Distributed throughput optimal scheduling for wireless networks

DISTRIBUTED THROUGHPUT OPTIMAL SCHEDULING FOR WIRELESS NETWORKS

A Thesis by

Shuang Xia

Bachelor of Engineering, Wuhan University of Science and Technology, 2010

Submitted to the Department of Electrical Engineering and Computer Science
and the faculty of the Graduate School of
Wichita State University
in partial fulfillment of
the requirements for the degree of
Master of Science
in Electrical Engineering

December 2014

DISTRIBUTED THROUGHPUT OPTIMAL SCHEDULING FOR WIRELESS NETWORKS

The following faculty members have examined the final copy of this thesis for form and content, and recommend that it be accepted in partial fulfillment of the requirement for the degree of Master of Science with a major in Electrical Engineering.

Pu Wang, Committee Chair

Vinod Namboodiri, Committee Member

Atri Dutta, Committee Member

ACKNOWLEDGEMENTS

I would like to thank my advisor, Dr. Pu.Wang, who made it possible for me to complete this thesis. Her support, knowledge, and patience have guided me from the very beginning to the end. I would also like to thank Dr. Vinod Namboodiri and Dr. Atri Dutta for their kind help and for serving on my thesis committee.

ABSTRACT

Recent advancement in distributed scheduling algorithms mainly focuses on designing CSMA-type protocols to achieve maximum network throughput in a fully distributive manner. However, it is inherently difficult for distributed scheduling algorithms to promise hard deadlines and a good performance in the presence of heavy-tailed traffic. To encounter this, there are two distributed throughput optimal scheduling to be proposed, which is timely-throughput optimal scheduling and throughput optimal scheduling with heavy-tailed traffic. The timely-throughput optimal scheduling distributed determines the optimal transmission times for network users so that the largest set of traffic rates of network users can be supported, while ensuring timely data delivery within hard deadlines. Then, the distributed throughput optimal scheduling with heavy-tailed traffic is proposed, which makes the scheduling decision based on the queue lengths raised to the α-th power. It is demonstrated that DMWS-α is throughput optimal with respect to moment stability in the sense that if the traffic arrivals rates are within the network stability region, all network users with light-tailed traffic arrivals always have bounded queueing delay with finite mean and variance.

TABLE OF CONTENTS

Chapter **Page**

TABLE OF CONTENTS (continued)

Chapter **Page**

LIST OF FIGURES

Figure **Page**

CHAPTER 1

INTRODUCTION

1.1 Background

In recent years, Because wireless networks have been deployed for a number of emerging applications, wireless traffic has grown tremendously in the past few years, while the spectrum resources are limited. To address such challenge, throughput optimal scheduling is desirable, which determines the optimal transmission times for network users so that the largest set of traffic rates can be supported, while maintaining desired network stability. As an important class of throughput optimal scheduling algorithms, the maximum-weight scheduling and many of its variants [11] are great interests because they can achieve throughput optimality only requiring the knowledge of users' current queue lengths and instantaneous channel rates.

Recently, distributed throughput optimal scheduling algorithms have drawn lots of attention because of their inherent throughput optimality, operating simplicity, and performance scalability. The main focus is on designing CSMA-type protocols to approximate the centralized maximum weight scheduling in a distributive fashion [4, 7]. For example, by exploiting glauber dynamics from statistical physics along the product form of transmission states of CSMA Markov chain, QCSMA [7] is shown to be throughput optimal via fully distributed operations in the sense that each user makes transmission decisions only based on its local information, including its current queue length, its instantaneous data rate, and its historical carrier sensing results.

Despite their great advantages, distributed throughput optimal algorithms are not able to support the applications that require critical data to be delivered within hard deadlines. Such critical data can be the speed information exchanged among nearby cars for vehicle collision avoidance [1] or can be the sensing data sent from neighboring robots for collaborative source seeking [17]. In both cases, the critical data will become useless if

they are not disseminated within certain delay deadlines. On the other hand, distributed throughput optimal algorithms are also developed and studied under the light-tailed traffic conditions, which is largely departing from the increased proliferation of video streaming, cloud computing, and Internet file transfer over mobile devices, which inevitably cause the rise of heavy tailed traffic over wireless networks, such as WLAN [3], mobile ad-hoc networks [12], and cellular networks [9]. Such heavy-tailed traffic can be either caused by the inherent heavy-tailed distribution in the traffic source such as the file size on the Internet servers or cellular base stations, the web access pattern, and the frame length of VBR (variable bit rate) video streams [8], or caused by the network protocols themselves such as retransmissions and random access schemes [10]. Different from the conventional light-tailed traffic (i.e., Markovian or Poisson traffic), heavy-tailed traffic exhibits high burstiness or dependence over a long range of time scale. Such highly bursty nature can induce significant performance degradation, including the considerably reduced network throughput [14], stability [16], and connectivity [15].

1.2 Research Objectives and Solutions

To address the challenges above, there are two new distributed throughput optimal algorithms to be proposed. The first one is Timely-Throughput Optimal Scheduling, which can guarantee critical data to be delivered within hard deadlines; the second one is Throughput Optimal Scheduling with Heavy-Tailed Traffic, which has a good performance in the presence of heavy-tailed traffic.

1.2.1 Timely-Throughput Optimal Scheduling

To address the issue of hard deadlines, we design a distributed timely-throughput optimal scheduling algorithm, which distributively determines the optimal transmission times for network users so that the largest set of traffic rates of network users can be supported, while ensuring timely data delivery within deadlines. More specifically, we first derive the maximum network capacity region that timely-throughput optimal scheduling algorithms

can achieve. Then, we develop a fully distributed scheduling algorithm by exploiting the principles of Glauber dynamics [6] from statistical physics in such a way that every user can make optimal transmission decision locally based on its incoming traffic rate, virtual debts, and previous channel sensing results. Finally, we prove that the proposed scheduling algorithm is timely-throughput optimal in the sense that it can guarantee the timely data delivery as long as the incoming traffic rates are within the maximum network capacity region.

1.2.2 Throughput Optimal Scheduling with Heavy-Tailed Traffic

In this thesis, we develop and study distributed throughput optimal scheduling algorithms, which can achieve *moment stability* in the entire network stability region. Moment stability is a new stability criterion first proposed in [13, 16], aiming to characterize the QoS performance of wireless networks in the presence of heavy-tailed environment. Moment stability requires that all the network users with light-tailed traffic arrivals always have bounded queueing delay with finite mean and variance. Compared to strong stability, moment stability not only requires the finiteness of lower order moments, such as mean, but also demands the boundedness of higher order moments, such as variance, provided that such moments exist. What is more important, achieving moment stability can prevent heavy-tailed traffic (e.g., video streaming and file transferring) from significantly degrading the queueing performance of light-tailed traffic (e.g., email deliveries, audio/voice traffic, and temperature and humidity readings). Based on the definition of moment stability, the *network stability region* is defined as the closure of the set of all arrival rate vectors for which the queues of all network users can be stabilized by a feasible scheduling policy. Moreover, a scheduling policy is *throughput optimal* if it stabilizes the system for any arrival rates in the stability region.

Furthermore, we first study the stability performance of a distributed maximum weight scheduling algorithm (DMWS), which is proven to be throughput optimal with light tailed traffic arrivals, while exhibiting the superior delay performance even under the chal-

lenging fast channel fading environment. More specifically, we show that heavy tailed traffic can significently degrade the stability of DMWS by inducing unbounded queueing delay. More specifically, it is shown that when a queue with light-tailed arrivals has the arrival rate λ below a threshold λ^*, its queue length is of finite mean. Otherwise, if λ is larger than a threshold λ', its queue length is of infinite mean. In other words, in the heavy tailed environment, the DMWS can only guarantee moment stability under a fraction of the network stability region. To encounter this, we propose the distributed maximum weight-α scheduling (DMWS-α) algorithm, which makes the scheduling decision based on the queue lengths raised to the α-th power. Accordingly, we show that DMWS-α is throughput optimal in the sense that it can guarantee moment stability as long as the traffic arrival rates are within the network stability region.

1.3 Organizations of Thesis

The remainder of this thesis is organized as follows: Chapter 2 presents the timely-throughput optimal scheduling for wireless networks. Then, we present and prove some useful lemmas that are applied throughout the thesis. Chapter 3 proposes the throughput optimal scheduling in the presence of heavy-tailed traffic. Then, we prove its throughput optimality and show simulation results. Chapter 4 concludes the thesis.

CHAPTER 2

Timely-Throughput Optimal Scheduling

2.1 System Model

Consider a fully-connected network with N wireless users sharing a single wireless channel. Time is slotted and during each time slot, only one packet is transmitted, slots are numbered by $t \in 1, 2, 3, ...$ furthermore, time slots are grouped into intervals $[kT, (k+1)T]$, where each interval consists of T consecutive time slots. At the beginning of each interval, that is, at times $1, T+1, 2T+1, ...,$, each user i generates packets following Bernoulli process $A_i(t)$ with average traffic rate of $E[A_i(t)] = \lambda_i$, which means that at the beginning of each interval, user i generates one packet with probability λ_i. The packets from all users need to be delivered within a delay bound of T time slots if they are to be useful. In other words, packets that are generated at the beginning of an interval are only useful if they are delivered no later than the end of the interval. If a packet is not delivered within the time interval, the packet is dropped. By this way, it is guaranteed that the delay of every delivered packet is at most T time slots. Since wireless channel is inherently lossy, each user i has a packet error rate $1 - m_i$, which means with probability m_i, a packet is delivered without errors. m_i can be different from user to user because of the channel diversity.

Based on the above network model, we formally define the time-throughput optimal scheduling problem as follows.

Definition 1 *(**Timely-delivery Ratio**) To enforce the QoS, each user can demand a timely-delivery ratio a_i, which means that at least a percentage a_i of packets from user i have to be delivered without errors before the deadline.*

Definition 2 *(**Viable Schedule**) A scheduling algorithm is viable if it can yield a transmission order for the network users in such a way that all network users can meet their respective timely-delivery ratio.*

Definition 3 *(**Maximum Achievable Capacity Region**) The maximum network capacity region is the closure of the set of all arrival rate vectors under which there exists an viable scheduling algorithm.*

Definition 4 *(**Timely-throughput Optimal Scheduling**) A scheduling policy is timely-throughput optimal if it is viable for any arrival rates within the maximum network capacity region.*

2.2 Maximum Network Capacity Region

In this section, we derive the maximum network capacity region for time-throughput optimal scheduling algorithms by constructing the following queueing system with virtual debt queues.

Assign a virtual debt queue q_i to each user i. Let $D_i(k)$ denote the length of the queue q_i by the end of time interval k. Let $R_i(k) = a_i$ demote the debt arrival process of queue q_i, where a_i is the required timely-delivery ratio of user i. Specifically, $R_i(k) = a_i$ means that at the beginning of each interval k, a new debt of size a_i arrives at queue q_i. Define $h_{ij}(k) \in \{0, 1\}, \forall i \leq N, j \leq T$ as the scheduling decision for queue i, where $h_{ij} = 1$ means if the j-th time slot of the time interval k is allocated to queue i. Let $M_i(k) \in \{0, 1\}$, where $M_i(k) = 1$ with probability $1 - m_i$ and m_i is the packet error rate of user i. Then, the queueing system with virtual debt queues can be represented by

$$D_i(k+1) = D_i(k) - \sum_{j=1}^{T} h_{ij}(k) A_i(k) M_i(k) + R_i(k) \tag{2.1}$$

subject to

$$h_{ij}(k) \in \{0, 1\}, \quad \forall i, j \tag{2.2}$$

$$0 \leq \sum_{j=1}^{T} h_{ij}(k) \leq 1 \quad \forall i \tag{2.3}$$

$$0 \leq \sum_{i=1}^{N} h_{ij}(k) \leq 1 \quad \forall j \tag{2.4}$$

where the second constraint implies that each user can only be allocated with one time-slot, while the third constraint means that each time slot can only be assigned to one user. By defining $U_i(k)$ as the total amount of debts which are released from queue i at time interval k under a certain scheduling policy, the queueing dynamics in (2.1) can be rewritten by

$$D_i(k+1) = D_i(k) - U_i(k) + R_i(k). \tag{2.5}$$

Let D_i denote the steady-state queue length of $D_i(k)$. Based on the queueing dynamics of (2.5), we have the following lemma.

Lemma 1 *A scheduling algorithm is viable if and only if it can stabilize the virtual debt queueing system so that* $E[D_i] \leq \infty, \forall 1 \leq i \leq N$.

Proof 1 *Recall that* $A_i(k)$ *is the number of packet that user* i *generates at the beginning of time interval* T*, and need to be delivered before the deadline* T*, and* $1-m_i$ *is the packet error rate of user* i*. It is easy to see that* $U_i(k) = \sum_{j=1}^{T} h_{ij}(k)A_i(k)m_i$ *is actually the number of packets transmitted by user* i *before the deadline* T *without errors. Therefore, on one hand, it follows by definition 2 that if a scheduling algorithm is viable, it has to guarantee that* $E[U_i(k)] \geq E[R_i(k)] = \alpha_i, \forall 1 \leq i \leq N$*, which means the virtual debt queueing system is stable, i.e.,* $E[D_i] \leq \infty, \forall 1 \leq i \leq N$*. On the other hand, if the virtual debt queueing system is stable, this necessarily implies that* $E[U_i(k)] = E[R_i(k)] = \alpha_i, \forall 1 \leq i \leq N$.

The next theorem provides the necessary condition for the existence of a viable scheduling. Such necessary condition also establishes the maximum achievable capacity region, characterized by (2.6), for timely-throughput optimal scheduling algorithms.

Theorem 1 *If a scheduling policy is viable, then we have*

$$\sum_{i \in D} \frac{a_i}{m_i} \leq T - \sum_{n=1}^{T} P(X(k) < n), \quad \forall D \subset \{1, ..., N\} \tag{2.6}$$

where $X(k)$ *is the number of users which generate packets at the beginning of the time interval* k*,* $X(k)$ *follows Poisson binomial distribution, denoted by* $X(k) \sim PB(\lambda, D)$ *and*

$\lambda = \{\lambda_i\}_{i \in D}$, *i.e.*,

$$P(X(k) < n) = \sum_{l=0}^{n} \sum_{A \in F_l} \left(\prod_{j \in A} \lambda_j \prod_{j \in A^c} (1 - \lambda_j) \right), \tag{2.7}$$

where A^c *is the complementary set of* A*, and* F_l *is the collection of all subsets of* l *integers that are selectable from set* D*.*

Proof 2 *Suppose the system is stable under a viable scheduling policy. Let* $H_i(k) = \sum_{j=1}^{T} h_{ij}(k) A_i(k)$*. The departure process from queue* i *is* $U_i(k) = \{H_i(k) M_i(k)\}_{k=1}^{\infty}$ *with the departure rate of* $E[U_i(k)]$*, where*

$$E[U_i(k)] = m_i E[H_i(k)] \tag{2.8}$$

Since the system is stable, in each queue, the debt departure rate should be equal to the debt arrival rate, i.e.,

$$m_i E[H_i(k)] = E[R_i(k)] = a_i \tag{2.9}$$

by which we have, for any set D *of users,*

$$\sum_{i \in D} \frac{a_i}{m_i} = \sum_{i \in D} E[H_i(k)]. \tag{2.10}$$

Define $X(k)$ *as the number of users in set* D*, which have packets to send at the beginning of time interval* T*, i.e.,* $X(k) = |\{i | A_i(k) \neq 0, i \leq N\}|$*. Since* $P(A_i(k) = 1) = \lambda_i$*, this implies that* $X(k)$ *follows Poisson binomial distribution, i.e.,*

$$X(k) \sim PB(\lambda, D) \tag{2.11}$$

where $\lambda = \{\lambda_i\}_{i \in D}$*. Now, equation* (2.10) *can be rewritten as*

$$\sum_{i \in D} \frac{a_i}{m_i} = E\left[E\left[\sum_{i \in D} H_i(k) | X(k), D_i(k-1), i \in D \right] \right] \tag{2.12}$$

The event $e = \{X(k), D_i(k-1), i \in D\}$ *can be partitioned into three disjoint sets*

$$\begin{aligned} e_1 &= \{X(k) = 0\} \\ e_2 &= \{X(k) = 0\}^c \wedge \{D_i(k-1) = 0, i \in D\} \\ e_3 &= \{X(k) = 0\}^c \wedge \{D_i(k-1) = 0, i \in D\}^c \end{aligned} \tag{2.13}$$

It is easy to verify that

$$E\left[\sum_{i\in D} H_i(k)|e_i\right] = 0, \quad i = 1, 2 \tag{2.14}$$

As to event e_3*, we can further divide it into two disjoint sets*

$$\begin{aligned} e_3^{(1)} &= \{X(k) < T\} \wedge \{D_i(k-1) = 0, i \in D\}^c \\ e_3^{(2)} &= \{X(k) \geq T\} \wedge \{D_i(k-1) = 0, i \in D\}^c \end{aligned} \tag{2.15}$$

which, in conjunction with (2.14) *and* (2.12)*, implies*

$$\sum_{i\in D} \frac{a_i}{m_i} = E\left[\sum_{i\in D} H_i(k) 1_{e_3^{(1)}}\right] + E\left[\sum_{i\in D} H_i(k) 1_{e_3^{(2)}}\right] \tag{2.16}$$

Define $u_j = \{X(k) = j\} \wedge \{D_i(k-1) = 0, i \in D\}^c$ *as the event that there are* j *users generating packets and at least one of the debt queues is not empty. For the first term on the right side of* (2.16)*, we have*

$$\begin{aligned} E\left[\sum_{i\in D} H_i(k) 1_{e_3^{(1)}}\right] &= \sum_{j=1}^{T-1} \sum_{i\in D} E[H_i(k)|u_j] P(u_j) \\ &= \sum_{j=1}^{T-1} jP\left(\{X(k) = j\} \right. \\ &\qquad \left. \wedge \{D_i(k-1) = 0, i \in D\}^c\right) \\ &\leq \sum_{j=1}^{T-1} jP(X(k) = j) \end{aligned} \tag{2.17}$$

The second equality is due to the fact that $\sum_{i\in D} H_i(k) = X(k)$ *if* $X(k) \leq T - 1$*. For the second term on the right side of* (2.16)*, we have*

$$\begin{aligned} E\left[\sum_{i\in D} H_i(k) 1_{T_3^{(1)}}\right] &= \sum_{j=T}^{|D|} \sum_{i\in D} E[H_i(k)|u_j] P(u_j) \\ &\leq T \sum_{j=T}^{|D|} P\left(\{X(k) = j\} \right. \\ &\qquad \left. \wedge \{D_i(t-1) = 0, i \in D\}^c\right) \\ &\leq TP(X(k) \geq T) \end{aligned} \tag{2.18}$$

The second inequality holds because the number of channels allocated can not exceed the maximum number of queues, which implies $\sum_{i\in D} H_i(k) \leq T$ *if* $X(k) \geq T$. *Combining* (2.16), (2.17), *and* (2.18), *we have*

$$\begin{aligned}\sum_{j\in D}\frac{a_j}{m_j} &\leq \sum_{j=1}^{T-1} jP(X(k)=j) + TP(X(k)\geq T)\\ &= \sum_{j=1}^{T-1}(\sum_{i=j}^{T-1} P(X(k)=i) + TP(X(k)\geq T)\\ &= T - \sum_{j=1}^{T} P(X(k)<j) \qquad (2.19)\end{aligned}$$

which, combining with (2.11), *completes the proof.*

2.3 Distributed Timely Throughput Optimal Scheduling

In this section, we first introduce our proposed scheduling algorithm and then prove its timely-throughput optimality.

Consider a time interval of T time lots. We divide each time slot into a control period and a data period, where the control period consists of W mini control slots. The proposed algorithm consists of two procedures: contention procedure during control period followed by data transmission procedure during data slot. As for contention procedure (line 1 - 3 in Algorithm 1), at the beginning of each time interval, each node i uniformly selects a time slot from T slots with probability $\frac{1}{T}$. Assume the node i selects j-th time lot. Then, it has to wait until j-th time slot to perform channel contention. Specifically, during the control period of time slot j, the node i randomly selects a control mini-slot with probability $\frac{1}{W}$. If node i receives the *CONTENTION* messages sent by other nodes before its selected mini-slot, it keeps silent during the following data period. Otherwise, node i sends out its *CONTENTION* message during its selected mini-slot and begins to perform the data transmission procedure. During this procedure (line 4 - 14 in Algorithm 1), node i first determines if the data period of its selected time slot j is used by another node for transmission in the previous time interval based on previous sensing results. If so, node i puts such node into its conflict set

$C(i)$ and node i does not transmit during the data period. Otherwise, node i transmits with a probability p_i and chooses to be inactive with probability $1 - p_i$ during the data period, where $p_i = \frac{e^{Di(k)Ai(k)}}{e^{Di(k)Ai(k)+1}}$

Algorithm 1 Basic Scheduling Algorithm (in Time Interval k)

1: Node i randomly select a time slot j in time interval T with probability $\frac{1}{T}$.
2: During its selected time slot j, node i randomly select a control mini-slot with probability $\frac{1}{W}$ and send a CONTENTION message during its selected mini-slot.
3: $i \in m(k)$ if node i did not hear CONTENTION messages from other nodes before its own control mini slot begins. Otherwise, $i \notin m(k)$
4: let C_i be a set containing the node $m \neq i$ that transmits in the time slot j of the previous time interval $k - 1$
5: **if** $i \in m(k)$ **then**
6: **if** $|C(i)| = 0$ **then**
7: (a) $x_i(k) = 1$ with probability $p_i = \frac{e^{D_i(k)A_i(k)}}{e^{D_i(k)A_i(k)+1}}$
8: (b) $x_i(k) = 0$ with probability $\overline{p}_i = 1 - p_i$
9: **else**
10: (c) $x_i(k) = 0$.
11: **end if**
12: **else**
13: (d) $x_i(k) = x_i(k - 1)$.
14: **end if**
15: In the data slot, use $x(k)$ as the transmission schedule.

Note that this deadline scheduling algorithm is the extension of the so-called Glauber dynamics from statistical physics, where multiple nodes can update their states in a time interval. In the following parts, we will prove our scheduling is a discrete-time Markov chain and introduce its stationary distribution.

Theorem 2 *The proposed distributed scheduling algorithms yields a transmission schedule vector $x(k) = \{x_i(k)\}_{i \leq N}$, which is a discrete-time Markov chain (DTMC) with stationary distribution as follow:*

$$\pi(x) = \frac{1}{Z} e^{\sum_{i \in x} D_i(k) A_i(k)} \tag{2.20}$$

$$Z = \sum_{x \in \mathcal{S}} e^{\sum_{i \in x} D_i(k) A_i(k)} \tag{2.21}$$

where $\mathcal{S}$ be the set consisting of all the possible transmission schedule.

Proof 3 *To obtain stationary distribution, we need first to prove our transmission schedules vector is a discrete-time Markov chain(DTMC). It is easy to see that all nodes in $m(k)$ will use different time slots for the following data transmission procedure, and the nodes in $m(k)$ wont interfere each other.*

Now we suppose $\forall i \in V$ and $x_i(k) = 1$. If $i \notin m(k)$, we have $x_i(k) = x_i(k-1) = 1$ based on step (d). On the other hand, If $i \in m(k)$ and $l \in C(i)$, we have $x_l(k-1) = 0$ based on step (c). In addition, since $C(i) \cap m(k) = \phi$, we have $l \notin m(k)$ and $x_l(k) = x_l(k-1) = 0$ based on step (d). Therefore, the $x(k)$ always depends on $x(k-1)$, so, the $x(k)$ is a discrete-time Markov chain (DTMC).

In addition, to prove our transmission schedules vector has a stationary distribution is sufficient to show that our DTMC is irreducible and aperiodic. By employing the similar techniques in [7], it is easy to obtain the transition probability as follows:

A state $x \in \mathcal{S}$ can make a transition to a state $y \in \mathcal{S}$ if and only if $x \cup y \in \mathcal{S}$ and there exists a set of decision schedule $m \in \mathcal{S}$ such that

$$x \triangle y = (x \setminus y) \cup (y \setminus x) \subseteq m \tag{2.22}$$

and in this case the transition probability from x to y is:

$$\begin{aligned} P(x,y) = \sum_{m \in \mathcal{S}: x \triangle y \subseteq m} \alpha(m) \Big(\prod_{i \in x \setminus y} \overline{p}_i \Big) \Big(\prod_{j \in y \setminus x} p_j \Big) \\ \Big(\prod_{k \in m \cap (x \cap y)} p_k \Big) \Big(\prod_{l \in m \setminus (x \cup y) \setminus C(x \cup y)} \overline{p}_j \Big) \end{aligned} \tag{2.23}$$

If $\cup_{m \in \mathcal{S}_0} m \neq V$, suppose $l \notin \cup_{m \in \mathcal{S}_0} m$, then from state 0 the DTMC will never reach another state or a feasible schedule including l. So, we can not say that the Markov chain is irreducible.

On the other hand, for any schedule m, m will be selected in the control slot with positive probability if $W > 2$. It is easy to check by Algorithm 1 that if $W > 2$, $\alpha(m(k)) \geq \frac{1}{T} \cdot \frac{1}{W} \cdot T = \frac{1}{W}$. As a consequence, using above transition function, we are able to show that the state 0 can reach any other state $x \in \mathcal{S}$ with a positive probability in a finite number of

steps, denoted by $P^n(0,x) > 0$ for a constant n. In addition, from the transition function, we also can know $P^n(x,x) > 0$, n is a constant, so, we can say the Markov chain is aperiodic. Therefore, the DTMC is irreducible and aperiodic.

Finally, if state x can make a transition to state y, through the distribution in (2.28)*, we have:*

$$\pi(x)P(x,y) = \pi(y)P(y,x), \quad x,y \in \mathcal{S} \tag{2.24}$$

which means that the DTMC is reversible [5] and the product-form stationary distribution [7] holds as follows

$$\pi(x) = \frac{1}{Z}\prod_{i\in x}\frac{p_i}{\overline{p}_i} \tag{2.25}$$

$$Z = \sum_{x\in\mathcal{S}}\prod_{i\in x}\frac{p_i}{\overline{p}_i} \tag{2.26}$$

Then, according to our scheduling algorithm, we have

$$p_i = \frac{e^{D_i(k)A_i(k)}}{e^{D_i(k)A_i(k)}+1} \tag{2.27}$$

which, combining with (2.25)*, leads to*

$$\pi(x) = \frac{1}{Z}e^{\sum_{i\in x} D_i(k)A_i(k)} \tag{2.28}$$

$$Z = \sum_{x\in\mathcal{S}} e^{\sum_{i\in x} D_i(k)A_i(k)} \tag{2.29}$$

Based theorem 1 and 2, next we prove the timely-throughput optimality of our proposed distributed scheduling algorithm.

Theorem 3 *The proposed distributed scheduling protocol in Algorithm* (1) *is timely-throughput optimal. That is, Algorithm* (1) *is necessarily viable if the incoming date rate $\{\lambda_i\}_{i\leq N}$ is within the maximum achievable capacity region (defined by Theorem 1) by satisfying*

$$\sum_{i\in D}\frac{a_i}{m_i} \leq T - \sum_{n=1}^{T} P(X(k) < n), \quad \forall D \subset \{1,...,N\} \tag{2.30}$$

where $X(k) \sim PB(\lambda, D)$ *and* $\lambda = \{\lambda_i\}_{i\in D}$*, i.e.,*

$$P(X(k) < n) = \sum_{l=0}^{n} \sum_{A\in F_l} \left(\prod_{j\in A} \lambda_j \prod_{j\in A^c} (1-\lambda_j) \right),$$

where A^c *is the complementary set of* A*, and* F_l *is the collection of all subsets of* l *integers that are selectable from set* D*.*

Proof 4 *It directly follows from Theorem 2 that at each time interval, with the highest probability, Algorithm* (1) *selects the maximum weight schedule* x^* *in such as way that*

$$\sum_{i\in x^*} D_i(k)A_i(k) = \max_{x\in\mathcal{S}} \sum_{i\in x} D_i(k)A_i(k) \tag{2.31}$$

where S *is the set of all the possible transmission schedule. It is easy to show that as the debt queue* $D_i(k)$ *increases, the maximum weight schedule* x^* *will be selected with probability approaching one. Thus, proving the proposed distributed scheduling algorithm is timely-throughput optimal is equivalent to proving the centralized algorithm in* (2.31) *is timely-throughput optimal. This fact, combining with lemma* (1) *and theorem* (1)*, implies that it is sufficient to prove if the incoming date rate* $\{\lambda_i\}_{i\leq N}$ *is within the maximum achievable capacity region (defined by Theorem 1, then the centralized algorithm in* (2.31) *guarantees that*

$$E[D_i] \leq \infty, \forall 1 \leq i \leq N.$$

Towards this, we adopt Lyapunov drift theory. Note that in this paper, we assume $T < N$ *because if* $T > N$*, it is easy to show that the simple TDMA scheduling can yield the timely-throughput optimality. Let* $\mathcal{D}(k) = (D_1(k), ..., D_N(k))$ *denote a vector process of debt lengths of N users, we define the Lyapunov function:*

$$L(\mathcal{D}(k)) = \sum_{i=1}^{N} L(D_i(k)) \tag{2.32}$$

where

$$L(D_i(k)) = m_i^{-1} D_i(k)^2 \tag{2.33}$$

$$L(\mathcal{D}(k)) = \sum_{i=1}^{N} m_i^{-1} D_i(k)^2 \tag{2.34}$$

To facilitate the proof, we rewrite the queueing dynamics in (2.1) *as follows*

$$D_i(k+1) = D_i(k) - H_i(k)M_i(k) + R_i(k) \tag{2.35}$$

where $H_i(k) = \sum_{j=1}^{T} h_{ij}(k)A_i(k)$. *As a consequence, we have*

$$\begin{aligned}
&E[L(\mathcal{D}(k+1))|\mathcal{D}(k) = D(k)] \\
&= E[\sum_{i=1}^{N} m_i^{-1} D_i(k+1)^2 | \mathcal{D}(k) = D(k)] \\
&\leq E[\sum_{i=1}^{N} m_i^{-1} (D_i(k) + R_i(k))^2 | \mathcal{D}(k) = D(k)] \\
&= L(\mathcal{D}(k)) + 2\sum_{i=1}^{N} m_i^{-1} a_i D_i(k) \\
&\qquad + \sum_{i=1}^{N} m_i^{-1} E[R_i^2(k)] < \infty
\end{aligned} \tag{2.36}$$

By some simple calculations we get

$$\begin{aligned}
&E[L(\mathcal{D}(k+1)) - L(\mathcal{D}(k))|\mathcal{D}(k)] \\
&= E[\sum_{i=1}^{N} m_i^{-1} (D_i(k+1)^2 - D_i(k)^2) | \mathcal{D}(k)] \\
&= E[\sum_{i=1}^{N} m_i^{-1} (D_i(k+1) - D_i(k)) \\
&\quad (D_i(k+1) - D_i(k) + 2D_i(k) | \mathcal{D}(k)] \\
&= E[\sum_{i=1}^{N} 2m_i^{-1} D_i(k)(D_i(k+1) - D_i(k)) | \mathcal{D}(k)] \\
&\quad + E[\sum_{i=1}^{N} m_i^{-1} (D_i(k+1) - D_i(k))^2 | \mathcal{D}(k)]
\end{aligned} \tag{2.37}$$

The second term of the sum in the right-hand side of (3.38) *can be upper bounded as*

$$\begin{aligned}
&E[\sum_{i=1}^{N} m_i^{-1}(D_i(k+1)-D_i(k))^2|\mathcal{D}(k)] \\
&\leq E[\sum_{i=1}^{N} m_i^{-1}(R_i(k))^2|\mathcal{D}(k)]+1 \\
&=\sum_{i=1}^{N} m_i^{-1}E[R_i(k)^2]+1. \qquad (2.38)
\end{aligned}$$

For the first term of the sum in the right-hand side of (3.38)*, we have*

$$\begin{aligned}
&E[\sum_{i=1}^{N} 2m_i^{-1}D_i(k)(D_i(k+1)-D_i(k))|\mathcal{D}(k)] \\
&=E[\sum_{i=1}^{N} 2m_i^{-1}D_i(k)R_i(k)|\mathcal{D}(k)] \\
&\quad -E[\sum_{i=1}^{N} 2m_i^{-1}D_i(k)H_i(k)M_i(k)|\mathcal{D}(k)] \qquad (2.39)
\end{aligned}$$

The first term of the sum in the right-hand side of (3.19) *is*

$$E[\sum_{i=1}^{N} 2m_i^{-1}D_i(k)R_i(k)|\mathcal{D}(k)]=2\sum_{i=1}^{N} D_i(k)\frac{a_i}{m_i} \qquad (2.40)$$

We now evaluate the expectation on the right side of (3.19)*. We arrange the debt queues in a decreasing order of the their products of debt queue length and packet arrival rate, i.e.,* $q_1, q_2, ..., q_N$ *with* $D_{q_i}(k)A_{q_i}(k) \geq D_{q_{i+1}}(k)A_{q_{i+1}}(k)$*, where ties are broken randomly.*

Now we can calculate the second term of the sum in the right-hand side of (3.19) *to be*

$$\begin{aligned}
&E[\sum_{i=1}^{N} 2m_{q_i}^{-1}D_{q_i}(k)H_{qi}(k)M_{q_i}(k)|\mathcal{D}(k)] \\
&=E[\sum_{i=1}^{N} 2D_{q_i}(k)H_{q_i}(k)|\mathcal{D}(k)] \\
&=\sum_{j=1}^{T} E\left[\sum_{i=1}^{N} 2D_{q_i}(k)H_{q_i}(k)|\mathcal{D}(k),X(k)=j\right]P(X(k)=j) \qquad (2.41)
\end{aligned}$$

where $X(k)$ is the number of users which generate packets at the beginning of interval k, i.e., $X(k) = |\{A_i(k)|A_i(k) \neq 0, \forall i \leq N\}|$

According to definition of the scheduling policy in (2.31)*, we have*

$$E\left[\sum_{i=1}^{N} 2D_{q_i}(k)H_{q_i}(k)|\mathcal{D}(k), X(k)=j\right] = \sum_{i=1}^{j} 2D_{q_i}(k)A_{q_i}(k)$$
$$= \sum_{i=1}^{j} 2D_{q_i}(k)$$

which combining with (2.41) *yields*

$$\begin{aligned}
&\sum_{j=1}^{T} E\left[\sum_{i=1}^{N} 2D_{q_i}(k)H_{q_i}(k)|\mathcal{D}(k), X(k)=j\right] P(X(k)=j) \\
&= \sum_{j=1}^{T} P(X(k)=j) \sum_{i=1}^{j} 2D_{q_i}(k) \\
&= \sum_{j=1}^{N} P(X(k)=j) \sum_{i=1}^{j} 2D_{q_i}(k) \\
&\qquad + \sum_{j=N+1}^{T} P(X(k)=j) \sum_{i=1}^{N} 2D_{e_i}(k) \\
&= \sum_{j=1}^{N} 2D_{q_j}(k) \sum_{i=j}^{N} P(X(k)=i) \\
&\qquad + \sum_{j=1}^{N} 2D_{q_j}(k) P(X(k) > N) \\
&= \sum_{j=1}^{N} 2D_{q_j}(k) P(X(k) \geq j) \qquad (2.42)
\end{aligned}$$

By some computations, we can rewrite (2.42) *as follows*

$$\begin{aligned}
&\sum_{j=1}^{N} 2D_{q_j}(k) P(X(k) \geq j) \\
&= \sum_{j=1}^{N-1} 2(D_{q_j}(k) - D_{q_{j+1}}(k)) \sum_{n=1}^{j} P(X(k) \geq n) \\
&\qquad + 2D_{q_N}(k) \sum_{n=1}^{N} P(X(k) \geq n) \qquad (2.43)
\end{aligned}$$

Similarly, we can rewrite (2.40) *as follows*

$$\begin{aligned}
&E[\sum_{i=1}^{N} 2m_{q_i}^{-1} D_{q_i}(k) A_{q_i}(k) | \mathcal{D}(k))] \\
&= \sum_{j=1}^{N-1} 2(D_{q_j}(k) - D_{q_{j+1}}(k)) \sum_{n=1}^{j} \frac{a_{q_n}}{m_{q_n}} \\
&\qquad\qquad +2D_{q_N}(k) \sum_{n=1}^{N} \frac{a_{q_n}}{m_{q_n}}
\end{aligned} \tag{2.44}$$

Combining (3.19), (2.43), *and* (2.44), *we obtain*

$$\begin{aligned}
&E[\sum_{i=1}^{N} 2m_{q_i}^{-1} D_{q_i}(k)(D_{q_i}(k+1) - D_{q_i}(k)) | \mathcal{D}(k))] \\
&= \sum_{j=1}^{N-1} 2(D_{q_j}(k) - D_{q_{j+1}}(k)) \sum_{n=1}^{j} \frac{a_{q_n}}{m_{q_n}} \\
&\quad +2D_{q_N}(k) \sum_{n=1}^{N} \frac{a_{q_n}}{m_{q_n}} \\
&\quad - \sum_{j=1}^{N-1} 2(D_{q_j}(k) - D_{q_{j+1}}(k)) \sum_{n=1}^{j} P(X(k) \geq n) \\
&\quad -2D_{q_N}(k) \sum_{n=1}^{N} P(X(k) \geq n) \\
&= \sum_{j=1}^{N-1} 2\left(D_{q_j}(k) - D_{q_{j+1}}(k)\right) \\
&\qquad \sum_{n=1}^{j} (\frac{a_{q_n}}{m_{q_n}} - P(X(k) \geq n)) \\
&\quad +2D_{q_N}(k) \sum_{n=1}^{N} (\frac{a_{q_n}}{m_{q_n}} - P(X(k) \geq n))
\end{aligned} \tag{2.45}$$

Define

$$c = \max_{D \subset \{1,\ldots,N\}} \left\{ \sum_{i \in D} \frac{a_{q_n}}{m_{q_n}} - \sum_{n=1}^{T} P(X(k) > n) \right\} \tag{2.46}$$

which is a negative constant because of (2.30). *This, combining with* (2.45) *and the fact*

$T < N$, implies that

$$
\begin{aligned}
&E[\sum_{i=1}^{N} 2m_{q_i}^{-1} D_{q_i}(k)(D_{q_i}(k+1) - D_{q_i}(k))|\mathcal{D}(k))] \\
&\leq \sum_{j=1}^{N-1} 2(D_{q_j}(k) - D_{q_{j+1}}(k))c + 2D_{q_N}(k)c \\
&= 2D_{q_1}(k)c
\end{aligned} \tag{2.47}
$$

From (3.38), (3.17), *and* (3.40) *we have*

$$
\begin{aligned}
&E[L(\mathcal{D}(k+1)) - L(\mathcal{D}(k))|\mathcal{D}(k)] \\
&\leq \sum_{i=1}^{N} m_i^{-1} E[R_i^2(k)] + 1 + 2D_{q_1}(k)c \\
&\leq \sum_{i=1}^{N} m_i^{-1} E[R_i^2(k)] + 1 + 2c\sum_{i=1}^{N} D_i(k)
\end{aligned} \tag{2.48}
$$

By Foster's criterion for ergodic Markov chain, the debt queueing length process converges in distribution. Using iterated mean and telescoping sums, we have

$$
\sum_{i=1}^{N} E[D_i(k)] \leq -\frac{1}{2c}(\sum_{i=1}^{N} m_i^{-1} E[R_i(k)^2] + 1) \tag{2.49}
$$

CHAPTER 3

Throughput Optimal Scheduling with Heavy-Tailed Traffic

3.1 System Model and Preliminaries

3.1.1 Preliminaries

In this paper we use the following notations. For any two real functions $a(t)$ and $b(t)$, we let $a(t) \sim b(t)$ denote $\lim_{t\to\infty} a(t)/b(t) = 1$. For any two non-negative r.v.s X and Y, we say that $X \leq_{a.s.} Y$ if $X \leq Y$ almost surely, and $X \leq_{s.t.} Y$ if X is stochastically dominated by Y, i.e., $P(X > t) \leq P(Y > t)$ for all $t \geq 0$. We say $X \stackrel{d}{=} Y$ if X and Y are equal in distribution. Also, let $F(x) = P(X \leq x)$ denote the cumulative distribution function (cdf) of a non-negative r.v. X. Let $\overline{F}(x) = P(X > x)$ denote its tail distribution function.

Definition 5 *A r.v. X is heavy tailed (HT) if for all $\theta > 0$*

$$\lim_{x\to\infty} e^{\theta x}\overline{F}(x) = \infty, \tag{3.1}$$

Definition 6 *A r.v. X is light tailed (LT) if it is not heavy tailed.*

Remark 1 *Generally speaking, a r.v. is HT if its tail distribution decreases slower than exponentially. On the contrary, a r.v. is LT if its tail distribution decreases exponentially or faster. Some typical HT distributions include Pareto and log-normal, while the typical LT distributions include exponential and Gamma.*

Based on the existence of the moments, we define the tail coefficient of a non-negative random variable.

Definition 7 *The tail coefficient κ_X of a nonnegative random variable X is defined by*

$$\kappa_X = \sup\{k \geq 0 : E[X^k] < \infty\} \tag{3.2}$$

In other words, the tail coefficient defines the threshold order above which a random variable has infinite moments. Some HT distributions, such as Pareto, have finite tail coefficient or equivalently have infinite moments of certain orders, while some HT distribution, such as log-normal, may have infinite tail coefficient or equivalently have finite moments of all orders. In this research, we focus on the random variables which are heavy tail distributed with finite tail coefficient.

An important subclass of HT distributions which have finite tail coefficient is the regularly varying distributions

Definition 8 *A r.v. X is called regularly varying with tail index $c > 0$, denoted by $X \in \mathcal{RV}(\beta)$, if*

$$\overline{F}(x) \sim x^{-\beta}\mathcal{L}(x), \tag{3.3}$$

where $L(x)$ is a slowly varying function.

Regularly varying distributions are a generalization of Pareto/Ziph/power-law distributions and they can effectively characterize a wide range of network attributes, such as the frame length of variable bit rate (VBR) traffic, the session duration of network users in WLANs, and the size of files at internet severs and cellular base stations. The tail index β indicates how heavy the tail distribution is, where smaller values of β imply heavier tail. Moreover, for a r.v. $X \in \mathcal{RV}(\beta)$, the tail coefficient κ_X of X is equal to the tail index β, which defines the maximum order of bounded moments X can have. Specifically, if $0 < \beta < 1$, X has infinite mean and variance. If $1 < \beta < 2$, X has finite mean and infinite variance.

3.1.2 System Model

Consider a fully-connected network with N wireless users sharing a single fading channel. Time is slotted with unit slot size. We capture the channel fading by $R_i(t)$, which is the data rate of user i if user i is transmitting during the time slot t. We assume $R_i(t)$ is random variable with the maximum value of r_i^{max} and mean $E[R_i(t)] = r_i$. We associate a

queue i to each user i. Let $A_i(t)$ denote the number of packets that arrive at queue i during time slot t with average traffic rate or traffic intensity of $E[A_i(t)] = \lambda_i$. We consider that there are two types of queues in the network: heavy queue and light queue.

Definition 9 *(**Heavy Queue**) A queue i belongs to the heavy queue, denoted by $i \in H$, if it has heavy-tailed arrivals, i.e., $A_i(t) \in \mathcal{RV}(\beta)$ with tail index β*

Definition 10 *(**Light Queue**) A queue i belongs to the light queue, denoted by $i \in L$, if it has light-tailed traffic arrivals, i.e., $A_i(t) \in LT$*

3.1.3 Queueing Dynamics

Let $Q_i(t)$ denote the number of packets in the queue i by the end of time slot i. Then, the queueing dynamics of user i can be represented by

$$Q_i(t+1) = Q_i(t) - R_i(t)H_i(t) + A_i(t) \tag{3.4}$$

where $H_i(t)$ is the transmission duration of user i during the time slot t. By defining $S_i(t)$ as the number of packets which depart from queue i at time slot t under any scheduling policy, the queueing dynamics in (3.4) can be rewritten by

$$Q_i(t+1) = Q_i(t) - S_i(t) + A_i(t) \tag{3.5}$$

3.1.4 Moment Stability

Definition 11 *A network user i is strongly stable if its steady-state queue length has bounded mean, i.e,*

$$E[Q_i(t)] < \infty \tag{3.6}$$

A network is strongly stable if there exists a scheduling algorithm, under which all network users are strongly stable.

The following lemma shows that strong stability is not achievable for the network user with heavy-tailed arrivals with tail index less than three.

Lemma 2 *[16]Let Q_i denote the steady-state queue length of user i. If the user i has HT arrivals with tail index β_i, i.e., $A_i(t) \in \mathcal{RV}(\beta_i)$, then under any scheduling algorithms, Q_i necessarily follow HT distribution with the tail coefficient* $\min_{j\in H} \beta_j - 1 < \kappa(Q_i) < \beta_i - 1$.

The Lemma above, combining with Definition 7, shows that under any scheduling algorithm, if a user i has HT traffic arrival with tail index less than 2, i.e., $\beta_i < 2$, then $E[Q_i^2] = \infty$, and if user i has HT traffic arrival with tailed index less than 3, i.e., $\beta_i < 3$, then $E[Q_i^2] = \infty$. The above observations mean that strong stability is difficult to achieve at least for the HT users and there, throughput-optimal scheduling with the respect to strong stability may not exist in the heavy-tailed environment. To address this problem, we now formally introduce moment stability.

Definition 12 *[16] A network is moment stable, if there exists a scheduling algorithm, under which the steady-state queue length of any user with light-tailed arrival has bounded mean and variance, i.e.,if $A_i(t) \in LT \forall i \leq N$, then*

$$E[Q_i(t)] < \infty, E[Q_i^2] < \infty \tag{3.7}$$

3.2 Stability Analysis of Distributed Maximum Weight Scheduling

In this section, we analyze the stability performance of a distributed maximum weight scheduling algorithm in the presence of heavy tailed traffic. Our main objective in this section is to reveal the critical conditions under which DMWS will lead to unbounded queueing delay. More specifically, we will show that when the light queue has the arrival rate $_i$ below a threshold *, its queue length is of finite mean. Otherwise, If $_i$ is larger than a threshold $'$, its expected queue length is infinite. As a consequence, we can conclude that DMWS can only achieve moment stability within a fraction of network stability region.

3.2.1 Distributed Maximum Weight Scheduling Algorithm

Definition 13 ***(DMWS Algorithm)***

Step I*: at the beginning of each time slot t, each user i determines the length of its carrier sensing period τ_i by independently generate an exponentially distributed random variable with the mean* $\exp(-w_i(t))$*, i.e.,*

$$P(\tau_i > x) = \exp\Big(- exp(w_i(t)x\Big) \tag{3.8}$$

and

$$w_i(t) = \min(Q_i(t)R_i(t), w^*) \tag{3.9}$$

where w^* is the predefined maximum value of $w_i(t)$.

Step 2*: during this sensing period τ_i, if no transmissions from other users re detected, the user i transmits its packets until the end of the time slot.*

It can be shown by following 3.6 that by the the average service rate provided to the use i during time slot t is given by

$$E[S_i(t)] = \frac{r_i \exp(w_i)}{Z}(1 - \frac{1}{Z}) \tag{3.10}$$

where $Z = \sum_{i=1}^{N} \exp(w_i)$, $exp(w_i)/Z$ is the probability that user i grabs the channel during t, and $1 - \frac{1}{Z}$ is average transmission time user i can have during the time slot t.

3.2.2 Stability Region

We first derive a feasible stability region of DMWS so that as long as the incoming traffic vector is within such region, the network is steady-state stable.

Theorem 4 *If*

$$\sum_{i=1}^{N} \frac{\lambda_i}{r_i} < \frac{N \exp(w^*) - 1}{N \exp(w^*)} \tag{3.11}$$

then, the network is the steady-state stable, i.e., the queue length process is positive recurrent and thus converges in distribution.

Remark 2 *It can be shown that as the w^* becomes large enough, the stability region of DWMS approaches the largest possible one, i.e., $\sum_{i=1}^{N} \lambda_i/r_i < 1$, as $w^* \to \infty$.*

Proof 5 *Firstly, we define the Lyapunov Function as* $L(t) := \sum_{i=1}^{N} Q_i(t) r_i^{-1}$. *We will show that if* $L(t) \geq Nw^*$, *we have a negative Lyapunov drift, i.e.,*

$$E[L(t+1) - L(t)|\mathcal{Q}] \leq -\epsilon \tag{3.12}$$

where $\mathcal{Q}(t) = (Q_1(t), ..., Q_N(t))$ *is a vector process of queue lengths of* N *users.*

Based on the queue dynamics (3.5), *we obtain*

$$L(t+1) = L(t) - \sum_{i=1}^{N} \frac{S_i(t)}{r_i} + \sum_{i=1}^{N} \frac{A_i(t)}{r_i}$$

Then,

$$E[L(t+1)|\mathcal{Q}] = E[L(t)|\mathcal{Q}] + \sum_{i=1}^{N} \frac{\lambda_i}{r_i} - \sum_{i=1}^{N} \frac{E[S_i(t)]}{r_i}$$

From (3.10), *we get*

$$E[S_i(t)] = \frac{r_i \exp(w_i)}{Z}(1 - \frac{1}{Z})$$

Assume that $Q_i(t) \geq \frac{w^*}{R_i(t)}, \forall i \leq N$, *which by* (3.9) *indicates* $L(t) \geq Nw^*$ *and* $w_i(t) = w^*$. *As a consequence, we have*

$$\sum_{i=1}^{N} \frac{E[S_i(t)]}{r_i} = \frac{N \exp(w^*) - 1}{N \exp(w^*)}$$

which indicates (3.12) *holds, i.e., the Lyapunov function has a negative drift when the queue lengths are large enough. By the Foster's criterion, the network is the steady-state stable and the queue length process converges in distribution.*

3.2.3 Steady-state Queue Length Behavior

Now, we will study the steady-state queue length under DMWS algorithm. Let Q_i denote the steady-state queue length of a queue i.

First, we will show that if the average traffic rate of a light queue is smaller than a threshold, its queue length is of bounded mean. Moreover, it is shown that such behavior is independent of the traffic burstiness, i.e., heavy tailness, of the heavy queues.

Theorem 5 *Under DMWS, the steady-state queue length of light queue i is of finite mean, i.e., $E[Q_i] < \infty$, if*

$$\lambda_i < \frac{N\exp(w^*) - 1}{N^2\exp(w^*)} \tag{3.13}$$

Proof 6 *We define the Lyapunov function:*

$$L(\mathcal{Q}(t)) = r_i^{-1}Q_i(k)^2 \tag{3.14}$$

From the (3.4)*, we have*

$$Q_i(t+1) = Q_i(t) - R_i(t)H_i(t) + A_i(t) \tag{3.15}$$

By some simple calculations, we get the conditional Lyapunov drift

$$\begin{aligned}
&E[L(\mathcal{Q}(t+1)) - L(\mathcal{Q}(t))|\mathcal{Q}(t)] \\
&= E[r_i^{-1}(Q_i(t+1)^2 - Q_i(t)^2)|\mathcal{Q}(t)] \\
&= E[2r_i^{-1}Q_i(t)(Q_i(t+1) - Q_i(t))|\mathcal{Q}(t)] \\
&\quad + E[r_i^{-1}(Q_i(t+1) - Q_i(t))^2|\mathcal{Q}(t)] \\
&= T_I + T_{II}
\end{aligned} \tag{3.16}$$

The second term in the RHS of (3.16) *can be upper bounded by*

$$\begin{aligned}
T_{II} &\leq E[r_i^{-1}(A_i(t))^2|\mathcal{Q}(t)] + 1 \\
&= r_i^{-1}E[A_i(t)^2] + 1.
\end{aligned} \tag{3.17}$$

For the first term of in the RHS of (3.16)*, we have*

$$\begin{aligned}
T_I &= E[2r_i^{-1}Q_i(t)A_i(t)|\mathcal{Q}(t)] - E[r_i^{-1}Q_i(t)H_i(t)R_i(t)|\mathcal{Q}(t)] \\
&= 2\big(\frac{\lambda_i}{r_i} - E[H_i(t)]\big)Q_i(t)
\end{aligned} \tag{3.18}$$

where $E[H_i(t)] = \frac{\exp(w_i)}{Z}(1 - \frac{1}{Z})$.

Suppose that $Q_i(t) > \frac{w^*}{R_i(t)}$*. Then, by* (3.9)*, we have* $w_i(t) = w^*$*, which implies*

$$E[H_i(t)] \geq \frac{N\exp(w^*) - 1}{N^2\exp(w^*)}$$

This, combining with Theorem's condition, yields

$$T_I = -2\epsilon Q_i(t) \tag{3.19}$$

where ϵ is a positive value. It follows from (3.16), (3.17), *and* (3.19) *that*

$$\begin{aligned} &E[L(\mathcal{Q}(t+1)) - L(\mathcal{Q}(t))|\mathcal{Q}(t)] \\ &\leq -2\epsilon Q_i(t) + \frac{E[A_i(t)^2]}{r_i} + 1 \end{aligned} \tag{3.20}$$

Using iterated mean and telescoping sums, we have

$$E[Q_i(t)] \leq \frac{1}{2\epsilon r_i} E[A_i(t)^2] + 1) \tag{3.21}$$

which completes the proof.

Next, we will prove that if the traffic intensity of a light queue is larger than a certain threshold, then the light queue will necessarily experience HT queue length with unbounded mean.

Theorem 6 *Under DMWS algorithm, the steady-state queue length of light queue i follows heavy tailed distribution with unbounded mean, i.e., $Q_i \in HT$ and $E[Q_i] = \infty$, if*

$$\lambda_i > \frac{r_i}{2} \quad \textit{and} \quad \min_{j \in H} \beta_j < 2 \tag{3.22}$$

Proof 7 *Since the system is steady-state stable, the queue length process is a positive recurrent regenerative process. Define T as the time interval between two consecutive instances when all queues are empty. Since system is positive recurrent, we have $E[T] < \infty$. It follows from the properties of regenerative process that*

$$P(Q_i > \frac{\eta\tau}{2}) = \frac{E\left[\sum_{t=1}^{T} I(Q_i(t) > \frac{\eta\tau}{2})\right]}{E[T]} \tag{3.23}$$

where $\eta = \lambda_i - r_i/2$ and $I(\varsigma)$ is a indicator function and $I(\varsigma) = 1$ if event ς happens.

Assume queue j has the HT arrival with the smallest tail index, i.e., $A_j(t) \in \alpha$ and $\alpha_j = \min_{j \in H}$. Then, let $b > w^*$ denote the amount of traffic arriving at heavy queue j at

time instant 1 *when the system is empty. Let* τ *be the least time that the queue* j *will spend until its queue length is decreased to* w^**. We have*

$$\tau = \frac{b - w^*}{r_i^{max}} \tag{3.24}$$

by which, we have

$$\begin{aligned} & E\left[\sum_{t=1}^{T} I(Q_i(t) > \frac{\eta\tau}{2})\right] \\ & \geq E\left[I(A_j(1) > b)\sum_{t=1}^{T} I(Q_i(t) > \frac{\eta\tau}{2})\right] \\ & \geq P(A_j(1) > b)E\left[\sum_{t=\frac{\tau}{2}}^{\tau} I(Q_i(t) > \frac{\eta\tau}{2})\right] \\ & = P(A_j(1) > r_i^{max}\tau + w^*)\sum_{t=\frac{\tau}{2}}^{\tau} P\left(Q_i(t) > \frac{\eta\tau}{2}\right) \end{aligned} \tag{3.25}$$

By the queueing dynamics (3.5) *and* $Q_i(1) = 0$*, we have*

$$Q_i(t) = \sum_{m=1}^{t}(A_i(t) - S_i(t)) \tag{3.26}$$

which, by applying the weak law of large numbers, yields

$$\begin{aligned} & \lim_{\tau\to\infty}\sum_{t=\frac{\tau}{2}}^{\tau} P\left(Q_i(t) > \frac{\eta\tau}{2}\right) \\ & \geq \lim_{\tau\to\infty}\sum_{t=\frac{\tau}{2}}^{\tau} P\left(\sum_{m=1}^{\frac{\tau}{2}}(A_i(t) - S_i(t)) > \frac{\eta\tau}{2}\right) = \frac{\tau}{2} \end{aligned} \tag{3.27}$$

The last equality holds due to fact that the queue length of heavy queue j *is lager than* w^* *during the time interval* $1 \leq t \leq \tau/2$*, which, by* (3.10)*, indicates* $E[S(t)] < r_i/2$ *when* $1 \leq t \leq \tau/2$*.*

Combining (3.23) *with* (3.25) *and* (3.27)*, it follows from the assumption* $A_j(t) \in \mathcal{RV}(min_{j\in H}\alpha_j)$ *and* $min_{j\in H}\alpha_j < 2$ *that*

$$\lim_{\tau\to\infty}\frac{\log[P(Q_i > \frac{\eta\tau}{2})]}{\log[\frac{\eta\tau}{2}]} \geq -min_{j\in H}\alpha_j + 1 \geq -1 \tag{3.28}$$

which, by applying the Theorem 2 in [2], implies $E[Q_i] = \infty$*.*

3.3 Distributed Maximum Weight-α Scheduling

In this section, we study the distributed distributed Maximum Weight-α Scheduling (DMWS-α) algorithm and prove that DMWS-α is throughput optimal in the presence of heavy tailed traffic, i.e., DMWS-α can guarantee moment stability for the largest possible network stability region.

Definition 14 *(**DMWS-α Algorithm**)*

***Step I**: at the beginning of each time slot t, each user i determines the length of its carrier sensing period τ_i by independently generate an exponentially distributed random variable with the mean $\exp(-w_i(t))$, i.e.,*

$$P(\tau_i > x) = \exp\Big(- exp(w_i(t)x\Big) \tag{3.29}$$

and

$$w_i(t) = \min(Q_i(t)^{\alpha_i} R_i(t), w^*) \tag{3.30}$$

where $\alpha_i = \beta_i - 1$ if user i has HT arrivals, i.e., $A_i \in \mathcal{RV}(\beta_i)$ and $\alpha_i > 3$ if user i has LT arrivals, i.e., $A_i \in LT$

***Step 2**: during this sensing period τ_i, if no transmissions are detected, the user i transmits its packets until the end of the time slot.*

Remark 3 *Similar to DMWS, which is a distributive approximation of maximum weight scheduling algorithm, DMWS-α can be considered as a distributed implementation of maximum-weight- scheduling algorithm [4], [17]*

Theorem 7 *Under DMWS-α algorithm, the network is moment stable if*

$$\sum_{i=1}^{N} \frac{\lambda_i}{r_i} \leq (1-\epsilon)\left(1 - \frac{N}{\exp(\epsilon w^*)}\right) \tag{3.31}$$

Remark 4 *The above theorem indicates DMWS-α algorithm is throughput optimal as w^* approaches infinity. In other words, the stability region of DMWS-α approaches the largest possible one, i.e., $\sum_{i=1}^{N} \lambda_i/r_i \leq 1$, as $w^* \to \infty$.*

Proof 8 *Let $\mathcal{Q}(t) = (Q_1(t), ..., Q_N(t))$ denote a vector process of queue lengths of N users. We define the Lyapunov function:*

$$L(\mathcal{Q}(t)) = \sum_{i=1}^{N} L(Q_i(t)) \tag{3.32}$$

where

$$L(Q_i(t)) = \frac{r_i^{-1} Q_i(t)^{\alpha_i+1}}{\alpha_i + 1} \tag{3.33}$$

We next evaluate each term $L(Q_i(t))$ under two cases:$1 \leq \alpha_i \leq \kappa(A_i(t)) - 1$ and $0 < \alpha_i < 1$. For the first case, using queueing dynamics and Taylor's expansions, we have

$$\begin{aligned} L(Q_i(t+1)) &= \frac{r_i^{-1} Q_i(t+1)^{\alpha_i+1}}{\alpha_i + 1} \qquad (3.34) \\ &= \frac{r_i^{-1}}{\alpha_i + 1}(Q_i(t) + A_i(t) - S_i(t))^{\alpha_i+1} \\ &= r_i^{-1}(\frac{Q_i(t)^{\alpha_i+1}}{\alpha_i + 1} + \Delta_i(t) Q_i(t)^{\alpha_i} + \alpha_i \frac{\Delta_i(t)^2}{2} \delta^{\alpha_i - 1}) \qquad (3.35) \end{aligned}$$

where $\Delta_i(t) = A_i(t) - H_i(t)$ and $\delta = [Q_i(t) - 1, Q_i(t) + A_i(t)]$. Therefore, by the fact that $\Delta_i(t)^2 \leq A_i(t)^2 + 1$ and $(Q_i(t) + A_i(t))^{\alpha_i - 1} < 2^{\alpha_i - 1}(Q_i(t)^{\alpha_i - 1} + A_i(t)^{\alpha_i - 1})$, for any positive constant θ, we have

$$\begin{aligned} & E[L_i(Q_i(t+1)) - L_i(Q(t))|\mathcal{Q}(t)] \\ &= r_i^{-1}(Q_i(t)^{\alpha_i} E[\Delta_i(t)|\mathcal{Q}(t)] + \frac{\alpha_i}{2} E[\Delta_i(t)^2 \delta^{\alpha_i - 1}|\mathcal{Q}(t)]) \\ &\leq r_i^{-1}(E[(A_i(t) - S_i(t) + \theta) Q_i(t)^{\alpha_i}|\mathcal{Q}(t)] + W_i(\theta)) \qquad (3.36) \end{aligned}$$

where

$$\begin{aligned} W_i(\theta) &= \left(\theta^{-1} 2^{\alpha_i - 2} \alpha_i E[A_i(t)^2 + 1]\right)^{\alpha_i - 1} \\ &\quad + 2^{\alpha_i - 2} \alpha_i E[A_i(t)^{\alpha_i + 1} + A_i(t)^{\alpha_i - 1}] \qquad (3.37) \end{aligned}$$

The last inequality in (3.36) *holds because $1 < \alpha_i < \kappa(A_i(t)) - 1$, which implies that $E[A_i(t)^2]$, $E[A_i(t)^{\alpha_i+1}]$, and $E[A_i(t)^{\alpha_i - 1}]$ are finite.*

For the second case $0 < \alpha_i < 1$*, by the similar arguments, we obtain*

$$\begin{aligned} & E[L_i(Q_i(t+1)) - L_i(Q(t))|\mathcal{Q}(t)] \\ & \leq r_i^{-1}(E[(A_i(t) - S_i(t) + \theta)Q_i(t)^{\alpha_i}|\mathcal{Q}(t)] + W_i(\theta)) \end{aligned} \tag{3.38}$$

where

$$W_i(\theta) = \theta + 1 + E[A_i(t)^{\alpha_i+1}]. \tag{3.39}$$

By (3.32)*,* (3.36)*, and* (3.38)*, the conditional Lyapunov drift is upper bounded by*

$$\begin{aligned} & E[L(\mathcal{Q}(t+1)) - L(\mathcal{Q}(t))|\mathcal{Q}(t)] \\ \leq & \sum_{i=1}^{N} \frac{\lambda_i}{r_i} Q_i(t)^{\alpha_i} + \sum_{i=1}^{N} \left(\frac{\theta Q_i(t)^{\alpha_i} + W_i(\theta)}{r_i} \right) \\ & -E\left[\sum_{i=1}^{N} r_i^{-1} S_i(t) Q_i(t)^{\alpha_1} |\mathcal{Q}(t) \right] \\ = & T_I + T_{II} + T_{III} \end{aligned} \tag{3.40}$$

We now evaluate the term T_I *and* T_{III}*. Towards this, we first define the following notations. At each time slot* t*, we arrange the queues in a decreasing order of the weight* $w_i(t) = Q_i(t)^{\alpha_i} R_i(t)$*, i.e.,* $w_1(t), w_2(t), ..., w_N(t)$ *with* $w_i(t) \geq w_{i+1}(t)$*, where ties are broken randomly.*

By some computations, we can rewrite T_I *as follows*

$$\begin{aligned} T_I = & \sum_{i=1}^{N-1} (Q_i(t)^{\alpha_i} - Q_{i+1}(t)^{\alpha_{i+1}}) \sum_{n=1}^{i} \frac{\lambda_n}{r_n} \\ & + Q_N(t)^{\alpha_N} \sum_{n=1}^{N} \frac{\lambda_n}{r_n} \end{aligned} \tag{3.41}$$

We now evaluate T_{III} as follows

$$\begin{aligned}
&T_{III} \\
&\geq E\left[\sum_{i=1}^{N} r_i^{-1} S_i(t) Q_i(t)^{\alpha_i} I\Big(w_i(t) > (1-\epsilon)w^*\Big) \;\Big|\; \mathcal{Q}(t)\right] \\
&\geq (1-\epsilon) Q_1(t)^{\alpha_1} E\left[\sum_{i=1}^{N} H_i(t) I\Big(w_i(t) > (1-\epsilon)w^*\Big) \;\Big|\; \mathcal{Q}(t)\right] \\
&\geq (1-\epsilon) Q_1(t)^{\alpha_1} \left(1 - \frac{N}{\exp(\epsilon w^*)}\right) \\
&= \mu \left(\sum_{i=1}^{N-1} \Big(Q_i(t)^{\alpha_i} - Q_{i+1}(t)^{\alpha_{i+1}}\Big) + Q_N(t)^{\alpha_N}\right) \qquad (3.42)
\end{aligned}$$

where $\mu = (1-\epsilon)\left(1 - \frac{N}{\exp(\epsilon w^*)}\right)$

Combining (3.40), (3.41), *and* (3.42) *then, we obtain*

$$\begin{aligned}
&E[L(\mathcal{Q}(t+1)) - L(\mathcal{Q}(t))|\mathcal{Q}(t)] \\
&\leq \sum_{i=1}^{N-1} \left(Q_i(t)^{\alpha_i} \left(\sum_{n=1}^{i} \frac{\lambda_i}{r_i} - \mu\right)\right) + \sum_{i=1}^{N} \frac{\theta}{r_i} Q_i(t)^{\alpha_i} \\
&\quad + \sum_{i=1}^{N} \frac{W_i(\theta)}{r_i} + Q_N(t)^{\alpha_i} \left(\sum_{i=1}^{N} \frac{\lambda_i}{r_i} - \mu\right) \qquad (3.43)
\end{aligned}$$

By defining

$$d = \max_{Q \subset \{1,\ldots,N\}} \left\{\sum_{i \in Q} \frac{\lambda_i}{r_i} - \mu\right\}$$

which is a negative constant, we can rewrite (3.43) *as*

$$\begin{aligned}
&E[L(\mathcal{Q}(t+1)) - L(\mathcal{Q}(t))|\mathcal{Q}(t)] \\
&\leq (d + \frac{\theta}{r_{min}}) \sum_{i=1}^{N} Q_i(t)^{\alpha_i} + \sum_{i=1}^{N} \frac{W_i(\theta)}{r_i}
\end{aligned}$$

where $r_{min} = \min_{i \leq N} r_i$ *Letting* $\theta = -\frac{d}{2} r_{min}$, *the Lyapunov drift can be bounded by*

$$\begin{aligned}
&E[L(\mathcal{Q}(t+1)) - L(\mathcal{Q}(t))|\mathcal{Q}(t)] \\
&\leq \frac{d}{2} \sum_{i=1}^{N} Q_i(t)^{\alpha_i} + \sum_{i=1}^{N} W_i(-\frac{d}{2} r_{min})
\end{aligned}$$

By Foster's criterion for ergodic Markov chain, the queueing length process converges in distribution. Using iterated mean and telescoping sums, we have

$$\sum_{i=1}^{N} E[Q_i(t)^{\alpha_i}] \leq (-\frac{w}{d}) \sum_{i=1}^{N} W_i(-\frac{d}{2} r_{min})$$

where $W_i()$ is defined in (3.37) *and* (3.39)*, respectively.*

3.4 Simulation Results

In this section, we use simulations to illustrate our theoretical results. More specifically, we choose Pareto and exponential distributions to represent HT and LT distributions, respectively. We say a random variable $X \in \mathcal{PAR}(\alpha, x_m)$ if it follows Pareto distribution with parameters α and x_m, i.e., $P(X > x) = (x_m/x)^{\alpha}$. We say a random variable $X \in \mathcal{EXP}(\lambda)$ if it follows exponential distribution with parameter λ, i.e., $P(X > x) = e^{-\lambda x}$. All the following simulation results are plotted on log-log coordinates, by which *HT* distribution can manifest itself as a straight line with the slope equal to the negative value of the tail index α.

We first evaluate the stability performance of DMWS by considering a scenario where three users are sharing one channel. We assume user 1 is transmitting heavy tailed traffic with traffic arrival processes $A_1(t) \in \mathcal{PAR}(1.5, 1)$ and user 2 and user 3 are sending light tailed traffic with $A_2(t) \in \mathcal{EXP}(1/3)$, $A_3(t) \in \mathcal{EXP}(1/2)$, respectively. Consequently, we have $\lambda_1 = E[A_1(t)] = 3$, $\lambda_2 = E[A_3(t)] = 3$, and $\lambda_3 = E[A_3(t)] = 2$, respectively. Assume the data rate of r of each channel is 18. In this case, $\sum_{i=1^3} \lambda_i < r$,which means the network is steady state stable. Moreover, since $\frac{\lambda_3}{\lambda_1+\lambda_2+\lambda_3} < \frac{1}{3}$. by Theorem 5, user 3 will have bounded average queue length. Specifically, it is shown in Fig 3.1. that the tail distribution of user 3 has a slope or decaying rate smaller than 1, which implies the finiteness of mean of the queue length.

Furthermore, we increase the arrival rate of user 3 to $\lambda_3 = E[A_3(t)] = 8$, According to our Theorem 6, such setting can lead user 3 to have HT queue length. As shown in Fig 3.2, the queue length of user 3 has a tail distribution, which exhibits itself as a straight line

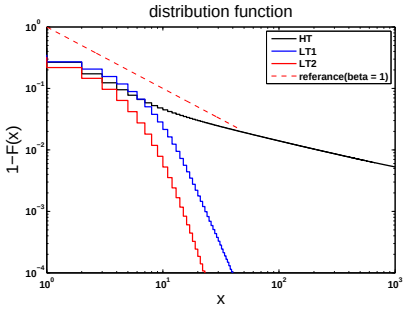

Figure 3.1: Queue-length tail-distribution under DMWS when network is stable

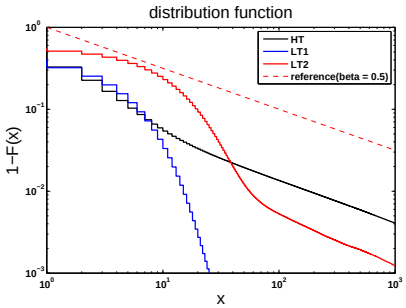

Figure 3.2: Queue-length tail-distribution under DMWS when network is not stable

parallel to that of the reference Pareto distribution with tail index 0.5.This indicates that the queueing length of user 2 has unbounded mean and thus moment stability is not achievable under DMWS. We next show that the moment stability can be achieved by applying the DMWS-α algorithm. More specifically, we assign user 1, user 2 and user 3 with weight $\alpha_1 = \beta 1 = 0.5$, $\alpha_2 = 2$ and $\alpha_3 = 2$, respectively. By Theorem 7, such setting can guarantee that user 2 and user 3 with light tailed traffic necessarily have bounded average queue length. In particular, it is shown in Fig 3.3 that the tail distribution of user 2 and user 3 have a slope or decaying rate smaller than 2, which indicate the queue lengths of both user 2 and user 3 are of finite mean and variance.

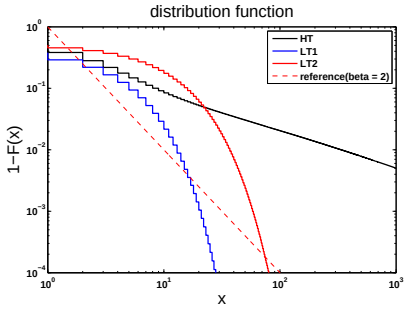

Figure 3.3: Queue-length tail-distribution under DMWS-α

CHAPTER 4

CONCLUSIONS AND FUTURE WORK

In this thesis, we introduce two scheduling algorithms, distributed timely-throughput optimal scheduling and throughput optimal with heavy-tailed traffic scheduling algorithms. The distributed timely-throughput optimal scheduling algorithm, which can aim to achieve maximum network capacity while guaranteeing the hard delay dealines can be met. Towards this, the necessary conditions for the existence of a viable scheduling algorithm to achieve timely-throughput are obtained, which also characterize the maximum achievable capacity region. In addition, we develop a distributed scheduling algorithm by exploiting the principles of Glauber dynamics and we prove that the proposed scheduling algorithm is timely-throughput optimal in the sense that it can guarantee the timely data delivery as long as the incoming traffic rates are within the maximum network capacity region.

In throughput optimal scheduling with heavy-tailed traffic, we analyze the stability performance of distributed maximum weight scheduling algorithms ($DMWS$), which are known to be throughput optimal under light tailed environment, is first analyzed. More specifically, we show that heavy-tailed traffic can significantly degrade the stability performance of DMWS in the sense that it can only guarantee moment stability under a fraction of the network stability region. To address this problem, the distributed maximum weight-α scheduling (DMWS-α) algorithm is proposed, which makes the scheduling decision based on the queue lengths raised to the $\alpha-th$ power. It is demonstrated that DMWS-α is throughput optimal with respect to moment stability.

In future work, expanding the distributed timely-throughput optimal and throughput optimal scheduling with heavy-tailed traffic scheduling from single-hop to multi-hop and keeping the same good performance will be considered. Moreover, proposing routing algorithm protocols to provide hard deadline and moment stability in the presence of heavy-tailed traffic will also be studied.

REFERENCES

REFERENCES

[1] A. Chakravarthy, K. Song, and E. Feron. Preventing automotive pileup crashes in mixed-communication environments. *IEEE Transactions on Intelligent Transportation Systems,*, 10(2):211–225, 2009.

[2] D. J. Daley. The moment index of minima 2. *J. Appl. Probab.*, 38(A):33– 36, 2001.

[3] A. Ghosh, R. Jana, V. Ramaswami, J. Rowland, and N. K. Shankaranarayanan. Modeling and characterization of large-scale wi-fi traffic in public hot-spots. In *Proc. IEEE INFOCOM 2011*, pages 2921 – 2929, Shanghai, China, 2011.

[4] L. Jiang and J. Walrand. A distributed csma algorithm for throughput and utility maximization in wireless networks. *IEEE/ACM Transactions on Networking*, 18(3):960–972, 2010.

[5] Frank P Kelly. *Reversibility and stochastic networks*. Cambridge University Press, 2011.

[6] F. Martinelli. Lectures on glauber dynamics for discrete spin models. In *Lectures on probability theory and statistics*, pages 93–191. Springer, 1999.

[7] 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*, 20(3):825–836, 2012.

[8] K. Park and W. Willinger. *Self-Similar Network Traffic and Performance Evaluation*. A Wiley-Interscience Pubilication, 2000.

[9] V. Ramaswami, K. Jain, R. Jana, and V. Aggarwal. Modeling heavy tails in traffic sources for network performance evaluation. *Computational Intelligence, Cyber Security and Computational Models*, 246:23–44, 2014.

[10] J. Tan and N. B. Shroff. Transition from heavy to light tails in retransmission durations. In *IEEE INFOCOM'10*, Mar. 2010.

[11] 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*, 37(12):1936–1948, 1992.

[12] P. Wang and I. F. Akyildiz. Spatial correlation and mobility aware traffic modeling for wireless sensor networks. *IEEE/ACM Transactions on Networking*, 19(6):1860 – 1873, 2011.

[13] P. Wang and I. F. Akyildiz. Network stability of dynamic spectrum access networks in the presence of heavy tailed traffic. In *Proc. IEEE SECON 2012*, pages 165 – 173, Seoul, Korea, 2012.

[14] P. Wang and I. F. Akyildiz. On the origins of heavy tailed delay in dynamic spectrum access networks. *IEEE Transactions on Mobile Computing*, 11(2):204–217, 2012.

[15] P. Wang and I. F. Akyildiz. Improving network connectivity in the presence of heavy-tailed interference. *to appear in IEEE Transactions on Wireless Communications*, 2014.

[16] P. Wang and I. F. Akyildiz. On the stability of dynamic spectrum access networks in the presence of heavy tails. *to appear in IEEE Transactions on Wireless Communications*, 2014.

[17] W. Wu and F. Zhang. A switching strategy for robust cooperative exploration. In *Proceeding of 49th IEEE Conference on Decision and Control (CDC),*, pages 5493–5498. IEEE, 2010.