Scheduling Flows with Multiple Service Frequency Constraints

Xu Zheng, Zhipeng Cai, *Senior Member, IEEE,* Jianzhong Li, and Hong Gao

Abstract—With the fast development of wireless technologies, wireless applications have invaded various areas in people's lives with a wide range of capabilities. Guaranteeing Quality-of-Service (QoS) is the key to the success of those applications. One of the QoS requirements, service frequency, is very important for tasks including multimedia transmission in the Internet of Things. A service frequency constraint denotes the length of time period during which a link can transmit at least once. Unfortunately, it has not been well addressed yet. Therefore, this paper proposes a new framework to schedule multi trans-mitting flows in wireless networks considering service frequency constraint for each link. In our model, the constraints for flows are heterogeneous due to the diversity of users' behaviors. We first introduce a new definition for network stability with service frequency constraints and demonstrate that the novel scheduling policy is throughput-optimal in one fundamental category of network models. After that, we discuss the performance of a wireless network with service frequency constraints from the views of capacity region and total queue length. Finally, a series of evaluations indicate the proposed scheduling policy can guarantee service frequency and achieve a good performance on the aspect of queue length of each flow.

Index Terms—IoT; Scheduling Plicy; Service Frquency; QoS

I. INTRODUCTION

During the past decade, the diversity of wireless network architectures manifested a tremendous growth with the im-provements of mobile devices and wireless communication techniques. Novel systems like Internet of Things (IoTs) or Cyber Physical Systems (CPSs) [1] have changed a wire-less network from a simple data transmitting platform to a complicated one. Such systems often request the support of real-time multimedia data transmission like Video on demand (VoD) and voice over IP (VoIP) [2], which pose various Quality-of-Service (QoS) requirements for including packet delay constraints, delivery ratio, traffic admission control, link service regularity, *etc.* for link scheduling in wireless networks. Motivated by this, a series of works have emerged which focus on scheduling traffic considering QoS constraints. Many of them consider the transmission of packets with stringent delay constraints [3], while some others focus on delivery ratio [4], admission control, *etc*. Surprisingly, service regularity, one of the most important QoS requirements for many streaming multi-media applications, has not been fully explored. To the best of our knowledge, only few works [5] [6] concentrate on

X. Zheng and Z. Cai are with the Department of Computer Science, Georgia State University, 25 Park Place, Atlanta, GA 30303, USA. Email: {xzheng5, zcai}@gsu.edu

J. Li and H. Gao are with the School of Computer Science, Harbin Institute of Technology, Harbin, 150001, China. Email: {lijzh, honggao}@hit.edu.cn

the link scheduling problem considering the service regularity requirement.

Service regularity refers to the time between two consec-utive services of a link. It indicates a link will not wait for too much time for the next service and the waiting time between every consecutive transmission pair is relatively identical. In traditional data-oriented wireless networks, a link has no constraint on service regularity as long as its overall throughput is guaranteed. Such tolerance no longer exists in the current IoTs carrying out multimedia data. For example, in some applications, videos are uploaded by the citizens to report emergencies around them. The officers at the controlling center request the videos to be presented smoothly to support accurate and rapid decisions. It indicates that the video blocks from each uploader should be transmitted periodically to guarantee the continuous playback of the video. This puts forward a service regularity requirement for the scheduling policy. Li *et al.* [5] introduced a new parameter called *time-since-last-service* to improve service regularity. The time-since-last-service indirectly refers to the time between two consecutive transmissions of a link. Their work balances the long-term average inter-service time of each user while still maintains the stability of the network. However, it fails in the situation when the service regularity of each link is strictly bounded. In this paper, we fill this gap by designing a novel scheduling policy. Within our framework, service frequency requirement, a variant of service regularity, of all the links can be completely satisfied.

Service frequency refers to the length of time during which a link obtains a chance to transmit. A constraint on service frequency indicates that a link should be scheduled at least once in every frame if we partition time into frames with length equaling to the constraint. Service frequency is closely related to service regularity in that it can bound the waiting time between two consecutive transmissions of a link, and can also lead to a stable waiting time if the links are properly scheduled. Service frequency is also essential for multimedia data transmission. Again, notice that the smoothly real-time playback of videos or voices are very important for some crowdsourcing monitoring systems, and such multimedia data is played in a batched manner [7]. An indispensable QoS requirement is to deliver the next block to the user before the end of her current one. *i.e.*, the user must get a new block in each frame whose length equals to the playing time of a block.

Furthermore, the service frequency requirements of users are often heterogeneous due to the diversity of users' be-haviors. For example, users may upload various types of

multimedia data or videos with different resolutions. Higher quality indicates larger service frequency when the size of a packet is fixed. To the best of knowledge, this is the first work to investigate the scheduling of flows in wireless networks with heterogeneous service frequency constraints.

Service frequency requirements in fact make a scheduling policy a combination of the round-robin and maximum-throughput methods. Users, on the one side, upload their multimedia data to participate in the monitoring tasks. This is like the round-robin scheduling, where each user is served periodically. At the same time, all users also hope their own applications running normally on mobile devices, which means the ones with longer packet queue must have higher priority to be served. This necessitates maximum-throughput scheduling. In this paper, we introduce a new definition for capacity region with service frequency constraints. The novel capacity region is the set of packet-arriving rate vectors in which no user will suffer from buffer overflowing and all the constraints can be satisfied. This new definition can also be utilized to judge the performance of a scheduling policy. Furthermore, we propose a novel scheduling policy that considers service frequency constraints, and the policy is throughput-optimal in some types of networks. Our main contributions are summarized as follows:

- We propose a novel network model accommodating service frequency constraints, which is more suitable for many real-time wireless applications. We also provide new definitions of the stability of a system and the capacity region.
- We design a new scheduling policy. The policy is very easy to be implemented. Furthermore, we prove that the policy is throughput-optimal in an extensively utilized category of network model.
- We compare the size of the new capacity region with the original one without the service frequency constraints. Meanwhile, we analyze the traffic load in a wireless network when a system is stable.
- Simulation results reveal that our scheduling policy achieves a good performance in both queue length and service regularity.

The rest of the paper is organized as follows. In Section 2, we have a literature review on the development of scheduling policies in wireless networks. Section 3 introduces the network model. In Section 4, the novel scheduling policy is proposed with some theoretical analysis. Section 5 presents the performance of a wireless network with service frequency constraints. The simulation results of our scheduling policy are illustrated in Section 6. Section 7 concludes the paper.

II. RELATED WORKS

Scheduling is a fundamental issue for wireless networks. It is well-known that the Maximum Weight scheduling policy is throughput-optimal [8]. The weight is usually denoted as the ratio between queue length and transmitting rate. Gupta and Shroff [9] showed that a generalized maximum weight scheduling could also be throughput-optimal. The policy scaled the queue length of each link by a weighted factor.

Some work [10] considers queue length and delay under the max weight scheduling policy.

However, the problem of finding a disjoint set of links with the maximum weight under the general interference model is NP-Hard. Hence, a maximum weight scheduling policy incurs high overhead. Much effort has been spent on designing low complexity scheduling policies with throughput performance guarantees. Some policies are based on the *picking-and-comparing method* [11] or *Carrier Sensing Multiple Access (CSMA)* [12]. Those policies gradually achieve the optimal throughput after many iterations, but are insensitive to channel fading. Lin and Shroff [13] proved that the maximum weight scheduling can achieve a guaranteed ratio of the capacity region. The performance is lower bounded by the ratio bound of the weight of the maximal independent set. Since then, many scheduling policies focus on finding a *maximal weight independent set (MWIS)* of links. The works in [14] and [15] present some centralized policies that can find a MWIS with provable ratio bounds. The works in [16] proposes some distributed scheduling policies for different network models.

The QoS requirements in link scheduling are also considered by many works for wireless networks. Some scheduling policies are designed to meet the packet delay constraint for each user [17][3]. They all partition time into frames, and packets will run out of time before the end of their arriving frames. They mainly differ in the ways of dealing with channel fading [19], nonidentical delays [3], *etc*. The work in [4] proposes a new policy that accommodates packet delivery ratio.

Very few works [5] [6] deal with service regularity. The work in [5] introduces a new type of parameter to capture the performance of service regularity, which is called time-since-last-service. It enriches the definition of weight to include the changing of this parameter. The weight is a combination of queue length and *time-since-last-service*. Then under the maximum weight scheduling policy, a user will get a second service with a growing probability as *time-since-last-service* increases, thus the inter-service time can be regulated. Li *et al*. [6] extended their work to consider the inter-service time performance of rate vectors close to the edge of capacity region. This work could still achieve the original capacity region, and only relatively improve the performance of service regularity. Actually, this work can achieve an approximate guarantee on service regularity. This is done by carefully choosing the parameters according to a comprehensive knowledge of users' behaviors, which is impossible in practice. It would fail when the service regularity of each user is exactly bounded and the users' behaviors change often. Besides, it cannot figure out the new capacity region in which the users follow strict service frequency requirements and the system is stable. This motivates us to design a novel scheduling policy which can satisfy the heterogeneous service frequency of each user.

III. NETWORK MODEL AND PRELIMINARIES

A. Network Model

A network is composed of N links, denoted as $\{l_1, l_2, \cdots, l_N\}$. They transmit via a common wireless channel. Each link connects one source node to one destination

node. Due to the inherent interference in wireless networks, two links can not transmit simultaneously when they share the same node or are geographically close to each other. The conflicts among links are presented by **conflict graph** $G_{NN} = \{g_{ij}\}_{i,j=1,2,\cdots,N}$ with binary entry. $g_{ij} = 1$ means link l_i and l_j conflict with each other and can not transmit simultaneously, and $g_{ij} = 0$ otherwise. Time is partitioned into slots, and in each slot a link can successfully transmit only when all the other links remain waiting.

In each slot, there are $A_i[t]$ packets arriving at link i. $A_i[t]$ is randomly and identically distributed over time with expectation $E(A_i[t]) = \lambda_i$. The second moments of $A_i[t]$ are finite for all $i = 1, 2, \cdots, N$ and $t = 1, 2, \cdots$. These packets will join the delivery queue of link i and be delivered to the destination. We assume that the wireless channel is unreliable. The channel condition of link l_i refers to the probability c_i that l_i can successfully transmitted in a slot. Then bandwidth for each link is represented as the number of delivered packets, *i.e.*, link l_i can deliver r_i packets per slot when succeeded. Finally, each link has a service frequency constraint δ_i, which means link i needs to be scheduled at least once in every frame with δ_i slots. In our model, λ_is, c_is, r_is, and δ_is are all heterogeneous, which indicates the fact that the traffic load, communication capability and QoS requirements of different mobile devices are often various. In this paper, we consider the case when the service frequency constraints are fixed, and deal with different arriving rates.

We use boolean variable $s_i[t] \in \{0, 1\}$ to denote whether user i is scheduled in slot t, with 1 for scheduling and 0 for waiting. The vector $S[t] = \{s_1[t], s_2[t], \cdots, s_N[t]\}$ is called a scheduling for slot t. In our model, it is guaranteed that

$$\sum_{i=1}^{N} s_i[t]$$

is maximal and all chosen links does not conflict with each other for every time slot t.

$Q_i[t]$ is the time consumption of transmitting all packets on link i at the beginning of slot t. $Q_i[t]$ is determined by $Q_i[t-1]$, $\frac{A_i[t]}{r_i c_i}$, and $s_i[t-1]$ as follows:

$$Q_i[t] = Max\{Q_i[t-1] - s_i[t-1], 0\} + \frac{A_i[t]}{r_i c_i} \quad (1)$$

where 0 means that user i has delivered all its packets in slot $t-1$. $Q_i[t]$ is closely related to the packet size on link i, while it can simplify our analysis.

Now we formalize the stability of networks by the following definitions.

Definition 1. *A network is stable under an arriving rate vector* $\lambda = \{\lambda_1, \lambda_2, \cdots, \lambda_N\}$ *when*

(1) the underlying Markov chain $Q_i[t]$ for each link is positive recurrent, and

$$\lim_{t \to \infty} \sum_{i=1}^{N} E(Q_i[t]) \le \infty, \quad (2)$$

(2) each link i is scheduled at least once in every δ_i slots.

We also use the term $\lim_{t \to \infty} \frac{\sum_{t'=1}^{t} \sum_{i=1}^{N} Q_i[t']}{t} \le \infty$ in this paper, which is actually the same as Formula (2). A scheduling

policy refers to a series of scheduling $S[1], S[2], S[3], \cdots$, one for each time slot. An arriving rate vector $\lambda = \{\lambda_i\}$ ($i = 1, \cdots, N$) is *supportable* by a network if the network is stable under λ when it utilizes some scheduling policies. All supportable rate vectors compose the *capacity region* of the network, denoted by Λ. The throughput-optimality of a scheduling policy is defined as follow:

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

We will show in next section that to design a throughput optimal solution is NP-hard. Our object is to design a scheduling policy that can achieve a good performance on throughput while maintaining a polynomial time complexity.

B. Preliminaries

Now we introduce some knowledge regarding the positive recurrent of Markov chains and the upper bounds on queue length of some well-known scheduling policies. There exist a lot of methods which prove the positive recurrent of Markov chains. In this paper, we use the Foster's criteria for positive recurrent and ergodic Markov chains [20], which is shown as follows.

Theorem 1. *A countable Markov chain $Q[t] = \sum_{i=1}^{N} Q_i[t]$ is positive recurrent and ergodic if and only if there exists a positive function $V > 0$ and a finite set of state ϵ_0, such that*
(1) when $Q[t] \in \epsilon_0$, $\Delta(Q[t]) < \infty$,
(2) when $Q[t] \notin \epsilon_0$, $\exists \xi > 0$, and $\Delta(Q[t]) < -\xi$,
where

$$\Delta(Q[t]) = E(V(Q[t+1]) - V(Q[t])|Q[t]). \quad (3)$$

There also exist many results on the queue length of a network with the maximum weight scheduling policy. Our work is not dominated by the choice of the result. Thus, this paper uses a simple bound given in [9].

Theorem 2. *Given a rate vector $\lambda = \{\lambda_i\}$ ($i = 1, \cdots, N$) inside the capacity region, the following bound on the expectation of the sum of queue length is true in a system under the maximum weight scheduling policy:*

$$\sum_{i=1}^{N} E(Q_i) \le \frac{\lambda_i + Var[A_i] - \lambda_i^2}{2}$$

IV. MULTI-STAGE MAXIMUM WEIGHT SCHEDULING POLICY

In this section, we first prove the complexity of the problem and propose a new scheduling policy which is a combination of round-robin and maximum-weight scheduling. In our scheduling policy, links are first categorized into different stages indicating the priority level this link should be served. This mechanism can ensure the service frequency constraint of each link. Meanwhile, a maximum weight scheduling policy is used for allocating the service among links of the same stage, resulting in a link with longer queue size to be served as soon as possible. We call this framework *Multi-Stage Maximum Weight* scheduling policy (MSMW for short).

A. Time Complexity of Throughput-optimal Solutions

The design of throughput-optimal policy is NP-complete. We consider the case where $\delta_i \to \infty$ for all links. Then our problem is same with the basic scheduling problem in general wireless networks, which is proved to be NP-complete [9]. To achieve an optimal solution means the algorithm must find a max weight independent set in a general graph in each slot, which can only be solved in exponential time.

B. MSMW Scheduling Policy

MSMW needs to take care of service frequency and queue length of each link. Therefore, it has two parameters: *stage* and *weight*.

stage shows whether a link has been scheduled in the current frame and the number of time slots to the end of the frame. The stage of link i in slot t is calculated as follows:

$$ST_i[t] = \begin{cases} 0 & \text{when link } i \text{ is scheduled} \\ & \text{in the current frame} \\ \delta_i - (t-1) \bmod \delta_i & \text{others} \end{cases}$$

$$(4)$$

where δ_i is the service frequency constraint of link i. In each slot, MSMW first considers the links with small stages which have not been scheduled. Then MSMW can continue in a round-robin manner since unscheduled links will eventually obtain a higher priority.

weight represents the traffic load of each link, as is used by most of the previous works. MSMW serves the link with the largest weight when multiple links are in the same stage. We define *weight* of link i at time slot t as follows:

$$w_i[t] = Q_i[t] \qquad (5)$$

We will use $w_i[t]$ and $Q_i[t]$ alternately in the rest of the paper.

In each time slot, MSMW has two phases:

Phase 1 : Update

At the beginning of each time slot, MSMW updates *stage* and *weight* for each link with Equations (4) and (5), based on the transmission information in the last slot and the arriving packets in the current slot. Then the policy partitions the links into different groups according to their stages. We denote the groups as $GP = \{gp_0, gp_1, gp_2, \cdots, gp_M\}$. The links in a same group have identical stage. We also assume that the stage of links in each group increases with the group number, *i.e.*, gp_1 includes the links with the minimum stage while gp_M contains the links with the largest stage. gp_0 is composed of the links whose stages are zero. All the links will fall in gp_0 when they have been scheduled in the current frame.

Phase 2: Scheduling

MSMW iteratively selects the links from group gp_M when $M = 0$, or from gp_1 to gp_M and then gp_0 when $M \neq 0$. During each iteration, MSMW chooses a link from first group containing links with following equation,

$$l_i = argmax_{l_j \in gp_k} w_j[t], \qquad (6)$$

where gp_k is the first group still has links remaining. After the selection, MSMW deletes all the conflicted links of l_i and starts next iteration. The scheduling phase ends when there is no link left in current slot. ∎

Now we use a simple example to demonstrate the scheduling policy. Suppose there are 2 links in the network, with parameters $A_1[t] = \lambda_1$, $A_2[t] = \lambda_2$, $\frac{\lambda_1}{r_1 c_1} = \frac{1}{2}$, and $\frac{\lambda_2}{r_2 c_2} = \frac{1}{8}$. $\delta_1 = 2$ and $\delta_2 = 4$. In slot 1, $ST_1[1] = 2$ and $ST_2[1] = 4$. Then l_1 is scheduled since it has a smallest stage. In slot 2, $ST_1[2] = 0$ and $ST_2[2] = 3$. Then l_2 is chosen. In slot 3, $ST_1[3] = 2$ since a new frame of l_1 begins. $ST_2[3] = 0$. Then l_1 is selected. In slot 4, $ST_1[4] = 0$ and $ST_2[4] = 0$. $w_1[4] = 1/2$ and $w_2[4] = 1/4$. Then l_1 is scheduled since it has a larger weight. The following process is similar, thus omitted here.

The computational complexity of MSMW is analyzed as follows. The time complexity of updating all the stages and weights is $O(N)$. Partitioning links into groups also costs $O(N)$ scale of time. During phase 2, the time complexity is $O(N^2)$ in the worst case, and $O(1)$ when there are constant number of links in the first group. As a whole, the time complexity of MSMW in each time slot is $O(N^2)$.

C. Performance of MSMW in Collocated Networks

In this part, we demonstrate that MSMW is throughput optimal in a specific type of networks, *i.e.*, collocated networks [5]. In a collocated network, all links conflict with each other, and only one link can be scheduled in each slot. This model is used by systems like cellular networks, which is an fundamental architecture to support the environment monitoring systems. We first show in Lemma 2 that all the supportable rate vectors follow the second condition in *Definition 1* when a collocated network employs the MSMW scheduling policy. Then in Lemma 5, all these vectors are proved to follow the first condition, which means the underlying queue length is finite.

We proposes a necessary condition for all the arriving rate vectors in the capacity region. This condition means that for a sufficient long time, the number of slots used to meet the service frequency constraints of all the links is no more than the total number of time slots.

Lemma 1. *The service frequency constraints of all the links must follow equation*

$$\sum_{i=1}^{N} \frac{1}{\delta_i} \leq 1, \qquad (7)$$

if the rate vector is supportable by the network.

Proof. The proof is by contradiction. Assume that $\sum_{i=1}^{N} \frac{1}{\delta_i} > 1$. Then the total services that all the links get are no less than $\sum_{i=1}^{N} \frac{T}{\delta_i}$ in T time slots. This is due to the fact that each link i gets a service every δ_i slots. However, it is infeasible since $\sum_{i=1}^{N} \frac{T}{\delta_i} = T \sum_{i=1}^{N} \frac{1}{\delta_i} > T$. It contradicts the fact that the arriving rate vector is supportable. □

Lemma 2 shows that MSMW can ensure the constraints when the arriving rate vector follows Lemma 1.

Lemma 2. *The service frequency constraints of the links in a network is achievable under MSMW if $\sum_{i=1}^{N} \frac{1}{\delta_i} \leq 1$.*

Proof. Assume link l_p does not get service in the kth interval that starts at slot t_k.

Consider an arbitrary link l_i. It will not receive service in the interval that includes the final slot of kth interval of l_p. The reason is that l_p always has a smaller stage than l_i in that interval of link l_i. It will only be scheduled once in a interval if the time period is totally covered by the kth interval of l_p. If l_i has a jth interval that covers slot t_k and its scheduling in this interval happens no early than slot t_k, all the other links can only be scheduled once per interval during the slots of jth interval for l_i.

We can iteratively look for such intervals back and find the first link l_x and its yth interval follow either of the following principals: 1) all the other links have new intervals that start at the same slot with yth interval of link l_x, 2) all the intervals that cover t_y are scheduled early than slot t_y.

Now we consider the time between t_y and $t_k + \delta_p$. Every link can only be scheduled once in each interval that is totally covered by this time period. This result can be combined with the fact that link l_p does not get service in the kth interval:

$$\sum_{i=1}^{N} \lfloor \frac{t_k + \delta_p}{\delta_i} \rfloor - 1 \leq \sum_{i=1}^{N} \frac{t_k + \delta_p}{\delta_i} - 1 \leq t_k + \delta_p - 1 < t_k + \delta_p$$

Then the number of feasible services is less than the size of time slots, which is a contradiction. It indicates a link could be scheduled at least once per interval when $\sum_{i=1}^{N} \frac{1}{\delta_i} \leq 1$. $\square$

MSMW can satisfy the second condition in *Definition 1* for all the rate vectors inside the capacity region since it is a subset of the vectors following Lemma 1.

Now we analyze the performance of MSMW on the first condition in *Definition 1*. Our proof is based on the conclusion in *Theorem 1*. MSMW actually runs in two manners: the maximum weight manner when it chooses a link from gp_0, and the round-robin manner otherwise. As we know, the maximum weight scheduling is throughput-optimal since it always chooses the link with the globally maximum weight. Our work turns to prove that the queue length will never be infinite when we scheduling a link not in gp_0. Specially, we need to discuss the performance of the system when the queue length of scheduled link is less than its transmission rate.

We start with another necessary condition for any supportable rate vector. It bounds the size of a packet that cannot be delivered if every link only transmits once in every frame. Notice that Lemma 3 is not restricted to MSMW.

Lemma 3. *For any supportable rate vector, it must follow that $\sum_{1}^{N} Max\{\frac{\lambda_i}{r_i c_i} - \frac{1}{\delta_i}, 0\} < 1 - \sum_{1}^{N} \frac{1}{\delta_i}$.*

Proof. The proof is straightforward. For a sufficient long time T, the expectation of the size of a packet that fails to be delivered equals to $\sum_{1}^{N} Max\{T \times \lambda_i - \frac{T \times r_i c_i}{\delta_i}, 0\}$ if each link is only scheduled once in every frame. Delivering these packets totally costs $T \times \sum_{1}^{N} Max\{\frac{\lambda_i}{r_i c_i} - \frac{1}{\delta_j}, 0\}$ slots. Meanwhile, the number of slots left is $T - \sum_{1}^{N} \frac{T}{\delta_i}$. Thus, $\sum_{1}^{N} Max\{\frac{\lambda_i}{r_i c_i} - \frac{1}{\delta_i}, 0\} < 1 - \sum_{1}^{N} \frac{1}{\delta_i}$ must hold. $\square$

In Lemma 4, we show that the expectation of the total queue length is bounded in slot t when a link i with $ST_i[t] \neq 0$ is scheduled. Denote T_0 as the least common multiple of $\delta_1, \delta_2, \cdots, \delta_N$.

Lemma 4. *When MSMW schedules a link not in gp_0 in slot t, the expectation of the total queue length $E(\sum_{i=1}^{N} Q_i[t]) \leq B$, where $B = N + h * B_1 + \sum_{i=1}^{N}(\sum_{j=1}^{N} \frac{1}{\delta_j} * \frac{\lambda_i}{r_i c_i} * T_0)$. B and h are constants, and $B_1 = \frac{1}{2(1-\sum_{j=1}^{N} \frac{1}{\delta_j})^2} \sum_{l_i \in \{l_j | \frac{\lambda_j}{r_j c_j} - \frac{1}{\delta_j} > 0\}}[(1 - \sum_{j=1}^{N} \frac{1}{\delta_j})(\frac{\lambda_i}{r_i c_i} - \frac{1}{\delta_i}) + Var(\frac{A_i[t]}{r_i c_i} - \frac{1}{\delta_i}) - (\frac{\lambda_i}{r_i c_i} - \frac{1}{\delta_i})^2]$.*

Proof. The proof of lemma is composed of two parts. The first part estimates the total queue length of the system at slot $k * T_0$, where k is a positive integer. The second part gives the upper bound of $E(\sum_{i=1}^{N} Q_i[t])$ when MSMW schedules a link not in gp_0. The bound also holds for the time when MSMW selects a link in gp_0. We discard this case since it has nothing to do with our subsequent proofs. The details of the proof is given in Appendix in our technique report [21]. $\square$

According to Lemma 4, the total queue length of the system is bounded by a constant B when MSMW schedules a link in $gp_i, i \neq 0$. Now we can prove that for any supportable rate vector, the underlying Markov chain is positive recurrent under MSMW. The proof is based on the Foster's criteria in Theorem 1.

Lemma 5. *For any supportable arriving rate vector, the underlying Markov chain $Q[t] = \sum_{i=1}^{N} Q_i[t]$ is positive recurrent when we apply MSMW to schedule the links.*

Proof. In this proof, we use the Lyapunov function defined as

$$V(Q[t]) = \frac{1}{2} \sum_{i=1}^{N} Q_i[t]^2.$$

to show the underlying Markov chain $Q[t] = \sum_{i=1}^{N} Q_i[t]$ is positive recurrent. The details of our proof is shown in our technique report [21]. $\square$

MSMW can support all the rate vectors in the capacity region by combining Lemma 2 and Lemma 5. We conclude it in the following theorem, and the proof is straightforward.

Theorem 3. *MSMW is a throughput-optimal scheduling policy under the proposed network model and the definition of stability.*

V. DISCUSSION

In this section, we continue our analysis on the performance of a collocated network with the service frequency constraints. Our objectives are mainly on three aspects. In part A, we consider the change of the network performance from an overall view. We discuss the change on the set of supportable arriving rate vectors when introducing a group of fixed service frequency constraints. We compare the size of the new capacity region with the original region without the constraints. In part B, we aim to check the in-depth performance of the system with service frequency constraints. We check the total queue length of the system under a rate vector that is supportable both with or without the constraints.

A. Decreasing in Capacity Region

The constraint on service frequency reduces the size of capacity region. This is due to the fact that some links with few packets can still be scheduled to meet the constraints, and then the bandwidth is wasted. Other links have to wait and the length of their queues could possibly be infinite. A larger $\sum_{i=1}^{N} \frac{1}{\delta_i}$ often leads to larger decrease in capacity region since more slots will be used by the links not in gp_0 to keep the service frequency. We formulate the decrease in Theorem 4. Lemma 6 estimates the size of the vector space composed of all vectors $k = (k_1, k_2, \cdots, k_N)$ that follow $k_i \geq 0, \forall i$ and $\sum_1^N k_i \leq \beta$. The conclusion in Lemma 6 is applied for the proof of Theorem 4.

Lemma 6. *For any vector* $k = (k_1, k_2, \cdots, k_n)$, $\oint_0^\beta 1 d\sigma = \frac{\beta^n}{n!}$ *if* $k_i \geq 0, \forall i$ *and* $\sum_1^N k_i \leq \beta$. $\oint_0^\beta 1 d\sigma$ *is a multiple integration on the* n *dimensions, also defined as the size of the vector space composed of all vectors like* k.

Proof. We prove the lemma by induction.

When $n = 1$, $\oint_0^\beta 1 d\sigma = \beta$. When $n = 2$, $\oint_0^\beta 1 d\sigma = \frac{\beta^2}{2}$.

Assume $\oint_0^\beta 1 d\sigma = \frac{\beta^k}{k!}$ is true when $n = k$. When $n = k+1$,

$$\oint_0^\beta 1 d\sigma = \int_0^\beta \oint_\beta^0 1 d\sigma' dx \tag{8}$$

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

$$\int_0^\beta \oint_\beta^0 1 d\sigma' dx = \int_0^\beta \frac{(\beta - x)^n}{n!} dx = -\int_\beta^0 \frac{(\beta - x)^n}{n!} d(\beta - x)$$
$$= -\int_\beta^0 \frac{(y)^n}{n!} d(y) = -\frac{1}{(n+1)!}(y)^{n+1}|_\beta^0$$
$$= \frac{\beta^{n+1}}{(n+1)!}$$

$$\tag{9}$$

□

Theorem 4 gives the ratio between the new capacity region and the original one without service frequency constraints.

Theorem 4. *The size of capacity region* Λ *of a network is no more than* $N! \sum_{t=0}^{N} C_N^t [\frac{\epsilon^t}{t!} (\frac{1}{\delta_{min}})^{N-t}]$ *of its original capacity region* Λ_0, *where* $\epsilon = 1 - \sum_1^N \frac{1}{\delta_i}$, *and* δ_{min} *is the minimum service frequency constraint among all the links.*

Proof. The sketch of this proof is to estimate the size of vector space for both Λ and Λ_0. The details of the proof is included in our technique report [21]. □

Theorem 4 shows that the scale of the capacity region keeps increasing as ϵ becomes larger. When $\epsilon \to 1$, we have $\sum_1^N \frac{1}{\delta_i} \to 0$, then $\sum_{t=0}^{N} \sum_{i=1}^{|C_t|} [\frac{\epsilon^t}{t!} \prod_{j \notin C_{ti}} (\frac{1}{\delta_j})] \to \frac{1}{N!}$. It means that $\Lambda/\Lambda_0 = 1$ when the service frequency constraints are sufficiently loose for all the links. On the other hand, when $\epsilon \to 0$, we have $\sum_1^N \frac{1}{\delta_i} \to 1$, then $\sum_{t=0}^{N} \sum_{i=1}^{|C_t|} [\frac{\epsilon^t}{t!} \prod_{j \notin C_{ti}} (\frac{1}{\delta_j})] = \prod_{i=1}^{N} (\frac{1}{\delta_i})$. $\prod_{i=1}^{N} (\frac{1}{\delta_i})$ is maximized when all the δ_i are identical. It indicates that with tough service frequency requirements, the capacity region is maximized when all the links have same constraints $\delta = N$, which refers to a strict round-robin scheduling.

B. Performance on Queue Length

The queue length of a system under an arriving rate vector usually increases with the existence of service frequency constraints. This is aroused by the scheduling of the links with very small queue length. These links cannot take full use of the channel, which will lead to the increasing of the global queue length. We estimate an upper bound on the size of the total queue length of the links under fixed $\{\lambda_1, \lambda_2, \cdots, \lambda_N\}$ and $\{\delta_1, \delta_2, \cdots, \delta_N\}$.

Theorem 5. *For an arriving rate vector* $\lambda = \{\lambda_1, \lambda_2, \cdots, \lambda_N\}$, *and service frequency constraints* $\{\delta_1, \delta_2, \cdots, \delta_N\}$, *the expectation of the average total queue length in a long time follows* $\lim_{t \to \infty} \frac{\sum_{t'=1}^{t} \sum_{i=1}^{N} Q_i[t']}{t} \leq B_3$ *if the system employs MSMW. Assume* $\epsilon = 1 - \sum_{i=1}^{N} \frac{1}{\delta_i}$. *Then* $B_3 = \frac{N+1}{2} + \frac{N'}{8} + \frac{1}{2\epsilon^2} \sum_{l_i \in L'} \frac{Var(A_i[t])}{(r_i c_i)^2}$ *when* $\epsilon > 0$, *where* L' *is the set of links with* $\frac{\lambda_j}{r_j c_j} - \frac{1}{\delta_j} > 0$, N' *is the size of* L'. $B_3 = \sum_{i=1}^{N} (Min\{\frac{\lambda_i}{r_i c_i}, \frac{1}{\delta_i}\} * \frac{(1+\delta_i)}{2})$ *when* $\epsilon = 0$.

Proof. See Appendix B in our technique report [21]. □

As we see, the queue length is affected by many factors, like the number of the total links, the sum of $\frac{1}{\delta_i}$'s and so on. To derive the accurate influence of these factors will be one of our future works.

VI. EVALUATION

In this section, we evaluate MSMW through a group of simulations. We mainly compare MSMW with a state-of-the-art scheduling policy *Regulated Throughput Optimal* (*RTO*) [5]. RTO is also a combination of the two fundamental scheduling policies: max weight scheduling and round-robin scheduling. RTO serves the link with the maximum weight in each slot. The weight of each link is a combination of queue length $Q_i[t]$ and the time-since-last-service $T_i[t]$. The time-since-last-service is the number of time slots that start from the link's last transmitting. There are two weighting factors, α_i and β_i, for the two components in the weight. We set $\alpha_i = \frac{1}{r_i c_i}$ and $\beta_i = \frac{1}{\delta_i}$ for each link, *i.e.*, $w_i[t] = \frac{1}{r_i c_i} Q_i[t] + \frac{1}{\delta_i} T_i[t]$. Since RTO only applies to collocated networks, we detailedly compare MSMW with RTO in this type of network. Furthermore, we also give the performance of MSMW in general networks. In every group of simulations, we run each scheduling policy ten times, with 10000 slots each time.

Our evaluation of MSMW mainly focuses on two metrics: the performance on service frequency constraints and the total queue length of the system. As proved, MSMW can keep the service frequency constraints for all the links. We are also interested in the performance of RTO on the service frequency, as it can regulate the services among links. Meanwhile, we check the queue length since it can tell the stability of the system. MSMW ensures service frequency at the cost of increasing total queue length. Such a cost is tolerable in our simulation results.

A. Number of Links

The first group of simulations test MSMW and RTO with different numbers of links, which are 4, 8, 16, 32, and 64. The service frequency constraint of each link is $N+1$, where N is the number of the links. We set $r_i = 2$ and $c_i = 0.5$ for link i ($i = 1, 2, \cdots, N$) and $E(A_i) = \frac{1}{N*i}$ for the ith link. The arriving rates are asymptotically decreasing. The result is shown in Fig.1.

As can be seen, MSMW can satisfy the service frequency constraints for all the links, while the ratio of the links that satisfy the constraints in RTO decreases as N increases. It falls from 1 (4 out of 4 links) to 0.109375 (only 7 out of 64 links). It means that more and more links have slowly increasing weights in RTO due to their small arriving rates. Their weights are not large enough to always be selected by the RTO within the service frequency constraints. The performance of MSMW is not affected since it is independent of the arriving rates.

At the same time, we find that the total queue length and maximum queue length of MSMW are comparable with those of RTO. It is only a bit larger (less than 0.2 for the total queue length and 0.3 for the maximum queue length). The increasing is mainly caused by the scheduling towards the links with very small queue length, which potentially wastes bandwidth. The queue lengths are equal when $N = 4$, which means the two policies have identical performance.

B. Service Frequency Constraints

The second group of simulations evaluates the performance of MSMW with different service frequency constraints. There are 32 links in the network, with $E(A_i) = \frac{1}{N*i}$ for link i. We set the service frequency constraint to be $N+k$, and $k = 1, 2, 3, 4, 5$. The result is shown in Fig.2.

We find that the performance of RTO rapidly improves as k increases, while MSMW always satisfies all the constraints. The reason for the improvement in RTO is that the selected α_i's and β_i's happen to be more fitful for larger k's. It is true that RTO can keep the service frequency if we carefully set the parameters according to the fixed arriving rates and service frequency constraints. The queue length of MSMW is still comparable with that of RTO, as shown in Fig.2.

C. Arriving Rates

In the third group of simulations, we evaluate the performance with different arriving rate vectors. There are in total 16 links in the network, and the service frequency constraint is $N + 5$ for all the links. We set the arriving rates of the first 8 links to be $E(A_i) = \frac{1}{N*i}$, and the last 8 links to be $E(A_i) = \frac{1}{N*(i+k)}$ ($k = 1, 2, \cdots, 20$). We call this rate vector two tails since the two parts of the links have different scales of arriving rates. The difference between the two parts becomes larger as k increases. We show the results in Fig.3.

MSMW can still satisfy the constraints in all the situations, while RTO's performance degrades as k increases. RTO can only ensure service frequency for the first part when $k \geq 17$. The queue length of MSMW is larger than that of RTO by a small fraction (around 10%).

D. Service Regularity

For service regularity, we use the standard deviation of the inter-service time as our measurement. The inter-service time is the duration between two consecutive transmissions of a link. A smaller standard deviation of inter-service time indicates a link will approximately wait for a same time between every two consecutive services. It means that the link has a good performance on service regularity.

There are 16 links in our simulations, with service frequency constraint $N+5$. The arriving rates are set as follows: $E(A_i) = \frac{1}{N*(i+k)}$ ($k = 1, 2, \cdots, 20$) for link i. The result is shown in Fig.4.

As can be seen, while the maximum standard deviations of MSMW and RTO are nearly identical, the average standard deviation of MSMW is much better. It is only about $\frac{1}{3}$ of that of RTO. MSMW has a better performance on the aspect of service regularity. Such an improvement is because of the fact that the service frequency constraints are mainly ensured by ST_i. The evolving of ST_i is regular, which leads to a good performance on service regularity. The standard deviation equals 0 when $k = 20$, which means all the links in MSMW are scheduled in cycles. This is achieved by the underlying evolving of ST_is and w_is.

E. Sensitivity for the Arriving Rate Vector

The performance of MSMW on service frequency does not depend on the arriving rate vectors. It always has a good performance on the service frequency no matter what the arriving rate vector is. There are 16 links in our simulations, with $\delta_i = N + 5$ for link i. We use an *asymptotically decreasing* rate vector with $E(A_i) = \frac{1}{N*(i+k)}$ ($k = 1, 2, \cdots, 20$) and also a *two tails* rate vector with $E(A_i) = \frac{1}{N*i}$ for the first 8 links, and $E(A_i) = \frac{1}{N*(i+k)}$ ($k = 1, 2, \cdots, 20$) for the last 8 links. The ratio of the links following the service frequency constraints is shown in Fig.5.

As can be seen, MSMW performs well for both types of rate vectors, while the chosen parameters of RTO can only fit the *asymptotically decreasing* rate vector. We must update the parameters of RTO to deal with the *two tails* one. Therefore, our scheduling policy is much more applicable in the practical case where the arriving rate of each link is often periodically changing and unpredictable.

F. Performance in General Networks

In this part, we evaluates the performance of our scheduling policy in general networks. We assume 40 nodes are randomly distributed in a 100m×100m area. The transmission and interference radius of a node is set to be 20m. The channel condition of each link is set to be 0.5, The transmission rate for each link is 2. The average arriving rate is set to be $\frac{1}{8}$ and max arriving rate is $\frac{1}{4}$. We use the maximal weight scheduling policy [9] as our comparison. In each slot, the maximal weight scheduling policy iteratively selects links with max queue length and minimum degree in the conflict graph, and deletes all links conflicting with the chosen one.

Fig.6 shows the results with different number of links. The service constraints is set to be 10, and we change the number
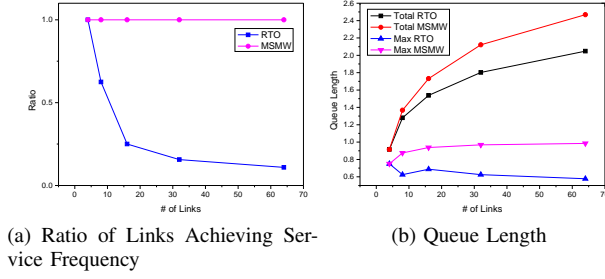
(a) Ratio of Links Achieving Service Frequency

(b) Queue Length

Fig. 1: System performance with different numbers of links

(a) Ratio of Links Achieving Service Frequency

(b) Queue Length
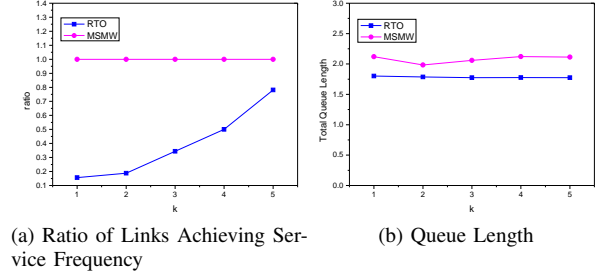
Fig. 2: System performance with various service frequency constraints

(a) Ratio of Links Achieving Service Frequency

(b) Queue Length
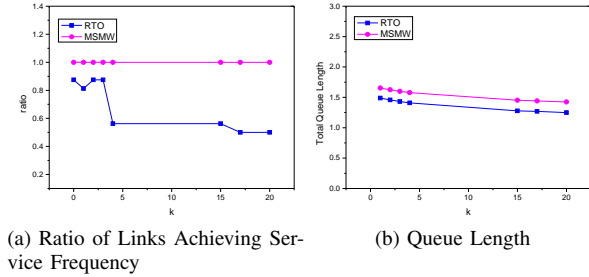
Fig. 3: System performance with various arriving rates

(a) Max Standard Deviation

(b) Average Standard Deviation
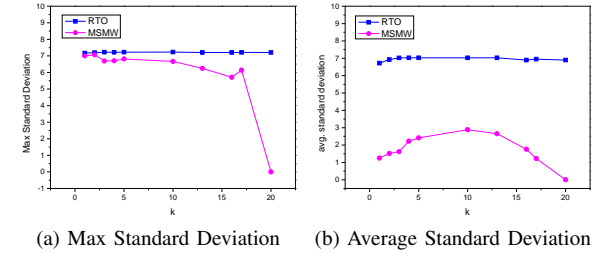
Fig. 4: Performance on service regularity

Fig. 5: Robust of MSMW.

(a) Ratio of Links Achieving Service Frequency

(b) Queue Length

Fig. 6: Performance with different number of links

(a) Ratio of Links Achieving Service Frequency
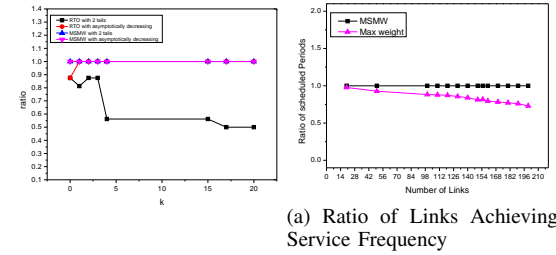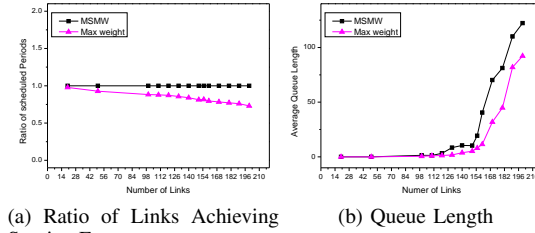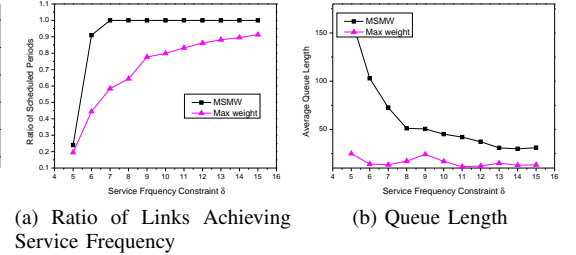
(b) Queue Length

Fig. 7: Performance with different service frequency constraints

of links from 20 to 200. As we see, MSMW has a good performance on both queue length and service frequency. All the links follow the constraint on service frequency, while the maximal weight scheduling policy can only follow the constraints on 70 percents of its periods. The average queue length in MSMW is less than 4 times of the that in maximal weight scheduling policy in worst case, *i.e.*, 40/11.5 when there are 160 links.

In Fig.7, we evaluate MSMW in general networks with different service frequency constraints. We set the number of links to be 160, and change the constraints from 5 to 15. Based on Fig.7, the average queue length becomes smaller as δ increases. The reason is that more slots can be released for scheduling links according to the queue length. When $\delta \geq 12$, the queue length of MSMW is already small and close to that of maximal weight scheduling policy. Meanwhile, the performance on service frequency quickly achieve the best in MSMW, while the ratio for our comparison method is only 90 percents even if $\delta = 15$.

As a whole, MSMW is efficient no matter what the users' behaviors are. It always guarantees service frequency while only increases the queue length by a little bit.

VII. CONCLUSION

This paper investigates the problem of scheduling links in wireless networks with service frequency constraints. To the best of knowledge, this is the first work dealing with traffics in wireless networks with stringent service frequency requirements. We improve the typical network models to handle the constraints. A new scheduling policy called MSMW is proposed, and it is proved to be throughput-optimal in collocated networks. Furthermore, this paper discusses the decreasing of capacity region and the changing of queue length caused by the existence of service frequency constraints. The simulation results show that MSMW performs well on the aspects of service frequency and total queue length. As future works, we are interested in the relationship between the system

performance and the correlation of arriving rates and service frequency constraints.

REFERENCES

[1] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Computer networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[2] M. Chen, S. Gonzalez, and V. Leung, "Applications and design issues for mobile agents in wireless sensor networks," *Wireless Communications, IEEE*, vol. 14, no. 6, pp. 20–26, 2007.

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

[4] 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 fourteenth ACM international symposium on Mobile ad hoc networking and computing*, pp. 99–108, ACM, 2013.

[5] R. Li, A. Eryilmaz, and B. Li, "Throughput-optimal wireless scheduling with regulated inter-service times," in *INFOCOM, 2013 Proceedings IEEE*, pp. 2616–2624, 2013.

[6] B. Li, R. Li, and A. Eryilmaz, "Heavy-traffic-optimal scheduling with regular service guarantees in wireless networks," in *MobiHoc, 2013 Proceedings ACM*, pp. 79–88, ACM, 2013.

[7] Y. Liu, S. Dey, D. Gillies, F. Ulupinar, and M. Luby, "User experience modeling for dash video," in *Packet Video Workshop (PV), 2013 20th International*, pp. 1–8, IEEE, 2013.

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

[9] G. R. Gupta and N. B. Shroff, "Delay analysis for wireless networks with single hop traffic and general interference constraints," *IEEE/ACM Transactions on Networking (TON)*, vol. 18, no. 2, pp. 393–405, 2010.

[10] C. Boyaci and Y. Xia, "Optimal delay bound for maximum weight scheduling policy in wireless networks," in *INFOCOM, 2014 Proceedings IEEE*, pp. 565–573, IEEE, 2014.

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

[12] 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," *Networking, IEEE/ACM Transactions on*, vol. 20, no. 3, pp. 825–836, 2012.

[13] X. Lin and N. B. Shroff, "The impact of imperfect scheduling on cross-layer congestion control in wireless networks," *Networking, IEEE/ACM Transactions on*, vol. 14, no. 2, pp. 302–315, 2006.

[14] 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, pp. 313–322, 2003.

[15] P.-J. Wan, O. Frieder, X. Jia, F. Yao, X. Xu, and S. Tang, "Wireless link scheduling under physical interference model," in *INFOCOM, 2011 Proceedings IEEE*, pp. 838–845, IEEE, 2011.

[16] 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 fourteenth ACM international symposium on Mobile ad hoc networking and computing*, pp. 89–98, ACM, 2013.

[17] I. Hou, P. Kumar, *et al.*, "Utility-optimal scheduling in time-varying wireless networks with delay constraints," in *Proceedings of the eleventh ACM international symposium on Mobile ad hoc networking and computing*, pp. 31–40, ACM, 2010.

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

[19] I.-H. Hou and P. Kumar, "Scheduling heterogeneous real-time traffic over fading wireless channels," in *INFOCOM, 2010 Proceedings IEEE*, pp. 1–9, IEEE, 2010.

[20] E. Leonardi, M. Mellia, F. Neri, and M. Ajmone Marsan, "On the stability of input-queued switches with speed-up," *Networking, IEEE/ACM Transactions on*, vol. 9, no. 1, pp. 104–118, 2001.

[21] "Technique report: Scheduling flows with multiple service frequency constraints," https://www.dropbox.com/s/0ol1bsz8yir7ngc/TechniqueReportMSMW.pdf?dl=0.

Xu Zheng received his B.S. and M.S. degree from School of Computer Science and Technology at Harbin Institute of Technology, where he is pursuing his PhD degree. Mr. Zheng is currently a visiting PhD student in the Department of Computer Science at Georgia State University. Mr. Zheng's research areas focus on wireless network.

Zhipeng Cai received his PhD and M.S. degree in Department of Computing Science at University of Alberta, and B.S. degree from Department of Computer Science and Engineering at Beijing Institute of Technology. Dr. Cai is currently an Assistant Professor in the Department of Computer Science at Georgia State University. Prior to joining GSU, Dr. Cai was a research faculty in the School of Electrical and Computer Engineering at Georgia Institute of Technology. Dr. Cai's research areas focus on Networking and Big data. Dr. Cai is the recipient of an NSF CAREER Award.

Jianzhong Li is a professor in the School of Computer Science and Technology at Harbin Institute of Technology. In the past, he worked as a visiting scholar at the University of California at Berkeley, USA, as a staff scientist in the Information Research Group at the Lawrence Berkeley National Laboratory, USA, and as a visiting professor at the University of Minnesota. His research interests include data management systems, sensor networks and data intensive computing.

Hong Gao received the B.S. degree in computer science from Heilongjiang University, the M.S. degree in computer science from Harbin Engineering University, China, and the Ph.D. degree in computer science from Harbin Institute of Technology. She is currently a professor in the School of Computer Science and Technology at Harbin Institute of Technology. Her research interests include graph data management, sensor network, and massive data management.