May 2021; Published 21 June 2021

Academic Editor: Yuanlong Cao

Copyright © 2021 Yuming Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Recent advances in the Internet of Things (IoT) technologies have enabled ubiquitous smart devices to sense and process various kinds of data. However, these innovations also raise the concern of efficient data transmission. Tackling the above issue is nontrivial since the resource constraints and environmental randomness in IoT require a lightweight transmission scheme while guaranteeing system stability. In this paper, we formulate the transmission scheduling problem of multi-interface IoT devices as a concave optimization, aimed at accommodating the randomness of the IoT environment within the network capacity. By applying the Lyapunov optimization technique, we divide the stochastic problem into a series of low-complex subproblems, which can be individually solved per time slot, and develop a dynamical control algorithm that does not require a priori knowledge such as link states. Theoretical analysis shows that our algorithms nicely bound the average queue length and are asymptotically optimal. Finally, extensive simulation results verify the theoretical conclusions and validate the effectiveness of the proposed algorithm.

1. Introduction

With the ubiquitously deployed smart devices, the Internet of Things (IoT) technology has been facilitating the intelligence of residential daily activities by providing advanced services for transportation, agriculture, industrial manufacturing, etc. [1–3]. However, the continuous growth of IoT applications with the proliferation of various devices has resulted in an unprecedented explosion of network data traffic [4]. According to related report [5], the global IoT cellular traffic is expected to grow to 1.7 exabytes per month by 2022, a two-fold increase over 2020. This huge amount of data traffic poses a critical challenge to the current networks [6, 7], making it impractical to provide transmission guarantees.

Recent innovations of scalable communication systems can empower the IoT devices to connect to heterogeneous networks concurrently, enabling IoT devices to transmit data through different networks in parallel [8]. In particular, an IoT device equipped with multiple interfaces can use multi-

ple channels simultaneously in the physical link layer [9] and deliver packets through Multipath TCP (MPTCP) in the network layer [10]. As a result, the IoT devices with multiple data flows can be treated as a transmission scheduler with a many-to-many traffic pattern. Namely, an IoT device can adopt different kinds of data and category them into different types of flows, depending on priorities or requirements. Then, the transmission scheduler can choose to deliver these data flows through either one link or multiple links. In this regard, the IoT device performs like an input-queued switch for data transmission. Moreover, this design can also bring other advantages for IoT applications, such as performance improvement and scalability support [11].

The novel transmission paradigm introduced by [8] also raises the concerns of transmission scheduling for achieving the desired transmission rate. For example, how many packets of different flows should be transmitted through which link, and how to adjust transmission rates for each flow with system stability guaranteed. These concerns are

further complicated with IoT devices' mobility, as it brings dynamical communication environments and stochastic link conditions. Few efforts like [12] consider the stochastic multipath IoT scenarios, as it is not easy to determine the optimal scheduling schemes. The essential reason lies in two aspects. On the one hand, foresightedly optimize the scheduling problems requires the full knowledge of environmental information. However, because of the randomness of packet arrival rates and link states, it is almost impractical to apply this kind of solution to IoT scenarios. Although predicting the network state may be an alternative approach, it is inefficient in facing emergencies, i.e., traffic bursts, which would result in system instability. On the other hand, most IoT devices are constrained with limited computation and energy resources, incapable of operating complex algorithms. Besides, it is also unnecessary to spare a large number of resources for determining data transmission decisions. Thus, efficient online scheduling solutions with low computation complexity are more suitable for IoT devices.

This paper introduces a novel stochastic optimization framework for multi-interface IoT devices to simultaneously achieve optimal transmission scheduling. We first formulate the transmission scheduling problem as a stochastic optimization with the objective of maximizing the long-term transmission utility within the network capacity. Then, we leverage the Lyapunov optimization technique to develop a low-complexity scheduling algorithm. Our main contributions are summarized as follows.

- (1) We characterize the multi-interface IoT devices with multiple traffics as a multiqueue model and decouple the scheduling process into two subproblems: admission control and output rate control. Then, we formulate the transmission scheduling problem as a stochastic concave optimization, which includes the randomness of network states and system stability constraint
- (2) By leveraging the *drift-plus-penalty* framework, we divide the proposed stochastic problem into three deterministic subproblems, which can be further separated based on their linearly coupling characteristics. Then, we propose a low-complex transmission scheduling algorithm and prove that it can provide an upper bound of the queue length and achieve a $[O(V), O(1/V)]$ trade-off between the queue length and the transmission utility
- (3) Extensive simulation results demonstrate the efficiency of our algorithms, which explicitly outperform the benchmarks in terms of system stability, average delay, and network utility

The rest of this paper is organized as follows. Section 2 reviews the related works. Section 3 describes the considered model and formulates the scheduling problem. In Section 4, we develop the scheduling algorithm and present the theoretical analysis. Experimental results are shown in Section 5, and the conclusion is given at last in Section 6.

2. Related Works

As transmission rate control plays a crucial issue in the research literature of communication, there have been extensive efforts in developing customized schemes for diverse network paradigms in different aspects. For example, to deal with the variation traffic requirements, the authors of [13] consider the situation that the resources can be dynamically shifted between cells and develop a dynamic resource allocation protocol. For device to device communication systems, the authors of [14] try to maximize the weighted sum transmission rate and use a two-step approach to solve the non-convex mixed-integer problem of resource allocation and subchannel assignment. In [15], the authors focus on the information-centric network and present a multipath-aware ICN rate-based congestion control algorithm to calculate per-link rates for the multipath scenario. The authors of [16] address the bandwidth sharing issues in a software-defined network and design a distributed resource allocation algorithm that can provide a trade-off between fairness and cost. For data center networks, the authors of [17] formulate the multiple rate control issue as a convex optimization problem and use their proposed transmission protocol to achieve efficient bandwidth allocation.

In addition, there are also many research works jointly considering the assignment of transmission and other kinds of resources. In [18], the authors study the trade-off between data rate performance and energy consumption in heterogeneous networks and introduce an energy-efficient scheduling scheme to improve system performance. The authors of [19] propose a low-complexity algorithm by difference-of-convex programming to simultaneously optimize service level selection and transmission resource allocation in mobile edge computing systems. By jointly considering task assignment, transmission, and computing resources allocation, the authors of [20] propose a multilayer data flow process system that can provide low latency services for real-time applications. The authors of [21] consider a three-node relay system and provide analytical solutions to the proposed optimization problem for power assignment and relay location. The authors of [22] formulate a delay-sensitive data offloading algorithm to optimize the computing and communication resources to minimize the execution delay and transmission delay concurrently for fog networks.

3. System Model and Problem Formulation

3.1. Multi-Interface System Model. In this paper, we consider the transmission scheduling problem in a typical multi-interface IoT scenario. Each IoT device is equipped with multiple antennas and can deliver data through different links concurrently. Their collected data are categorized into different types, forming transmission flows, respectively. The transmission scheduler operates at the IoT device and makes transmitting decisions according to current network condition. Hence, the multi-interface IoT system can be viewed as a model of a single node with multiple uplink channels. Table 1 summarizes the notations used in this paper.

TABLE 1: Notations used in this paper.

Notation	Description
M	Number of data flows
N	Number of available links
$\alpha^m(t)$	Arrived packet number of flow m at time t
$\mu_n(t)$	The transmission rate of link n at time t
$a_n^m(t)$	Number of packets from data flow m allocated to link n at t
$u_n^m(t)$	The transmission rate of link n assigned to data flow m at t
$Q_n^m(t)$	Queue backlog of data flow m scheduled to link n at t
$\bar{a}_n$	Time-average admitted packets of link n
$\Phi_n(\cdot)$	The utility function of link n
ω_n	Weight factor of link n
$Y_n(t)$	Length of virtual queue for link n
$\gamma_n(t)$	The auxiliary variable of link n at t
V	Nonnegative penalty parameter

To facilitate the analysis of the above model and deal with the time-varying link states, we assume that the system operates in discrete time with unit time slots $t \in \{0, 1, 2, \dots\}$. At every time slot, packets randomly enter the transmission scheduler. We define $\alpha^m(t)$ as the amount of packets of flow m (in the unit of packet number) that arrive at time t . For link n , we use $\mu_n(t)$ to denote its maximum allowable rate, the number of packets that can be transmitted at time t . To make our model practical, we only assume $\alpha^m(t)$ and $\mu_n(t)$ are independent and identically distributed (i.i.d) over time and rate-convergent, which means that equations (1) and (2) hold with probability 1. Note that, in our work, the scheduler does not need to know the average arrival rate $\bar{\alpha}^m$ and transmission rate $\bar{\mu}_n$ previously.

$$\bar{\alpha}^m = \lim_{t \rightarrow \infty} \frac{1}{t} \sum_{\tau=0}^t \alpha^m(\tau) < \infty, \quad (1)$$

$$\bar{\mu}_n = \lim_{t \rightarrow \infty} \frac{1}{t} \sum_{\tau=0}^t \mu_n(\tau) < \infty. \quad (2)$$

In addition, we assume a maximum arrival rate $\alpha^{m,\max}$ and a maximum transmission rate $\mu_{n,\max}$, regardless of the time and the channel state, so that

$$0 \leq \alpha^m(t) \leq \alpha^{m,\max}, 0 \leq \mu_n(t) \leq \mu_{n,\max}. \quad (3)$$

Our goal in this work is to design a dynamic, optimal algorithm for the transmission scheduler to make the following decisions strategically: (1) *scheduling decision*: how many packets of flow m should be transmitted through link n at each time slot? (2) *Rate control*: how does the device allocate the transmission rate while ensuring system stability?

We next propose a multiqueue model to characterize the scheduling problem and then develop an optimization

framework to solve the first problem. After that, we introduce a virtual queue to cope with the second issue.

3.2. Queue Model and Optimization Objective

3.2.1. Queue Model. According to the above system, we assume that the transmission scheduler holds multiple queues for each transmitting link and flow, respectively, as shown in Figure 1. Let $Q_n^m(t)$ represent the queue backlog of flow m scheduled to link n on time t . At each slot t , the scheduler observes the arrived packets $\alpha_m(t)$ and then chooses an admission schedule policy $\mathbf{a}_m(t) = \{a_1^m(t), a_2^m(t), \dots, a_n^m(t)\}$. $a_n^m(t)$ denotes the number of packets of flow m allocated to link n . Besides, it also determines a transmission vector $\mathbf{u}_n(t) = \{u_n^1(t), u_n^2(t), \dots, u_n^m(t)\}$ based on the current link condition, where $u_n^m(t)$ represents the transmission rate of link n assigned to flow m . Hence, the queue length $Q_n^m(t)$ evolves according to the following equation.

$$Q_n^m(t+1) = \max \{Q_n^m(t) - u_n^m(t), 0\} + a_n^m(t). \quad (4)$$

To ensure system stability, we use the definition of queue stability in [23], which is given as follows.

$$\bar{Q} \triangleq \lim_{t \rightarrow \infty} \frac{1}{t} \sum_{\tau=0}^t \sum_n \sum_m Q_n^m(\tau) < \infty. \quad (5)$$

By definition, the queue length can be bounded by a positive constant, implying that the average queuing latency is also constrained.

Additionally, the admission decision $\mathbf{a}_m(t)$ is made subject to the constraint, $\alpha^m(t) = \sum_n a_n^m(t)$, implying that all packets should be admitted. Similarly, $\mathbf{u}_n(t)$ must satisfy $0 \leq \sum_m u_n^m(t) \leq \mu_n(t)$, which means that the delivered packet rate could not exceed the link capacity at any time.

FIGURE 1: Illustration of the queue model at the IoT device.

3.2.2. Optimization Objective. To introduce the optimization objective, we define the time-average admitted packets on link n , $\bar{a}_n$.

$$\bar{a}_n = \lim_{t \rightarrow \infty} \frac{1}{t} \sum_{\tau=0}^t \sum_m a_n^m(\tau). \quad (6)$$

After that, we introduce a continuous, concave, and non-decreasing utility function, $\Phi_n(\bar{a}_n)$, to represent our optimization target. According to [23], this kind of utility function can be used to measure network fairness. Intuitively, this utility can be the profit gained by transmitting packets through link n or the reciprocal of transmission cost of link n . An example of $\Phi_n(\cdot)$ can be given by $\Phi_n(x) = \log(1+x)$, which is also used in our simulation experiments. Hence, to achieve long-time utility maximization, we choose to maximize the weighted sum of all transmission utilities. Then, the formulated scheduling problem can be as follows.

$$\max \sum_n \omega_n \Phi_n(\bar{a}_n), \quad (7a)$$

$$\text{s.t. } \alpha^m(t) = \sum_n a_n^m(t), \quad \forall t, n, m, \quad (7b)$$

$$0 \leq \sum_m u_n^m(t) \leq \mu_n(t), \quad \forall t, n, m, \quad (7c)$$

$$\text{All queues } Q_n^m(t) \text{ are stable}, \quad (7d)$$

where ω_n is the weight factor which can be used to dominate the transmission fairness. Additionally, we assume that $\Phi_n(\cdot)$ satisfies the Lipschitz condition, $|\Phi_n(x) - \Phi_n(y)| \leq D|x - y|$, where D is a constant.

Intuitively, the above optimization problem is an integer programming problem. It is hard to derive the optimal solution directly. Moreover, it requires the full knowledge of state information of all time, which is almost impossible in the realistic environment.

In the next section, we will leverage the Lyapunov optimization technique to decompose the scheduling problem and

provide an online algorithm to approximate the optimal solution with system performance guaranteed.

4. Dynamic Scheduling Algorithm

4.1. Problem Transformation with Virtual Queue. To cope with the aforementioned transmission scheduling problem, we introduce a vector of auxiliary variables, $\gamma(t) = \{\gamma_1(t), \gamma_2(t), \dots, \gamma_n(t)\}$, with constraints $0 \leq \gamma_n(t) \leq \mu_{n,\max}$. Then, the problem ((7a), (7b), (7c), and (7d)) can be transformed as follows.

$$\max \sum_n \omega_n \Phi_n(\bar{\gamma}_n), \quad (8a)$$

$$\text{s.t. } a_n^m(t), u_n^m(t) \in \Omega, \quad \forall t, n, m, \quad (8b)$$

$$\bar{\gamma}_n \leq \bar{a}_n, \quad \forall n, \quad (8c)$$

$$0 \leq \gamma_n(t) \leq \mu_{n,\max}, \quad \forall t, n, \quad (8d)$$

$$\text{All queues } Q_n^m(t) \text{ are stable}, \quad (8e)$$

where Ω denotes the state space given by (7a) and (7b) that arbitrary stationary algorithms can achieve. According to [24], the auxiliary variables, $\gamma(t)$, are introduced to decouple the variables from the optimization objective. Besides, in this paper, the auxiliary variables can also simplify the optimization problem by replacing multiple variables with a single variable.

The explicit explanation of the problem transformation can be as follows. According to the definition of Ω , the constraint (8a), which is equivalent to (7a) and (7b), stands for the feasible region of the problem ((7a), (7b), (7c), and (7d)). Then, if we always choose $\gamma_n(t) = \sum_m a_n^m(t)$, (8b) and (8c) are always satisfied, making these two problems the same. This is because the queue stability requirement constrains that the average arrival rate cannot exceed the maximum transmission rate. Besides, as $\Phi_n(\cdot)$ is nondecreasing, if the optimal solution of the problem ((8a), (8b), (8c), (8d), and (8e)) is obtained when constraint (8c) holds with inequality, it provides a utility that is at least as good as the optimal value of (7a), (7b), (7c), and (7d). Therefore, the scheduling policy determined by (8a), (8b), (8c), (8d), and (8e) also solves (7a), (7b), (7c), and (7d). Readers interested in the proof of the transformation can refer to [24–26].

To satisfy (8c), we introduce a virtual queue $Y_n(t)$ for each link n , with an arrival rate of $\gamma_n(t)$ and a departure rate of $\sum_m a_n^m(t)$, which evolves as

$$Y_n(t+1) = \max \left\{ Y_n(t) - \sum_m a_n^m(t) + \gamma_n(t), 0 \right\}. \quad (9)$$

According to [23], if this virtual queue is stable, the time-average value of $\sum_m a_n^m(t)$ is greater than or equal to the time-average value of $\gamma_n(t)$, which ensures (8c), and so the optimization objective will also be large.

4.2. Problem Decomposition via Drift-plus-Penalty Minimization. Next, we solve the problem ((8a), (8b), (8c), (8d), and (8e)) via the drift-plus-penalty framework, aimed at developing a dynamic scheduling algorithm that can preserve the queue stability and solution optimality.

Let $\Theta(t) \triangleq [\mathbf{Q}(t), \mathbf{Y}(t)]$ be the collective vector of all $Q_n^m(t)$ and $Y_n(t)$ queues. Then, we define the following quadratic Lyapunov function.

$$L(\Theta(t)) = \frac{1}{2} \left[\sum_n \sum_m (Q_n^m(t))^2 + \sum_n (Y_n(t))^2 \right]. \quad (10)$$

Denote $\Delta(\Theta(t)) = L(\Theta(t+1)) - L(\Theta(t))$ as the *one-step conditional Lyapunov drift*. According to [23], we can determine our scheduling policy by minimizing the following *drift-plus-penalty* expression at each time slot.

$$\Delta(\Theta(t)) - V\mathbb{E} \left\{ \sum_n \omega_n \Phi_n(\gamma_n(t)) | \Theta(t) \right\}, \quad (11)$$

where V is a nonnegative penalty parameter that will affect the utility delay trade-off. Instead of minimizing equation (11), our algorithm minimizes its upper bound, which has the following expression.

$$\begin{aligned} \Delta(\Theta(t)) - V\mathbb{E} \left\{ \sum_n \omega_n \Phi_n(\gamma_n(t)) | \Theta(t) \right\} &\leq B \\ &- V\mathbb{E} \left\{ \sum_n \omega_n \Phi_n(\gamma_n(t)) | \Theta(t) \right\} \\ &+ \sum_n \sum_m Q_n^m(t) \mathbb{E} \{ (a_n^m(t) - u_n^m(t)) | \Theta(t) \} \\ &+ \sum_n Y_n(t) \mathbb{E} \left\{ \left(\gamma_n(t) - \sum_m a_n^m(t) \right) | \Theta(t) \right\}, \end{aligned} \quad (12)$$

where B is a positive constant satisfying

$$B \geq \frac{1}{2} \left[\sum_n \sum_m (a_n^m(t)^2 + u_n^m(t)^2) + \sum_n \left(\gamma_n(t)^2 + \left(\sum_m a_n^m(t) \right)^2 \right) \right]. \quad (13)$$

As $a_n^m(t)$, $u_n^m(t)$, and $\gamma_n(t)$ are all bounded, it is easy to prove that such a constant b always exists for all t . For example, an upper bound of B can be defined as $1/2(1 + |M|)(|N| |M| \alpha_{\max}^2 + |N| \mu_{\max}^2)$, where $\alpha_{\max}$ and $\mu_{\max}$ denote the maximum value of the arrival and transmission rate, respectively.

Our dynamic algorithm is given to make scheduling decisions by minimizing the right hand of equation (12) in each time slot t , based on the observed queue states, $Q_n^m(t)$ and $Y_n(t)$. Furthermore, by observing the form of equation (12), the scheduling decisions for $a_n^m(t)$, $u_n^m(t)$, and $\gamma_n(t)$ are linearly coupled. Thus, we can decompose the scheduling problems into the following three subproblems *auxiliary variable selection*, *admission control*, and *transmission rate allocation*.

(1) *Auxiliary Variable Selection.* The auxiliary variables can be chosen by minimizing the following expression:

$$\begin{aligned} \min \quad & \sum_n (Y_n(t) \gamma_n(t) - V \omega_n \Phi(\gamma_n(t))) \\ \text{s.t.} \quad & 0 \leq \gamma_n(t) \leq \mu_{n,\max}. \end{aligned} \quad (14)$$

As $\gamma(t)$ is linearly coupled, $\gamma_n(t)$ can be separable with each other. Besides, given that $\Phi_n(\gamma_n(t))$ is continuous and concave, we can provide a closed-form solution of (14), $[\Phi_n'^{-1}(Y_n(t)/(V\omega_n))]_0^{\mu_{n,\max}}$, where $\Phi_n'^{-1}(\cdot)$ is the inverse of $\Phi_n(\cdot)$'s derivative and $[x]_y^z$ denotes $\min(\max(x, y), z)$.

(2) *Admission Control.* By omitting the terms containing $a_n^m(t)$, the admission control can be determined by solving the following problem:

$$\begin{aligned} \min \quad & \sum_m \left[\sum_n (Q_n^m(t) - Y_n(t) a_n^m(t)) \right] \\ \text{s.t.} \quad & \alpha^m(t) = \sum_n a_n^m(t). \end{aligned} \quad (15)$$

In the case of the separable objective, problem (15) can be divided into m independent subproblems, each for one flow. The optimal solution can be as follows.

$$a_n^m(t) = \begin{cases} \alpha^m(t), & \text{argmin}(Q_n^m(t) - Y_n(t)), \\ 0, & \text{otherwise,} \end{cases} \quad (16)$$

(3) *Transmission Rate Allocation.* Similarly, the transmission rate allocation variable, $u_n^m(t)$, can be obtained by maximizing the following expression:

$$\begin{aligned} \max \quad & \sum_m \sum_n Q_n^m(t) u_n^m(t) \\ \text{s.t.} \quad & 0 \leq \sum_m u_n^m(t) \leq \mu_n(t). \end{aligned} \quad (17)$$

Intuitively, the transmission rate allocation algorithm always trends to serve the longest queue.

The scheduling problem can be described by Algorithm 1. At every time slot t , the scheduler receives the packets, $a_n^m(t)$, and observes the current states of all queues, $Q_n^m(t)$ and $Y_n(t)$. Then, it makes scheduling decisions by solving (14), (15), and (17). After that, it updates the virtual queues $Y_n(t)$ according to equation (9) and the actual queues $Q_n^m(t)$ according to equation (4), with the derived $\gamma_n(t)$, $a_n^m(t)$, and $u_n^m(t)$.

Initialization: Set $t = 0$ and choose a nonnegative penalty parameter V ; set up transmission queues $Q_n^m(0) = 0$ and virtual queues $Y_n(0) = 0$ for all n and m ;

In each time slot t :

- 1: The scheduler receives the arrived packets $a^m(t)$ and observes the current queue states $Q_n^m(t)$ and $Y_n(t)$;
- 2: Solve (14), (15), and (17) to obtain the decision variable $\gamma_n(t)$, $a_n^m(t)$, and $u_n^m(t)$ as follows;
- 3: **for** $n \in N$ **do**
- 4: $\gamma_n(t) \leftarrow [\Phi_n^{I^{-1}}(Y_n(t)/(V\omega_n))]_0^{\mu_{n,\max}}$;
- 5: $u_n^m(t) \leftarrow \mu_n(t)$, if $m = \text{argmax}(Q_n^m(t))$;
- 6: **end for**
- 7: **for** $m \in M$ **do**
- 8: $a_n^m(t) \leftarrow \alpha^m(t)$, if $n = \text{argmin}(Q_n^m(t) - Y_n^m(t))$
- 9: **end for**
- 10: Assign packets to queue $Q_n^m(t)$ according to $a_n^m(t)$, and deliver packets through link n based on $u_n^m(t)$;
- 11: Update all queues according to (4) and (9).

ALGORITHM 1: Pseudocode of the proposed algorithm.

FIGURE 2: Queue length of the weighted round robin, weighted random, and myopic algorithm.

4.3. Performance Analysis. Here, we provide the performance of our dynamic algorithm, with respect to the queue length and utility.

Theorem 1. Suppose the problem ((8a), (8b), (8c), (8d), and (8e)) is feasible, all queues are initially set to be zero, and the above algorithm is used in each time slot t with a fixed V ; then:

(a) All queues $Q_n^m(t)$ are stable for all t . Specifically, the following equation holds:

$$\limsup_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^T \sum_n \sum_m Q_n^m(t) \leq \frac{B + V|W|DR}{\epsilon}, \quad (18)$$

where D is the Lipschitz constant, R represents the region of γ , ϵ

is a positive constant, and $|W|$ denotes the 1-norm of the weight factor.

(b) The utility achieved by the proposed algorithm satisfies

$$\liminf_{t \rightarrow \infty} \left(\Phi^{\text{opt}} - \sum_n \omega_n \Phi(\bar{a}_n) \right) \leq \frac{B}{V}, \quad (19)$$

where Φ^{opt} is the maximum utility of the problem ((7a), (7b), (7c), and (7d)).

Proof. According to Theorem 4.5 in [23], the feasibility of the problem ((7a), (7b), (7c), and (7d)) implies that for any δ , $\delta > 0$, there exists a scheduling policy $\widehat{\gamma}_n(t)$, $\widehat{a}_n^m(t)$, and $\widehat{u}_n^m(t)$, which yields

$$\begin{aligned} \mathbb{E} \left\{ \widehat{a}_n^m(t) - \widehat{u}_n^m(t) \right\} &\leq \delta, \\ \mathbb{E} \left\{ \widehat{\gamma}_n(t) - \sum_m \widehat{a}_n^m(t) \right\} &\leq \delta, \\ \Phi^{\text{opt}} - \Phi_n(\widehat{\gamma}_n(t)) &\leq \delta. \end{aligned} \quad (20)$$

Using $\delta = 0$ and plugging the above into equation (12) yield

$$\Delta(\Theta(t)) - V \mathbb{E} \left\{ \sum_n \omega_n \Phi_n(\gamma_n(t)) \right\} \leq B - V \Phi^{\text{opt}}. \quad (21)$$

Summing up the above equation of all t , $t \leq T$, and dividing both side by T , we can have

$$\frac{1}{T} \sum_{t=0}^T \left(\Phi^{\text{opt}} - \mathbb{E} \left\{ \sum_n \omega_n \Phi_n(\gamma_n(t)) \right\} \right) \leq \frac{B}{V} + \frac{\mathbb{E}\{L(\Theta(0))\}}{VT}. \quad (22)$$

FIGURE 3: Average delay of the weighted round robin, weighted random, and myopic algorithm.

FIGURE 4: Average utility of the weighted round robin, weighted random, and myopic algorithm.

As $\Phi_n(\cdot)$ is concave and nondecreasing, based on Jensen's inequality, the following equation can be derived.

$$\frac{1}{T} \sum_{t=0}^T \sum_n \omega_n \Phi_n(\gamma_n(t)) \leq \sum_n \omega_n \Phi_n(\bar{\gamma}_n) \leq \sum_n \omega_n \Phi_n(\bar{a}_n). \quad (23)$$

Plugging the above equation into equation (22) and taking $\liminf_{T \rightarrow \infty}$ of both sides prove part (b).

To prove part (a), we can use $\widehat{\gamma}_n(t) = \mathbb{E}\{\sum_m \widehat{a}_n^m(t)\}$. Then, according to the queue stability constraint, we have the following equations.

$$\begin{aligned} \mathbb{E}\{\widehat{a}_n^m(t) - \widehat{u}_n^m(t)\} &\leq -\varepsilon, \\ \mathbb{E}\{\Phi_n(\widehat{\gamma}_n(t))\} &= \Phi_n(\varepsilon). \end{aligned} \quad (24)$$

Plugging these two equations in equation (12), we can have

$$\Delta(\Theta(t)) - V \mathbb{E}\left\{\sum_n \omega_n \Phi_n(\gamma_n(t))\right\} \leq B - V \Phi_n(\varepsilon) - \varepsilon \sum_n \sum_m Q_n^m(t). \quad (25)$$

Rearranging the terms of the above equation and using telescoping yield

$$\varepsilon \sum_{t=0}^T \sum_n \sum_m Q_n^m(t) \leq BT - V \sum_{t=0}^T \sum_n \omega_n \Phi_n(\varepsilon) + V \sum_{t=0}^T \mathbb{E}\left\{\sum_n \omega_n \Phi_n(\gamma_n(t)) | \Theta(t)\right\}. \quad (26)$$

Consider that $\Phi_n(\cdot)$ satisfies the Lipschitz condition. Then, the following equation can be derived.

$$\begin{aligned} \sum_n \omega_n (\Phi_n(\gamma_n(t)) - \Phi_n(\varepsilon)) &\leq D \sum_n \omega_n |\gamma_n(t) - \varepsilon| \\ &\leq |W|DR, \end{aligned} \quad (27)$$

Using this in equation (26) and dividing both sides by εT , we can derive for all $t > 0$

$$\frac{1}{T} \sum_{t=0}^T \sum_n \sum_m Q_n^m(t) \leq \frac{B + V|W|DR}{\varepsilon} + \frac{L(\Theta(0))}{\varepsilon T}. \quad (28)$$

Finally, taking $\limsup_{T \rightarrow \infty}$ of both sides proves part (a).

Theorem 1 demonstrates that the control parameter V dominates the performance of our algorithm. In particular, the upper bound of the queue length grows linearly with V . This situation implies we can choose a small V to achieve a short queue length, which corresponds to a short queuing delay according to *Little's theory*. Besides, it also confirms that the gap between the achieved utility and the optimal one is bounded by $O(1/V)$. The gap can be made arbitrarily small by increasing V , which declares that increasing V can approximate the optimal solution of (8a), (8b), (8c), (8d), and (8e). Overall, we can conclude that our algorithm achieves the trade-off between the queuing delay and the transmission utility with $[O(V), O(1/V)]$. This feature shows that improving the transmission utility is at the expense of increasing the queuing delay.

Additionally, consider that (14), (15), and (17) are independent. All these subproblems can be solved in parallel. This situation indicates that the overall complexity of our algorithm depends on that of each subproblem. Furthermore, by observing their forms, the complexity of solving (14),

FIGURE 5: Average queue length, average delay, and average utility with different average arrival rates (varying from 60 to 240).

(15), and (17) is related to the number of data flows and links, M and N . In particular, the computation complexity is bounded by $O(MN)$. Normally, N is constant in practical scenarios, and M varies with the number of data types, which is relatively small. Thus, the complexity of our algorithm is acceptable.

5. Simulation Results

This section presents the simulation results of the proposed algorithm, named myopic algorithm. For comparison, we also simulate two other heuristic algorithms as benchmarks:

(1) *weighted round robin*: the scheduler admits and sends packets of different flows in turn and the time slice of each flow is determined based on its average arrival and transmission rates, and (2) *weighted random*: the scheduler admits and transmits packets of different flows according to a given probability distribution, proportional to the average arrival and transmission rates.

5.1. Experimental Setting. In this work, we consider a 5×3 wireless transmission scheduler with five input ports, each admitting one certain flow, and three output ports, each delivering packets through one wireless link [8]. Hence, there

FIGURE 6: Queue length with different V (varying from 50 to 300). FIGURE 7: Average utility with different V (varying from 50 to 300).

are fifteen queues $Q_n^m(t)$, representing that packets arrived from input port m must be delivered to output port n , for $m \in \{1, 2, 3, 4, 5\}$ and $n \in \{1, 2, 3\}$. The arrival processes follow uniform distributions, i.i.d. over time slots, varying from 60 to 240. Considering that link states are time-varying, transmission rates of all links are arbitrarily distributed, i.i.d. over time slots with average rates $\{30, 90, 150\}$. We use $\log(1+x)$ as the utility function for all links and, respectively, set the maximum arrival rate and transmission rate to twice the average rates of the two. We implement a software scheduler with Python and deploy all algorithms in it. To better evaluate the proposed algorithms, we run these three algorithms simultaneously with the same instantaneous arrival and transmission rates. Note that all simulations are

over 5000 time slots, and each data point in the figures is averaged based on 10 times independent runs.

5.2. System Performance. In this section, we verify the system performance with respect to queue length, average delay, and achieved utility, with $V = 100$.

Figure 2 plots the queue length of the three algorithms over time slots. The results confirm that our algorithm outperform the other two algorithms. In particular, the weighted random algorithm varies dramatically over time, while the other algorithms are relatively stable. This is because only the weighted random algorithm schedules the packet randomly, without considering the environment state. Definitely, it can only guarantee that the queue length cannot grow indefinitely.

Figure 3 compares the average delay between the three algorithms. We can see that the myopic algorithm achieves the lowest average delay, followed by myopic weighted round robin, and the weighted random algorithm shows the worst performance. This situation is consistent with the queue length. *Little's theory*, $N = \lambda T$, can provide a reasonable explanation that the average delay grows proportionally with the arrival rate.

In our simulation, we also measure the achieved utility, shown in Figure 4, calculated by the number of packets. Obviously, all these three algorithms converge to a stable value, and our proposed algorithms show a better performance than the benchmarks. The former situation demonstrates the system's stability. The latter is because our proposed algorithms leverage the utility function as targets while the benchmarks can only be designed to ensure system stability. Moreover, the weighted random algorithm converges the slowest. Similar to the queue length, this phenomenon is also induced by the weighted random algorithm's randomness.

5.3. Impact of Algorithm Parameters. In this section, we investigate the impact of the arrival rate and V on system performance.

Figure 5 reveals how the queue length, average delay, and achieved utility vary with the arrival rate. Figure 5(a) shows that the queue length increases with the increase in λ . In particular, the weighted random algorithm grows the fastest, followed by the weighted round robin algorithm, and our proposed algorithm has the shortest queue lengths. Figure 5(b) shows the same trend as Figure 5(a). These two subfigures demonstrate that our proposed algorithm can always provide queuing delay guarantees. As for Figure 5(c), it reveals that the achieved utility also grows as λ rises. However, the increasing rates of the proposed algorithm decrease with λ . This trend can be explained as follows. When λ is small, the primary objective is to maximize the utility. However, when λ is large, system stability becomes essential. Otherwise, the queue length will grow infinitely. As a comparison, the weighted round robin and weighted random algorithms show an increasing trend almost linearly.

Figure 6 shows the average queue length under different values of V . Obviously, for all cases, the queue length converges to be stable. In particular, we can see that, when V

grows, the stable queue length also increases. Besides, the increasing rates of the stable queue length and V have a linear approximation relationship. Figure 7 depicts the average utility under different values of V . Similar to the queue length, all average utilities finally remain stable and increase with the growth of V . However, the growth rate of the average utility is inversely proportional to that of V . Hence, these two figures together confirm the correctness of the aforementioned theorems.

Another observation from these two figures is that a larger V results in a higher convergence rate. As shown in Figure 6, the convergence time of $V = 300$ is more than 1500 time slots, which is almost three times that of $V = 50$. Moreover, in Figure 7, when $V = 300$, the average utility remains stable after the 3000th time slots, and when $V = 50$, the algorithm reaches its stable value around the 1000th time slots. Therefore, it is necessary to choose V carefully when deploying our algorithms in practical systems.

6. Conclusion

In this paper, we investigate the transmission scheduling in IoT. We propose a generic optimization problem and solve it via the Lyapunov optimization technique. Both theoretical proofs and simulation results confirm that our approach can achieve the optimal utility while guaranteeing system stability and constraining average delay. In the future, we will continue our work in two aspects. (1) Extend our algorithms into more scenarios, i.e., delay-sensitive scenarios, where the delay requirement of a specific flow must be satisfied for all times. (2) Improve the performance of our proposed algorithm, i.e., investigating how to choose parameters dynamically to provide real-time transmission resource allocation.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is supported by the National Key Research and Development Program of China under Grant No. 2018YFE0206800, by the Natural Science Foundation of Beijing, China, under Grant No. 4212010, and by the National Natural Science Foundation of China under Grant No. 61802014.

References

- [1] S. Chen, H. Xu, D. Liu, B. Hu, and H. Wang, "A vision of IoT: applications, challenges, and opportunities with China perspective," *IEEE Internet of Things Journal*, vol. 1, no. 4, pp. 349–359, 2014.
- [2] A. H. H. Ngu, M. Gutierrez, V. Metsis, S. Nepal, and M. Z. Sheng, "IoT middleware: a survey on issues and enabling technologies," *IEEE Internet of Things Journal*, vol. 4, no. 1, pp. 1–20, 2017.
- [3] Y. Zhang, C. Xu, H. Li, K. Yang, N. Cheng, and X. Shen, "PROTECT: efficient password-based threshold single-sign-on authentication for mobile users against perpetual leakage," *IEEE Transactions on Mobile Computing*, vol. 20, no. 6, pp. 2297–2312, 2021.
- [4] K. Peng, H. Huang, S. Wan, and V. C. M. Leung, "End-edge-cloud collaborative computation offloading for multiple mobile users in heterogeneous edge-server environment," *Wireless Networks*, vol. 7, no. 4, pp. 2622–2629, 2020.
- [5] G. Forecast, "Cisco visual networking index: global mobile data traffic forecast update, 2017–2022," *Update*, vol. 2017, 2019.
- [6] H. Zhang, W. Quan, H. Chao, and C. Qiao, "Smart identifier network: a collaborative architecture for the future internet," *IEEE Network*, vol. 30, no. 3, pp. 46–51, 2016.
- [7] Y. Zhang, C. Xu, N. Cheng, H. Li, H. Yang, and X. Shen, "Chronos+: an accurate blockchain-based time-stamping scheme for cloud storage," *IEEE Transactions on Services Computing*, vol. 13, no. 2, pp. 216–229, 2019.
- [8] H. Zhang, P. Dong, S. Yu, and J. Song, "A scalable and smart hierarchical wireless communication architecture based on network/user separation," *IEEE Wireless Communications*, vol. 24, no. 1, pp. 18–24, 2017.
- [9] X. Deng, J. Luo, L. He, Q. Liu, X. Li, and L. Cai, "Cooperative channel allocation and scheduling in multi-interface wireless mesh networks," *Peer-to-peer Networking and Applications*, vol. 12, no. 1, pp. 1–12, 2019.
- [10] C. Yu, W. Quan, N. Cheng, S. Chen, and H. Zhang, "Coupled or uncoupled? multi-path TCP congestion control for high-speed railway networks," in *2019 IEEE/CIC International Conference on Communications in China (ICCC)*, pp. 612–617, Changchun, China, 2019.
- [11] P. Dong, T. Zheng, S. Yu, H. Zhang, and X. Yan, "Enhancing vehicular communication using 5G-enabled smart collaborative networking," *IEEE Wireless Communications*, vol. 24, no. 6, pp. 72–79, 2017.
- [12] M. Wang, C. Xu, X. Chen, H. Hao, L. Zhong, and D. O. Wu, "Design of multipath transmission control for information-centric Internet of things: a distributed stochastic optimization framework," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 9475–9488, 2019.
- [13] L. P. Do and R. Lehnert, "Dynamic resource allocation protocol for large PLC networks," in *2012 IEEE International Symposium on Power Line Communications and Its Applications*, pp. 41–46, Beijing, China, 2012.
- [14] H. Tang and Z. Ding, "Mixed mode transmission and resource allocation for D2D communication," *IEEE Transactions on Wireless Communications*, vol. 15, no. 1, pp. 162–175, 2016.
- [15] M. Mahdian, S. Arianfar, J. Gibson, and D. Oran, "MIRCC: multipath-aware ICN rate-based congestion control," in *Proceedings of the 3rd ACM Conference on Information-Centric Networking (ACM-ICN '16)*, pp. 1–10, New York, USA, 2016.
- [16] Z. Allybokus, K. Avrachenkov, J. Leguay, and L. Maggi, "Multi-path alpha-fair resource allocation at scale in distributed software-defined networks," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 12, pp. 2655–2666, 2018.
- [17] L. Chen, Y. Feng, B. Li, and B. Li, "Promenade: proportionally fair multipath rate control in datacenter networks with

- random network coding,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 30, no. 11, pp. 2536–2546, 2019.
- [18] H. Zhang, F. Fang, J. Cheng, K. Long, W. Wang, and V. C. M. Leung, “Energy-efficient resource allocation in NOMA heterogeneous networks,” *IEEE Wireless Communications*, vol. 25, no. 2, pp. 48–53, 2018.
- [19] Y. Gao, Y. Cui, X. Wang, and Z. Liu, “Optimal resource allocation for scalable mobile edge computing,” *IEEE Communications Letters*, vol. 23, no. 7, pp. 1211–1214, 2019.
- [20] P. Wang, C. Yao, Z. Zheng, G. Sun, and L. Song, “Joint task assignment, transmission, and computing resource allocation in multilayer mobile edge computing systems,” *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2872–2884, 2019.
- [21] Y. Jia and A. Vosoughi, “Transmission resource allocation for training based amplify-and-forward relay systems,” *IEEE Transactions on Wireless Communications*, vol. 10, no. 2, pp. 450–455, 2011.
- [22] M. Mukherjee, S. Kumar, M. Shojafar, Q. Zhang, and C. X. Mavromoustakis, “Joint task offloading and resource allocation for delay-sensitive fog networks,” in *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, pp. 1–7, Shanghai, China, 2019.
- [23] M. Neely, “Stochastic network optimization with application to communication and queueing systems,” *Synthesis Lectures on Communication Networks*, vol. 3, no. 1, pp. 1–211, 2010.
- [24] L. Georgiadis, M. Neely, and L. Tassiulas, *Resource Allocation and Cross-Layer Control in Wireless Networks*, Foundations and Trends in Networking, Hanover, MA, 2006.
- [25] M. J. Neely, “Super-fast delay tradeoffs for utility optimal fair scheduling in wireless networks,” *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 8, pp. 1489–1501, 2006.
- [26] M. Neely, “Delay-based network utility maximization,” *IEEE/ACM Transactions on Networking*, vol. 21, no. 1, pp. 41–54, 2013.