Research Article

Handling Interservice Time Constraints in Wireless Networks

Xu Zheng, Jianzhong Li, and Hong Gao

School of Computer Science and Technology, Harbin Institute of Technology, Heilongjiang 150001, China

Correspondence should be addressed to Xu Zheng; zhengxuhit@gmail.com

Received 30 August 2014; Accepted 28 October 2014

Academic Editor: Xiuzhen Cheng

Copyright © 2015 Xu Zheng 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.

Cyberphysical systems collect plenty of information from physical world. This information must be transmitted to the base station immediately to support the making of controlling orders since the physical world keeps transforming rapidly. Such a fact needs the support of real-time transmitting in CPS. Real-time traffic often has some extra QoS requirements. For example, regulating the interservice time, which is the time between two consecutive transmissions of a link, is essential for the real-time traffic in wireless networks. A guarantee of the interservice time of a single user is a precondition to support the normal operating of a system. As far as we know, none of previous work can guarantee the performance on the interservice time. Motivated by this, we design a framework for interservice time guaranteed scheduling. We first define a new capacity region of networks with a strict interservice time guarantee. It is an extension of the well-accepted definition on basic capacity region. Then we propose a novel scheduling policy that is both throughput-optimal and interservice time guaranteed. Simulation results show the policy performs well in interservice time and throughput.

1. Introduction

As mobile devices and wireless communication technology keep improving in recent years, there have been more and more complicated wireless networks based applications like cyber physical systems, mobile social networks, and so forth. Real-time traffic occurs frequently in these applications. For example, in a cyber physical system, the sensors collect huge size of information from physical world. And the data is sent to the base station for further decision making [1]. Due to the quick evolving of physical world [2, 3], the CPS must ensure a rapid response to events in the physical world. Then a basic condition of CPS systems is that packets must be delivered to the base or cluster head before their bringing ground truth [4, 5] goes out of date. Such a condition will lead to a great many real-time transmissions in wireless network. Meanwhile, real-time traffic often has some quality-of-service requirements besides the basic need of throughput optimizing and bandwidth saving [6, 7]. These QoS requirements mainly include regularity of the interservice times, packet delay constraints, and packet delivery ratio. As we know, such requirements fail to be captured by those works that only focus on the optimality of throughput performance when scheduling [8] or routing [9, 10]. Some efforts have been made to improve different aspects of QoS. For example, many scheduling policies [11–16] are proposed to handle the transmitting of packets with deadlines. These policies differ in their definitions of delay constraints. And the performance of packet delivery ratio is considered in [17].

Interservice time is the time between two consecutive services for a user, and regularizing the interservice time means a user can get a second chance to transmit in less than a bounded period of time since its last transmission. The regularity of the interservice time is a precondition to support real-time applications like multimedia streaming [18] and also an index to assess the performance of real-time traffic. It is also very important for data collection in CPS systems, since each node must periodically upload its data to the sink, to make sure the sink can capture the event in the corresponding area. However, only very few works study the regularity of the interservice times [18, 19]. Li et al. design a new scheduling policy [18] that keeps the stability of the system under an arriving rate vector while ensuring the regularity of interservice time for each user. The policy
is mainly designed to achieve provable performance on the mean value of interservice time. However, to guarantee the interservice time of a single user is also essential. A user may grow into impatience with large interservice time, besides the risk in the nonsupport of its real-time traffic. Some problems remain open when the system has a strict constraint on the length of interservice time for every user.

(1) What is the set of arriving rate vectors under which the system is stable and each user keeps a guarantee of the interservice time (i.e., the interservice time of any user is less than a threshold)?

(2) Is there a scheduling policy that supports such set of arriving rate vectors while meeting the requirement on service regularity?

In this work, we focus on designing a scheduling policy that keeps the wireless network stable while providing a guarantee of the regularity of interservice time for each user. The contributions of this paper are summarized as follows.

(i) We formulate the problem of handling interservice time constraint in wireless network and define the service-regularity guaranteed capacity region of the system with a given interservice time constraint.

(ii) We propose a novel scheduling policy in which the interservice time of any user is exactly below the required upper bound given by the constraint. We derive the set of arriving rate vectors inside the service-regularity guaranteed capacity region and prove that the proposed scheduling policy is throughput-optimal, which means it can support all the arriving rate vectors in the capacity region.

(iii) Simulation results indicate that our scheduling policy can meet the interservice time constraint and performs well on the stability of networks.

The rest of the paper is organized as follows. In Section 2, we bring a literature of the review on the development of scheduling policies in wireless network. In Section 3, we introduce the network model used by our work. In Section 4, the novel scheduling policy is proposed. In Section 5, we prove some properties of the scheduling policy. In Section 6, we show and analyse the simulation results of our scheduling policy. And Section 7 concludes the paper.

2. Related Work

Job scheduling is extensively studied in the real-time operating system [20]. However, link scheduling in wireless network is different from the job scheduling. For example, the length of standard packet is often fixed in a wireless network, which means a link with few data may waste the bandwidth. In the operation system, a proper quantum length can be selected to minimize the waste. There are also many differences in the model assumptions. Thus the conclusions for the operating systems can not be directly used for wireless networks.

Link scheduling is a fundamental challenge in the design of wireless networks. It is well-known that the max weight scheduling policy is throughput-optimal [8]. However, the problem of finding the disjoint set of links with max weight is NP-hard. Hence, a max weight scheduling policy incurs high overhead. There have been many efforts to design low complexity scheduling policies with throughput performance guarantees. Some policies are based on the picking-and-comparing method [21] or carrier sensing multiple access (CSMA) [22]. Those policies gradually achieve the optimal throughput after many iterations but are insensitive to the channel fading [23]. Other policies focus on finding a max-imal weight independent set (MWIS) of links. Sakai et al. [24] and Wan et al. [25] designed centralized policies that could find a MWIS with provable ratio bounds. Basagni [26], Hoepman [27], and Joo et al. [28] separately propose distributed scheduling policies for different network models. However, none of these works deal with the QoS requirements of users.

Some scheduling policies are designed to meet the packet delay constraint for each user [11,12,14–16]. They all partition the time into frames, and packets will run out of time before the end of their arriving frames. They are different in the assumption of the traffic. Some of them assume the traffic arrives at the beginning of the frame [11,12], while others assume the arriving happens any time in the frame [14–16]. They also differ in the introducing of channel fading [13], nonidentical delays [16], and so forth. Kang et al. [17] propose a new policy that accommodates packet delivery ratio.

For all we know, very few works [18,19] deal with the performance of interservice time. Li et al. [18] use a new type of weight to capture the interservice time performance. The weight is a combination of queue length and time-since-last-service. Then under the max weight scheduling policy, a user will get a second service with a growing probability as the time-since-last-service increases. And the interservice time can be regulated. Li et al. [19] extend his work to consider the interservice time performance of rate vectors close to the edge of capacity region. Li et al.'s works fail in the situation when the interservice time of each user is strictly bounded and one hopes to know the exact set of arriving rate vectors that can meet the boundary.

3. Network Model

In our network, $N$ users hope to transmit to a common destination via the same channel. Time is partitioned into slot, and in each slot only one user can communicate with the destination because of the destructive interference in wireless communication. In each slot, $A_i[t]$ packets arrive in user $i$. $A_i[t]$ is randomly and identically distributed over time with the expectation $E(A_i[t]) = \lambda_i$. These packets will join the delivering queue and be transmitted to the destination. We assume that the wireless channel is always in good condition, and user $i$ can send $r_i$ packets in a slot. In our model, both $\lambda_i$s and $r_i$s are heterogeneous, which indicates the fact that the size of traffic and communication capability of different mobile device are often various.
We say that the network is stable under an arriving rate vector if the underlying Markov chain is positive recurrent for each user, and
\[
\lim_{t \to \infty} \sum_{i=1}^{N} Q_i[t] \leq \infty.
\]

An arriving rate vector \( \lambda = \{ \lambda_i \}_{i=1}^{N} \) is supported by a network if the network is stable under \( \lambda \) by utilizing some scheduling policy. All supportable rate vectors compose the capacity region of the network, denoted by \( \Lambda \).

Definition 2. A scheduling policy is throughput-optimal for a network if \( \Lambda' = \Lambda \), where \( \Lambda' \) is the set of rate vectors supported by the network when using the scheduling policy.

Interservice time is the number of slots between two consecutive transmissions of a user. We use \( T_i[m] \) to denote the number of time slots between the \( m-1 \)th and \( m \)th transmissions of user \( i \). Accordingly, \( T_i = \lim_{m \to \infty} (\sum_{j=1}^{m} T_i[j])/m \) is the long-term average interservice time of user \( i \).

In this paper, we focus on the network model with an extra constraint that \( T_i \leq N + \epsilon \), for all \( i \), where \( \epsilon \) is a nonnegative real. Then the network not only is required to be stable, but also guarantees a bounded interservice time for all users. \( \epsilon \) is always nonnegative since \( N \) is the lower bound for the constraint, and the proof is trivial. Then we can extend the definitions of stability, capacity region, and throughput-optimal scheduling policies.

Definition 3. We say that the network is \( \epsilon \)-stable under a rate vector \( \lambda \) if the network is stable and \( T_i \leq N + \epsilon \), for all \( i \).

The definitions of \( \epsilon \)-supported and \( \epsilon \)-capacity region \( \Lambda_\epsilon \) are similar.

Definition 4. A scheduling policy is \( \epsilon \)-throughput-optimal for a network if \( \Lambda'_\epsilon = \Lambda_\epsilon \), where \( \Lambda'_\epsilon \) is the set of rate vectors \( \epsilon \)-supported by the network when using the scheduling policy.

Our objective is to design a scheduling policy that is \( \epsilon \)-throughput-optimal for our network model.

### 4. Service-Guaranteed Maximum Weight Scheduling Policy

In this section, we propose our scheduling policy (see Algorithm 1). The novel scheduling policy is a maximum weight type policy, while guaranteeing an upper bound on the average interservice time of all users. We call this policy service-guaranteed maximum weight scheduling (SGMW for short).

SGMW partitions time into equal sized intervals. Each interval contains \( N + \epsilon \) slots, where \( N + \epsilon \) is the upper bound of the average interservice time.

In each interval, the scheduling policy is composed of two components: loose phase and tight phase. The policy first starts in loose phase at the beginning of each interval. In loose phase, all users compete for the channel, and our policy is the same as maximum weight scheduling. SGMW goes into tight phase when the number of time slots left in an interval is the same as the number of users that have not been scheduled in this interval. In tight phase, the policy repeatedly chooses the user whose weight is the largest among the set of users never scheduled in current interval. Obviously, the tight phase lasts until the end of the interval. An interval will terminate before \( N + \epsilon \) slots if all the users have been scheduled at least once. Then the scheduling policy turns into the next interval.

The weight of user \( i \) at time slot \( t \) is calculated by the following equation:

\[
\omega_i[t] = \frac{I_i[t]Q_i[t]}{r_i},
\]

where \( Q_i[t] \) is the length of packets at user \( i \) at the beginning of slot \( t \) and \( r_i \) is transmission rate. \( I_i[t] = -1 \) when the policy is in tight phase and the user has been scheduled in current interval, and \( I_i[t] = 1 \) otherwise.

Our scheduling policy is a maximum weight type policy and here is an instance about how it processes.

![Algorithm 1](image-url)
We present the pseudocode of the SGMW as the end of this section. The weight of all users, and the time complexity is $O(N)$. In Section 5.1, we prove that SGMW is $\varepsilon$-throughput-optimal. In Section 5.2, we derive the ratio between $\varepsilon$-capacity region and the original capacity region. The interservice time guarantee of a given rate vector is discussed at the end of this section.

5.1. $\varepsilon$-Throughput-Optimality. We first give the necessary condition for a rate vector to be $\varepsilon$-supportable.

**Lemma 6.** For any arriving rate vector $\lambda = \{\lambda_i\}_i$ that is $\varepsilon$-supportable by a network with some scheduling policies, $\sum_{i=1}^{N} \max((\lambda_i \ast (N + \varepsilon)/r_i) - 1, 0) \leq \varepsilon$.

**Proof.** We prove the lemma by contradiction. Assume $\lambda' = \{\lambda'_i\}_i$ is an $\varepsilon$-supportable arriving rate vector, and $\sum_{i=1}^{N} \max((\lambda'_i \ast (N + \varepsilon)/r_i) - 1, 0) \geq \varepsilon$.

The number of slots in which user $i$ transmits during a total of $T$ slots is no less than $T/(N + \varepsilon)$. $Q'_i[T]$ denotes the number of packets that remain in $Q_i[T]$ when user $i$ is only scheduled once in each interval:

$$E(Q'_i[T]) = \max\left\{\frac{\lambda'_i}{r_i} T - \frac{T}{N + \varepsilon}, 0\right\},$$

$$\sum_{i=1}^{N} E(Q'_i[T]) = \frac{T}{N + \varepsilon} \sum_{i=1}^{N} \max\left\{\frac{\lambda'_i \ast (N + \varepsilon)}{r_i} - 1, 0\right\}.$$ (5)

There are in total $T - N \ast (T/(N + \varepsilon))$ extra slots except the basic scheduling for all users. The expectation of total queue length is estimated as follows:

$$E\left(\sum_{i=1}^{N} Q_i[T]\right)$$

$$= \sum_{i=1}^{N} E(Q_i[T])$$

$$= \sum_{i=1}^{N} E(Q'_i[T]) - \left(T - N \ast \frac{T}{N + \varepsilon}\right) \ast 1$$ (6)

$$= \frac{T}{N + \varepsilon} \sum_{i=1}^{N} \max\left\{\frac{\lambda'_i \ast (N + \varepsilon)}{r_i} - 1, 0\right\} - \varepsilon \ast T$$

$$= \frac{T}{N + \varepsilon} \left(\sum_{i=1}^{N} \max\left\{\frac{\lambda'_i \ast (N + \varepsilon)}{r_i} - 1, 0\right\} - \varepsilon\right).$$

According to the constraint that $\sum_{i=1}^{N} \max((\lambda'_i \ast (N + \varepsilon)/r_i) - 1, 0) \geq \varepsilon$,

$$E\left(\sum_{i=1}^{N} Q_i[T]\right) \rightarrow +\infty \text{ as } T \rightarrow +\infty.$$ (7)

Then the network is unstable under $\lambda'_i$, which contradicts the assumption. So the condition $\sum_{i=1}^{N} \max((\lambda_i \ast (N + \varepsilon)/r_i) - 1, 0) \leq \varepsilon$ holds for all $\varepsilon$-supportable rate vectors.

---

**Example 5.** There are two users in the network, with $A_1[r] = \lambda_1$, $A_2[r] = \lambda_2$, $\lambda_1/r_1 = 1/2$, and $\lambda_2/r_2 = 1/8, \varepsilon = 1$, so the upper bound of average interservice time is 3.

In slot 1, the SGMW is in loose phase. $w_1[1] = 1/2$ and $w_2[1] = 1/8$; then, user 1 is scheduled.

In slot 2, the SGMW is still in loose phase because there are 2 slots left and only user 2 has not been scheduled. $w_1[2] = 1/2, w_2[2] = 1/4$, and user 1 is scheduled.

In slot 3, the SGMW turns to the tight phase. $w_1[3] = -1/2$ since user 1 has been scheduled in current interval. $w_2[3] = 3/8$; user 2 is scheduled.

In slot 4, a new interval begins, and the process repeats itself. Figure 1 shows the evolution of the network in an interval.

The time complexity of the policy is analyzed as follows. In each slot, the policy needs to check the weight of each user and selects the largest one. This requires linear scanning of all users, and the time complexity is $O(N)$. We need to store the weight of all $N$ users; thus, the space complexity is $O(N)$. We present the pseudocode of the SGMW as the end of this section.

5. Properties of SGMW

In Section 5.1, we prove that SGMW is $\varepsilon$-throughput-optimal. We first put forward a necessary condition for a rate vector to be $\varepsilon$-supportable. Then we prove that SGMW can support all rate vectors following the necessary condition. SGMW can also bound the average interservice time for each user; thus, it is $\varepsilon$-throughput-optimal. In Section 5.2, we derive the ratio between $\varepsilon$-capacity region and the original capacity region. The interservice time guarantee of a given rate vector is discussed at the end of this section.

**Figure 1:** An instance of SGMW.
Lemma 7 proves that SGMW can support (may not be $\epsilon$-supportable) all the rate vectors following the condition in Lemma 6.

**Lemma 7.** For any arriving rate vector $\lambda = \{\lambda_i\}$ with the condition that $\sum_{i=1}^{N} \max\{\lambda_i(N+\epsilon)/r_i - 1, 0\} \leq \epsilon$, the network is stable by utilizing the SGMW.

**Proof.** Each interval can be partitioned into two parts: round robin stage and max weight stage. The round robin stage is composed of $N$ slots, one for each user. The max weight stage includes $\epsilon$ slots, and the user with the largest weight in each of these slots is scheduled. Slots of two stages may appear crosswise with each other. A slot belongs to the round robin stage if it is the only slot that the user transmits in current interval. And if a user transmits multiple times, arbitrary one of them belongs to the round robin stage.

Then we are about to prove the system is stable in both of the two stages. For any user $i$,

$$\frac{\lambda_i}{r_i} = \begin{cases} \frac{\lambda_i}{r_i} + 0, & \text{when } \frac{\lambda_i}{r_i} \leq \frac{1}{N + \epsilon}, \\ \frac{1}{N + \epsilon} + \left(\frac{\lambda_i}{r_i} - \frac{1}{N + \epsilon}\right), & \text{when } \frac{\lambda_i}{r_i} > \frac{1}{N + \epsilon}. \end{cases}$$

(8)

The first part of the right side of the equation is supportable in the round robin stage since it is no larger than $1/(N + \epsilon)$.

The second part of the right side is supported by the max weight stage, which can be deduced as follows:

$$\sum_{i=1}^{N} \max\left\{\frac{\lambda_i}{r_i} - \frac{1}{N + \epsilon}, 0\right\}$$

$$= \frac{1}{N + \epsilon} \sum_{i=1}^{N} \max\left\{\frac{\lambda_i(N + \epsilon)}{r_i} - 1, 0\right\}$$

(9)

$$\leq \frac{\epsilon}{N + \epsilon}.$$  

The network is stable in its max weight stage when $t \cdot \sum_{i=1}^{N} \max\{(\lambda_i/r_i) - (1/(N + \epsilon)), 0\} \leq (t/(N + \epsilon)) \cdot \epsilon$, just as what is proved in (9).

Then the whole rate vector is supportable under our interval-based scheduling policy SGMW.

Lemma 8 is to prove that SGMW can also keep a bound on the interservice time for all users.

**Lemma 8.** If an arriving rate vector $\lambda = \{\lambda_i\}$ is supported by SGMW, the average interservice time $T_i$ of any user $i$ is no more than $N + \epsilon$.

**Proof Sketch.** Time slots are partitioned into intervals in SGMW, and length of each interval is less than $N + \epsilon$. Each user can be scheduled at least once in each interval; thus, the number of services for any user is no less than $t/(N + \epsilon)$, and average interservice time $T_i$ of any user $i$ is no more than $t/(t/(N + \epsilon)) = N + \epsilon$.

SGMW is then $\epsilon$-throughput-optimal by combining Lemmas 7 and 8, which is concluded as Theorem 9.

**Theorem 9.** SGMW is $\epsilon$-throughput-optimal scheduling policy in the proposed network model.

5.2. Discussion. The constraint on interservice time reduces the size of capacity region. This is due to the fact that some users with long queue and large arriving rate have to wait for other users to meet the constraint, and length of their queues could possibly tend to infinity. A smaller $\epsilon$ often leads to larger decrease in capacity region since the waiting appears more frequently. We formulate the decreasing in the size of capacity region in Theorem 11. Lemma 10 estimates the size of vector space composed of all vectors $k = (k_1, k_2, \ldots, k_N)$ that follow $k_i \geq 0$, for all $i$, and $\sum_{i=1}^{N} k_i \leq \beta$. The result of Lemma 10 is applied for the proof of Theorem 11.

**Lemma 10.** For any vector $k = (k_1, k_2, \ldots, k_N)$, $\int_{0}^{\beta} 1 d\sigma = \beta^n/n!$ if $k_i \geq 0$, for all $i$, and $\sum_{i=1}^{N} k_i \leq \beta$. $\int_{0}^{\beta} 1 d\sigma$ is a multiple integration on the $N$ dimensions, also defined as the size of vector space composed of all vectors like $k$.

**Proof.** We prove the lemma by induction. Obviously, when $N = 1$, $\int_{0}^{\beta} 1 d\sigma = \beta$ and when $N = 2$, $\int_{0}^{\beta} 1 d\sigma = \beta^2/2$.

Assume $\int_{0}^{\beta} 1 d\sigma = \beta^k/k!$ is true when $N = k$. When $N = k + 1$,

$$\int_{0}^{\beta} 1 d\sigma = \int_{0}^{\beta} \int_{0}^{\beta} 1 d\sigma' dx,$$  

(10)

where $\sigma'$ is a $k$-dimensional space with $\sum_{i=1}^{k} r_i \leq \beta - r_{k+1}$, and

$$\int_{0}^{\beta} \int_{0}^{\beta} 1 d\sigma' dx = \int_{0}^{\beta} (\beta - x)^n/n! dx$$

$$= - \int_{0}^{\beta} (\beta - x)^n/n! d(\beta - x)$$

$$= - \int_{0}^{(\beta - x)^n/n!} d(y)$$

(11)

$$= - \frac{1}{(n + 1)!} (y')^{n+1} \bigg|_{0}^{\beta}$$

$$= \frac{\beta^{n+1}}{(n + 1)!}.$$  

Theorem 11 gives the ratio between $\epsilon$-capacity region and the original one.

**Theorem 11.** The size of $\epsilon$-capacity region $\Lambda_{\epsilon}$ of a network is no more than $(N!(N + \epsilon)^N) \sum_{i=1}^{N} (C_N/i)! \epsilon$ of its original capacity region $\Lambda$. 


\[ \sum_{i=1}^{N} \text{Max} \left\{ k_i - \frac{1}{N+\epsilon}, 0 \right\} \leq \frac{\epsilon}{N+\epsilon}. \] (12)

Assume there are \( t \) users with \( k_i \geq 1/(N+\epsilon) \), \( t = 0, 1, \ldots, N \), and we define \( \sigma_t \) as the size of vector space:

\[ \sigma_t = C_N^t \int_0^{e/(N+\epsilon)} 1 \, d\sigma_t \ast \left( \frac{1}{N+\epsilon} \right)^{N-t}. \] (13)

\( C_N^t \) is the combinatorial number of picking \( t \) out of \( N \). The second part \( \int_0^{e/(N+\epsilon)} 1 \, d\sigma_t \) is the size of space composed by the vector \( \{k_1 - 1/(N+\epsilon), k_2 - 1/(N+\epsilon), \ldots, k_t - 1/(N+\epsilon)\} \), where \( \sum_{i=0}^t (k_i - 1/(N+\epsilon)) \leq e/(N+\epsilon) \) and \( k_i \geq 1/(N+\epsilon) \) for \( i = 1, \ldots, t \). The third part \( (1/(N+\epsilon))^{N-t} \) is the vector space of the users, whose \( k_i \leq 1/(N+\epsilon) \).

According to Lemma 10, \( \int_0^{e/(N+\epsilon)} 1 \, d\sigma_t = e^t/t!(N+\epsilon)^t \). We add up all the \( \sigma_t \) from \( t = 0 \) to \( t = N \):

\[
\sigma = \sum_{t=0}^{N} \sigma_t = \sum_{t=0}^{N} C_N^t \int_0^{e/(N+\epsilon)} 1 \, d\sigma_t \ast \left( \frac{1}{N+\epsilon} \right)^{N-t}
= \sum_{t=0}^{N} C_N^t \frac{e^t}{t!(N+\epsilon)^t} \ast \left( \frac{1}{N+\epsilon} \right)^{N-t}
= \left( \frac{1}{N+\epsilon} \right)^N \sum_{t=0}^{N} C_N^t \frac{e^t}{t!}.
\]

The size of original capacity region is \( \Lambda = 1!/N! \); thus, \( \Lambda_{\epsilon}/\Lambda \leq (N!/(N+\epsilon)^N) \sum_{t=0}^{N} (C_N^t e^t/t!) \).

We find that when \( \epsilon \to \infty \), \( \Lambda_{\epsilon}/\Lambda \leq (N!/(N+\epsilon)^N) \sum_{t=0}^{N} (C_N^t e^t/t!) \to 1 \). The two capacity regions are relatively identical since the loose interservice time constraint can be omitted. When \( \epsilon \to 0 \), \( \Lambda_{\epsilon}/\Lambda \leq (N!/(N+\epsilon)^N) \sum_{t=0}^{N} (C_N^t e^t/t!) = N!/N^N \), the \( \epsilon \)-capacity region of a round robin manner is much smaller than the max weight scheduling policy.

The performance of a system under an arriving rate vector is mainly represented by the total and max queue length of all users. An interservice time constraint can impair the performance due to the following reasons. (1) Some users with few data are scheduled for meeting the constraints, which potentially wastes the bandwidth. (2) A user with high arriving rate and large queue length still has to wait for other users due to the constraint, which potentially aggravates the max queue length. SGMW can achieve the best performance when it is always in the loose phase. Such situations occur when the constraint \( N+\epsilon \) is larger than the max interservice time of users under the max weight scheduling policy. This max interservice time is determined by the variance of the rate vector, which means it can be small when each user has a relatively identical arriving rate. Meanwhile, the max interservice time has nothing to do with the average arriving rate. We will investigate the relation between the variance and max interservice time in future work. On the other hand, there is a lower bound on the constraint \( N+\epsilon \) for any arriving rate vector. The vector is not \( \epsilon \)-supportable when the required constraint is below the bound. According to Lemma 6, we can derive a lower bound \( \epsilon_L \) for an arbitrary arriving rate vector from the inequality \( \sum_{i=1}^{N} \text{Max} \left\{ \lambda_i \ast (N+\epsilon)/r_i \right\} - 1, 0 \leq \epsilon \).

The bound \( \epsilon_L \) is tight since SGMW can support the vector under all \( \epsilon \geq \epsilon_L \).

6. Evaluation

In this section, we provide the simulation results of SGMW. Our results mainly include three parts. In Section 6.1, we check the interservice time performance of the policy. In Section 6.2, we compare the SGMW with baseline max weight scheduling policy. Both the total queue length and the max single queue length are selected as the evaluation criterion. In Section 6.3, we simulate the performance of SGMW under various arriving rate vectors. The vectors differ in the means and variances. The performance is evaluated by the statistics of the average appearance slot of tight phase in each interval. We assume \( r_i = 1 \) for all user \( i \) in this section.

6.1. Interservice Time Guarantee

To illustrate the interservice time performance of SGMW, we assume four groups of flows, each with 4/8/16/32 flows. The arriving rate of user \( i \) is \( (1/(N+\epsilon) + e/(N+\epsilon) \ast 0.5^i \), where \( N \) is the number of flows, \( e \) is the interservice time constraint, and \( i \) stands for the \( i \)th user. Such a rate vector is supportable by the system since its summation is less than 1. Our result considers both the max and average interservice time, as is shown in Figure 2.

It is observed in Figure 2 that the max interservice time never violates the constraint while \( N \) and \( \epsilon \) vary. We also find that the max interservice time when \( N = 64 \) is equivalent to the constraint \( N+\epsilon \), and this indicates the fact that some users with small rate can be scheduled only once in each interval. The average interservice time increases with \( \epsilon \) since the number of users only scheduled in the tight phase keeps increasing under our setting of arriving rates.

In Figure 3, we investigate the performance of SGMW when the variance of arriving rate vector changes. The system is composed of 8 users, with \( \epsilon = 8 \). \( \lambda_i = (1/(N+\epsilon) + e/(N+\epsilon) \ast (1-\delta)^{i-1} \ast \delta \) for each user \( i \). Obviously, the max interservice time in Figure 3 is staying below the constraint of \( N+\epsilon = 16 \) as \( \delta \) increases.

6.2. Throughput Performance

This subsection illustrates the throughput performance of SGMW. We compare it with the max weight scheduling policy. The total queue length and max queue length are two indicators that define the loads of a system and a single user. In Figure 4, we check the performance under different number of users. We assume \( \epsilon = N \), and \( \lambda_i = (1/(N+\epsilon) \ast 0.99^i + e/(N+\epsilon) \ast 0.5^i \), where \( N \) is the number of users.
The total queue length of SGMW grows faster than max weight scheduling policy. This is due to the increasing number of users whose rate is less than $1/(N + \epsilon)$. To schedule such users will cause the waste of bandwidth and a longer tight phase. The increasing in max queue length is also caused by this reason: some users have to keep waiting in the tight phase; then, their queue length grows rapidly. Meanwhile, some users perform bad in the interservice time under max weight scheduling, as can be found in the third graph of Figure 4.

In Figure 5, we investigate the throughput performance of SGMW when the variance of arriving rate vector changes. The setting of the system is the same with that of simulation in Figure 3. The two policies are the same when $\delta$ is relatively small since the weight of any user can always grow to the maximum before the tight phase occurs. As $\delta$ keeps increasing, a growing number of users with small rate will only be scheduled in tight phase. Then the throughput performance of SGMW is no longer optimal.

As a result, we find that SGMW achieves interservice time guaranteed by potentially increasing the total and max queue length. SGMW will schedule users with very small queue length for meeting the interservice time constraints. This leads to the increasing of the max queue length since a user with max queue length may still wait for other users. The total length of the system is further increased when SGMW schedules the user with queue length smaller than its transmitting rate. But the total and max queue length are still finite in the $\epsilon$-capacity region.

6.3. Adaptability under Various Rate Vectors. As we can see, the performance of SGMW is mainly determined by the proportion of time that SGMW is in the tight phase and the number of users that are scheduled in this phase. We evaluate them in this subsection. There are 8 flows in the system, with arriving rate $\lambda_i = (1/(N + 2)) \times 0.9^i + (2/(N + 2)) \times 0.5^i$. The constraint $\epsilon$ varies from 2 to 32. In Figure 6, the number of intervals that have a tight phase decreases as $\epsilon$ grows. The tight phase disappears when $\epsilon \geq 25$. It indicates the fact that max weight scheduling policy can also ensure a loose constraint on interservice time. And SGMW is the same as max weight scheduling when $\epsilon$ is sufficiently large. In the second part of Figure 6, there is only one flow scheduled in the tight phase when $\epsilon = 24$, which means all the left flows can accumulate its weight to the maximum in a long loose phase.

7. Conclusion

This paper investigates the problem of interservice time guaranteed scheduling policy designing in wireless networks. We formalize the definition of modified capacity region with interservice time guarantee and propose a scheduling policy that meets the interservice time guarantee for all users. The scheduling policy is also proved to be throughput-optimal for the modified capacity region. Simulation results demonstrate that the scheduling policy has good performance on both the guarantee of interservice time and the stability of networks.
Figure 4: Throughput and interservice time performance under various users.

Figure 5: Throughput and interservice time performance under various arriving vectors.
Figure 6: Performance of SGMW under various interservice constraints.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

The work in this paper was partially supported by the National Natural Science Foundation of China (NSFC) under Grant nos. 61190115 and 61033015.

References

[1] L. Guo, Y. Li, and Z. Cai, “Minimum-latency aggregation scheduling in wireless sensor network,” Journal of Combinatorial Optimization, 2014.

[2] M. Ding, D. Chen, K. Xing, and X. Cheng, “Localized fault-tolerant event boundary detection in sensor networks,” in Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM ’05), pp. 902–913, March 2005.

[3] X. Cheng, A. Thaeler, G. Xue, and D. Chen, “TPS: a time-based positioning scheme for outdoor wireless sensor networks,” in Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM ’04), vol. 4, pp. 2685–2696, IEEE, March 2004.

[4] S. Cheng, J. Li, and Z. Cai, “O(ε)-Approximation to physical world by sensor networks,” in Proceedings of the 32nd IEEE Conference on Computer Communications (INFOCOM ’13), pp. 3084–3092, April 2013.

[5] J. Li, S. Cheng, H. Gao, and Z. Cai, “Approximate physical world reconstruction algorithms in sensor networks,” IEEE Transactions on Parallel and Distributed Systems, 2014.

[6] A. Boukerche, X. Cheng, and J. Linus, “Energy-aware data-centric routing in microsensor networks,” in Proceedings of the 6th ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM ’03), pp. 42–49, September 2003.

[7] Z. Cai, G. Lin, and G. Xue, “Improved approximation algorithms for the capacitated multicast routing problem,” in Computing and Combinatorics, vol. 3595 of Lecture Notes in Computer Science, pp. 136–145, Springer, 2005.

[8] L. Tassiulas and A. Ephremides, “Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks,” IEEE Transactions on Automatic Control, vol. 37, no. 12, pp. 1936–1948, 1992.

[9] X. Cheng, D.-Z. Du, L. Wang, and B. Xu, “Relay sensor placement in wireless sensor networks,” Wireless Networks, vol. 14, no. 3, pp. 347–355, 2008.

[10] X. Cheng, X. Huang, D. Li, W. Wu, and D.-Z. Du, “A polynomial-time approximation scheme for the minimum-connected dominating set in ad hoc wireless networks,” Networks, vol. 42, no. 4, pp. 202–208, 2003.

[11] I.-H. Hou, V. Borkar, and P. R. Kumar, “A theory of QoS for wireless,” in Proceedings of the 28th Conference on Computer Communications (INFOCOM ’09), pp. 486–494, IEEE, April 2009.

[12] I.-H. Hou and P. R. Kumar, “Admission control and scheduling for QoS guarantees for variable-bit-rate applications on wireless channels,” in Proceedings of the 10th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc ’09), pp. 175–184, New Orleans, La, USA, May 2009.

[13] I.-H. Hou and P. R. Kumar, “Scheduling heterogeneous real-time traffic over fading wireless channels,” in Proceedings of the IEEE INFOCOM, pp. 1–9, IEEE, March 2010.

[14] I. H. Hou and P. R. Kumar, “Utility-optimal scheduling in time-varying wireless networks with delay constraints,” in Proceedings of the 11th ACM International Symposium on Mobile Ad Hoc Networking and Computing, pp. 31–40, ACM, September 2010.

[15] J. J. Jaramillo and R. Srikant, “Optimal scheduling for fair resource allocation in ad hoc networks with elastic and inelastic traffic,” in Proceedings of IEEE INFOCOM, pp. 1–9, 2010.

[16] J. J. Jaramillo, R. Srikant, and L. Ying, “Scheduling for optimal rate allocation in ad hoc networks with heterogeneous delay constraints,” IEEE Journal on Selected Areas in Communications, vol. 29, no. 5, pp. 979–987, 2011.

[17] X. Kang, W. Wang, J. J. Jaramillo, and L. Ying, “On the performance of largest-deficit-first for scheduling real-time traffic in wireless networks,” in Proceedings of the 14th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc ’13), pp. 99–108, ACM, August 2013.
[18] R. Li, A. Eryilmaz, and B. Li, “Throughput-optimal wireless scheduling with regulated inter-service times,” in Proceedings of the 32nd IEEE Conference on Computer Communications (INFOCOM ’13), pp. 2616–2624, Turin, Italy, April 2013.

[19] B. Li, R. Li, and A. Eryilmaz, “Heavy-traffic-optimal scheduling with regular service guarantees in wireless networks,” in Proceedings of the 14th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc ’13), pp. 79–87, August 2013.

[20] N. K. Jaiswal, Priority Queues, Elsevier, New York, NY, USA, 1968.

[21] L. X. Bui, S. Sanghavi, and R. Srikant, “Distributed link scheduling with constant overhead,” IEEE/ACM Transactions on Networking, vol. 17, no. 5, pp. 1467–1480, 2009.

[22] J. Ni, B. Tan, and R. Srikant, “Q-CSMA: queue-length-based CSMA/CA algorithms for achieving maximum throughput and low delay in wireless networks,” IEEE/ACM Transactions on Networking, vol. 20, no. 3, pp. 825–836, 2012.

[23] X. Liu, E. K. P. Chong, and N. B. Shroff, “A framework for opportunistic scheduling in wireless networks,” Computer Networks, vol. 41, no. 4, pp. 451–474, 2003.

[24] S. Sakai, M. Togasaki, and K. Yamazaki, “A note on greedy algorithms for the maximum weighted independent set problem,” Discrete Applied Mathematics, vol. 126, no. 2–3, pp. 313–322, 2003.

[25] P.-J. Wan, O. Frieder, X. Jia, F. Yao, X. Xu, and S. Tang, “Wireless link scheduling under physical interference model,” in Proceedings of the IEEE INFOCOM, pp. 838–845, IEEE, Shanghai, China, April 2011.

[26] S. Basagni, “Finding a maximal weighted independent set in wireless networks,” Telecommunication Systems, vol. 18, no. 1–3, pp. 155–168, 2001.

[27] J.-H. Hoepman, “Simple distributed weighted matchings,” http://arxiv.org/abs/cs/0410047v1.

[28] C. Joo, X. Lin, J. Ryu, and N. B. Shroff, “Distributed greedy approximation to maximum weighted independent set for scheduling with fading channels,” in Proceedings of the 14th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc ’13), pp. 89–98, August 2013.