Emulating Round-Robin in Wireless Networks

Bin Li
Department of ECBE
University of Rhode Island
Kingston, Rhode Island 02881
binli@uri.edu

Atilla Eryilmaz
Department of ECE
The Ohio State University
Columbus, Ohio 43210
eryilmaz.2@osu.edu

R. Srikant
CSL and Department of ECE
UIUC
Urbana, Illinois 61801
rsrikant@illinois.edu

ABSTRACT

Round robin and its variants are well known scheduling policies that are popular in wireline networks due to their throughput optimality, delay insensitivity to file size distributions and short-term fairness. The latter two properties are also extremely important for emerging wireless applications, such as Internet of Things and cyber-physical systems. However, there is no direct wireless analog of round robin with all the desirable properties in wireless networks, where wireless interference and channel fading are predominant. The main reason is due to the fact that it is very difficult to even define what round robin means in wireless networks. This motivates us to develop a round-robin-like algorithm in wireless networks that has nice properties as round robin in wireline networks. To that end, we utilize a counter called the Time-Since-Last-Service (TSLS) that keeps track of the time of each file since its last service, and observe that scheduling a file with maximum TSLS in a single server is equivalent to serving files in a round robin fashion. Based on this key observation, we develop a TSLS-based algorithm that balances the tradeoff between the TSLS value and the channel rate for each link and show that the proposed algorithm achieves maximum system throughput, which demands a nontraditional approach due to the abrupt dynamics of the TSLS metrics. Numerous simulations are provided to validate its desired properties such as delay insensitivity and excellent short-term fairness performance as in the case of round robin algorithms of wireline networks.

CCS CONCEPTS

•Networks →Network resources allocation; Wireless access networks;

KEYWORDS

Wireless scheduling, round robin, throughput, mean delay, service regularity

ACM Reference format:
Bin Li, Atilla Eryilmaz, and R. Srikant. 2017. Emulating Round-Robin in Wireless Networks. In *Proceedings of Mobihoc '17, Chennai, India, July 10-14, 2017,* 10 pages.
DOI: 10.1145/3084041.3084052

Mobihoc '17, Chennai, India

DOI: 10.1145/3084041.3084052

1 INTRODUCTION

Round robin and its variants, such as Weighted Fair Queueing (WFQ) [6], are widely used in wireline networks due to their several attractive properties. First, it is very easy to be implemented. Fig. 1 illustrates the operation of round robin algorithm in a single server system. Second, round robin provides maximum throughput since it continuously utilizes the full capacity of a link and it has an insensitivity property (see [8]). The insensitivity property refers to the fact that the probability distribution of queue lengths (and hence the average delay) is insensitive to the file-size distributions. This property is particularly attractive since it is well-known that file-size distributions are heavy-tailed (or at least have very large variance) and service disciplines other than round robin (such as first-in, first out or FIFO) have mean delays that are very sensitive to file-size distributions (see [10]), which cannot meet the stringent performance needs of diverse network applications. Moreover, round robin provides short-term fairness in the sense that if there are n files, then each file gets $1/n$ fraction of the bandwidth. This short-term fairness can be measured as the variance of inter-service time of each file characterizing how often the file is served, also called service regularity. This together with mean delay and throughput are quite important performance metrics for Internet of things (IoT) and cyber-physical systems.

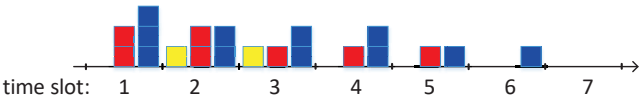


Figure 1: Round robin algorithm in a single server system: At the first slot, there are one "red" file with two packets and one "blue" file with three packets. A new "yellow" file with one packet joins the system at the second slot.

For example, in almost all IoT applications (e.g., health care, smart home, environmental monitoring, and manufacturing), there are a large number of devices, where each device generates traffic that is sparse or intermittent in time but delay sensitive. To see this, consider a smart home application, where some sensors are monitoring the house condition while others monitoring the safety of the house. In such an application, each sensor intermittently updates its status. Hence, in these IoT applications, it is quite important to develop scheduling algorithms that can respond quickly to the service needs of each device while efficiently utilizing the limited wireless resources. One solution is to periodically schedule each device for data transmission; however, one has to trade off this periodicity with the need to adapt unpredictable traffic patterns under the wireless interference and channel fading.

Surprisingly, existing wireless scheduling policies do not mimic the behavior of round robin algorithms exactly or even closely in general wireless networks. One of the first works to study round-robin-type algorithms for wireless networks is [17], where an approximation to WFQ was proposed for wireless networks. The main idea was built around a heuristic to limit the amount by which a flow would lead or lag behind a true WFQ scheduler. However, this algorithm still serves packet at each link in a first-come-first-serve order and thus can suffer large delay when the arrival process is bursty. Moreover, it is only limited to fully-connected networks. Since then, extensive research effort has been exerted in the wireless scheduling design that targets various aspects of Quality-of-Service (QoS), such as throughput (e.g., [1, 14, 18, 21, 23–25]), mean delay (e.g., [7, 16, 22]), and service regularity (e.g., [15, 16]). Motivated by the research activities in bandwidth sharing networks (e.g., [19, 1, 20, 25]), the authors in [2, 5] generalized the ideas of processor-sharing in queueing networks and developed balance fairness schedulers in wireless networks that exhibit delay insensitivity property. However, this balance fairness scheduler requires the full knowledge of the capacity region, leading to the difficulty in practical implementations. More recently, the authors in [19] developed a proportional scheduler that possesses excellent delay performance (see [27]). But, similar to balance fairness scheduler, this scheduler also requires the full knowledge of the capacity region, and thus faces the same implementation issues in practical networks. In [3], the authors developed a flow-aware CSMA algorithm, where each dynamic flow attempts to access the wireless channel after some random time and transmits a packet if the channel is sensed idle. This algorithm serves each file in a fair way and can be regarded as the randomized version of round robin algorithm. Unfortunately, this CSMA algorithm is hard to extend to the case of wireless fading, which is the predominant phenomenon in wireless networks. Indeed, as pointed out in [13], the existing throughput-optimal CSMA algorithms heavily rely on the fact that the CSMA parameters do not change significantly over time so that the instantaneous service rate distribution can stay close to the stationary distribution, which is violated under channel fading. Another interesting work [9] developed MAC-layer queue-length-based policy that essentially runs MaxWeight across queues based on the number of their files and serves files within each queue in a round-robin fashion. Although this scheduler exhibits good mean delay performance over previous scheduling rules, this scheduler does not perform round robin across queues, and fail to achieve good short-term fairness.

The main reason behind the difficulty in developing a wireless counterpart of round robin algorithm is the wireless interference as well as the fact that channel conditions of different users can be widely different, and thus, unlike wireline networks, it is very difficult to even define what round robin means in wireless networks, let alone to develop such algorithms. This motivates us to develop a round-robin-like wireless scheduling algorithm that possesses the desired properties of round robin algorithm in wireline networks. The closest prior work on emulating round robin in wireless networks is the so-called proportional fair scheduler [11]. In the case of downlink wireless transmission, proportional fair scheduler suggests transmitting to the user which has the largest value of the following index: the index is the ratio of the maximum possible data rate that can be realized under the current channel state to the average rate that the user has received so far. The proportional fair scheduling rule maximizes the sum of the log mean rates of the users and it is known that this algorithm specialized to the case of wireline networks reduces to the round robin scheme. Thus, in one sense, this scheduler can be viewed as a natural generalization of round robin. However, proportional fairness achieves only long-term fairness and does not have tight control over short-term fairness as round robin does in wireline networks. Further, proportional fairness is not delay insensitive to file-size distributions unless one makes significant approximations in modeling the implementation of proportional fairness. To the best of our knowledge, delay insensitivity properties of proportional fairness (e.g., [2, 4, 5]) have been established only under the assumption that the achieved rate of a user is equal to its mean available rate, whereas in reality, these two quantities can be quite different. Additionally, proportional fairness cannot be easily implemented in distributed settings. Another interesting work [26] proposed a wireless scheduling algorithm that guarantees a maxmin fairness across links. However, it is totally unclear how to extend this algorithm to the case with channel fading. More recent work [20] developed distributed randomized TDMA scheduling for wireless networks. But, this algorithm does not adapt the dynamic network traffic.

Our work is motivated by the recent observations in [15, 16]: in a single-server system, if we maintain a Time-Since-Last-Service (TSLS) counter for each file that keeps track of how much time has been passed since its last service, then serving the file with the maximum TSLS is exactly equivalent to the round robin algorithm. Indeed, in the steady-state, the TSLS vector in a single-server system under the round robin algorithm is a permutation of $\{0, 1, 2, \ldots, N-1\}$, where N is the number of files in the system. Thus, this provides a promising method to emulate round robin algorithms for wireless networks taking into account wireless interference and channel fading. However, it is still unclear how to balance the weight of each link between the TSLS and the channel rate to achieve the desired properties of round robin algorithms in wireline networks. The following items list our main contributions along with references to where they appear in the text:

- In Section 2, we introduce the time-since-last-service counter for each file and establish a tight relationship between the time-since-last-service and service regularity performance in the presence of flow-level dynamics.
- In Section 3, we develop a novel scheduling algorithm that mimics the round robin algorithm, and show that the proposed scheduling algorithm achieves throughput optimality.
- Comparative simulations are provided in Section 4 to show that our proposed algorithm exhibits excellent delay and service regularity performance compared to the existing algorithms.

2 SYSTEM MODEL

We consider a wireless network with K links, where each link represents a transmitter-receiver pair that are within the transmission range of each other. We assume that the system operates in slotted time with normalized slots $t \in \{1, 2, \ldots\}$. Due to the wireless interference, the success or failure of a transmission over a link depends on whether one of its interfering links is also active in the same slot, which is called the link-based conflict model. We call a set of

links that can be active simultaneously a feasible schedule. Let Ω be the collection of feasible link schedules.

We use $A_i[t]$ to denote the number of files arriving at link i in slot t that are independently distributed over links, and independently and identically distributed (i.i.d.) over time with finite mean $\lambda_i > 0$, and $A_i[t] \le A_i^{\max}$ for some $A_i^{\max} < \infty, \forall i, t \ge 0$. Also, these newly arriving files cannot be scheduled until the next time slot. We use $F_{i,j}[t]$ to denote the number of packets of file j arriving at link i that follows any random distribution with finite mean $\eta_i > 0$ and $F_{i,j}[t] \le F_i^{\max}$ for some $F_i^{\max} < \infty$. Also, we assume that $q_i \triangleq \Pr\{F_{i,j}[t] = F_i^{\max}\} > 0, \forall i = 1, 2, ..., K$. Let $\rho_i \triangleq \lambda_i \eta_i$ be the traffic intensity at link i.

Let $\mathcal{N}_i[t]$ be the set of files at link i in time slot t. We use $S_{i,j}[t] = 1$ to denote that the file j at link i is scheduled in slot t and $S_{i,j}[t] = 0$ otherwise. We call $\mathrm{S} = (S_{i,j}[t], j \in \mathcal{N}_i[t], i = 1, 2, \ldots, K)$ the feasible file schedule denoting which files can be served simultaneously at time t. Let $\mathcal{S}$ be the collection of feasible file schedules. With a little abuse of notation, we also use $S_i[t]$ to denote whether the service is provided at link i in time slot t. We use $R_{i,j}[t]$ to denote the number of residual packets of file j at link i in slot t. The file leaves the system once all its packets have been served, i.e., its residual file size reduces to 0.

We assume that each link i independently experiences i.i.d. ON-OFF channel fading over time with $C_i[t] = 1$ denoting that the channel is available for link i at time t. Let $\mathbf{C}[t] = (C_i[t])_{i=1}^K$ be the channel state at time t and $\phi_{\mathbf{c}} \triangleq \Pr\{\mathbf{C}[t] = \mathbf{c}\}$ denote the probability that the channel state is $\mathbf{c}$ at time t, where $\mathbf{c}$ is $K-$dimensional. Let C be the collection of all possible channel state vectors of 0s and 1s. Then, the capacity region is defined as $\Lambda \triangleq \sum_{\mathbf{c} \in C} \phi_{\mathbf{c}} \cdot \mathrm{Conv}\{\mathbf{c} \cdot \Omega\}$, where $\mathrm{Conv}\{\mathcal{A}\}$ is the convex hull of the set $\mathcal{A}$ and $\mathbf{x} \cdot \mathbf{y}$ denotes the componentwise product of the vectors $\mathbf{x}$ and $\mathbf{y}$.

We call the system *stable* if the underlying Markov Chain is positive recurrent. It has been shown in [23, 24] that the capacity region Λ gives upper bounds on the achievable link rates in packets per slot that can be supported by the network under stability for a given interference model. We say that a scheduler is *throughput-optimal* if it achieves the network stability for any traffic intensity vector $\boldsymbol{\rho} \triangleq (\rho_i)_{i=1}^K$ that lies strictly inside the capacity region Λ.

In addition to the throughput performance, in this work, we are also interested in the mean delay and service regularity of each file. The mean file delay is defined as the expected time for each file to complete its service after it arrives at the system. The service regularity performance measures how often each file gets service, which together with mean file delay are extremely important for real-time applications. As pointed by our work [15, 16], the service regularity performance is highly related to the statistics of the inter-service time. To that end, we use $I_{i,j}[m]$ to denote the time between the $(m-1)^{th}$ and the m^{th} service for file j at link i. We use the variance of inter-service time of files to characterize the service regularity performance. The smaller the variance of the inter-service time, the more regular the service. Our goal is to develop a round-robin-like algorithm that possesses good throughput, small delay and excellent service regularity performance that meets the stringent needs of fast growing wireless applications.

3 WIRELESS ROUND ROBIN SCHEDULING

In this section, we develop a round-robin-like scheduling algorithm that is throughput-optimal in general wireless networks.

3.1 Algorithm Description

Motivated by [15, 16], we introduce a counter $T_{i,j}[t]$ for file j at link i at time t, namely Time-Since-Last-Service (TSLS), to keep track of the time since file j at link i was last served. In particular, $T_{i,j}$ increases by 1 in each time slot when file j at link i does not transmit one packet, either because it is not scheduled or because the channel at link i is not available, and drops to 0 otherwise. More precisely, the evolution of $T_{i,j}$ is described as follows:

$$T_{i,j}[t+1] = \left(T_{i,j}[t] + 1\right)\left(1 - S_{i,j}[t] C_i[t]\right). \tag{1}$$

Thus, the TSLS records the file's "age" since the last time it received service, and its evolution is tightly related to the inter-service time. Indeed, we can establish a key relationship between the mean TSLS and the second moment of inter-service time.

Lemma 3.1. *For any policy under which the system is stable (i.e., the expected number of files is finite), we have*

$$\mathbb{E}\left[\overline{T}_i\right] = \frac{\mathbb{E}\left[\overline{I}_i\left(\overline{I}_i - 1\right)\right]}{2\mathbb{E}\left[\overline{I}_i\right]},$$

where $\overline{T}_i$ and $\overline{I}_i$ denote the time-average TSLS and inter-service time of all files that arrive at link i, respectively.

The proof is available in Appendix A. Although Lemma 3.1 is closely related to [15, Lemma 1], the implementation scenarios are quite different. Indeed, in [15, Lemma 1], a TSLS counter is maintained for each link and a relation is established between its mean value and the second moment of inter-service time for each link, while in this paper we maintain a TSLS counter for each file at each link. In this sense, Lemma 3.1 can be regarded as the flow-level version of [15, Lemma 1].

Lemma 3.1 provides a method to improve service regularity performance by keeping the mean TSLS low, which can be achieved by serving the file with the maximum TSLS at each time. In fact, in fully-connected non-fading networks, serving the file with the maximum TSLS is exactly equivalent to the round robin algorithm. However, it is not clear how to develop the TSLS-based algorithm further to mimic the behavior of round robin algorithm in general wireless networks that take both wireless interference and channel fading into account. In such a case, we need to carefully balance the weight for each file between the TSLS and its channel rate. To facilitate the flexibility in the algorithm design, we define a set of functions:

$\mathcal{F} \triangleq$ set of non-negative, increasing, differentiable and concave functions $f(\cdot) : \mathbb{R}_+ \to \mathbb{R}_+$ with $f(0) = 0$ and $\lim_{y \to \infty} f(y) = \infty$.

$$\mathcal{G} \triangleq \{f \in \mathcal{F} : \text{for any } k \ge 1, b \ge 0, \text{there exists a constant } c > 0 \text{ such that } f(kx + b) \le f(x) + c, \forall x \ge 0\}.$$

Roughly speaking, $\mathcal{G}$ is a class of logarithmic functions, such as $f(x) = \log(1+x)$, $f(x) = \log\log(e+x)$, and $f(x) = \log(1+x)/g(x)$, where $g(x)$ is an arbitrary positive, increasing and differentiable

function which makes $f(x)$ an increasing concave function. Next, we propose the following TSLS-based algorithm.

Algorithm 1 (TSLS-Based Algorithm). *In each time slot t, select a feasible schedule $\mathbf{S}^*[t] \in \mathcal{S}$ such that*

$$\mathbf{S}^*[t] \in \arg\max_{\mathbf{S}[t]\in\mathcal{S}} \sum_{i=1}^{K} \sum_{j\in\mathcal{N}_i[t]} f\left(T_{i,j}[t]\right) C_i[t] S_{i,j}[t], \tag{2}$$

where $f \in \mathcal{G}$.

Under the TSLS-based algorithm, only the file with the maximum TSLS at each link can be served, and thus the TSLS-based algorithm serves files at each link in a round robin fashion and run MaxWeight across links based on their product of the logarithmic function of maximum TSLS and channel rate. It is clear that the TSLS-based algorithm is exactly the round robin algorithm in fully-connected networks with non-fading.

It is worth mentioning that the TSLS-based algorithm can be easily amended for existing protocol design. At each link, there is usually a two-layer structure for controlling file transfer: transport and MAC layers. Transport layer controls the packet injection into the MAC layer, while the MAC layer makes the scheduling decisions to transmit the MAC-layer packets. The TSLS-based algorithm corresponds to the case when the transport layer of each file injects one packet into the MAC layer whenever the MAC layer has no packet belonging to this file. Hence, the MAC layer always contains one packet of each existing file. In this sense, the TSLS $T_{i,j}[t]$ can be interpreted as the delay of the packet of file j in the MAC layer at link i in slot t. Next, we show that the proposed TSLS-based algorithm is throughput optimal.

Proposition 3.2. *The TSLS-based Algorithm is throughput-optimal, i.e., it stabilizes the system for any traffic intensity vector $\boldsymbol{\rho}$ that is strictly within the capacity region Λ.*

Here, we would like to point out that the proof of throughput optimality of our proposed TSLS-based policy is quite different from that of traditional queue-length-based policies, whose proofs are based on the drift analysis of a quadratic Lyapunov function or its variants. Note that the TSLS of a file resets to zero whenever this file is served, no matter how large its previous value was. Therefore, the dynamics of TSLS are quite abrupt and are fundamentally different from that of queue-lengths, which poses significant challenges on the proof of throughput optimality of TSLS-based policies. Indeed, in [15], we proposed a MaxWeight policy using a weight of each link that is a linear combination of queue-length and TSLS, and showed its throughput optimality through a new Lyapunov function. But, in that work, the queue-length still plays an important role in the decision making and thus its throughput optimality is not quite surprising. In contrast, our proposed algorithm is only based on the TSLS values and does not depend on the queue-length in an obvious manner, and thus its throughput optimality is more challenging to establish. Nevertheless, we successfully tackle this difficulty in the following subsection.

3.2 Proof of Throughput-Optimality

In this subsection, we show that the proposed TSLS-based Algorithm is throughput-optimal in the sense that it stabilizes the system for any traffic intensity vector $\boldsymbol{\rho}$ under which there exists a scheduling algorithm stabilizing it.

We first establish an important fact that the difference between the expected logarithmic function of maximum TSLS and that of maximum age is constant, which is the key in establishing the throughput optimality of the TSLS-based Algorithm. To that end, we use $W_{i,j}[t]$ to denote the age of file j at link i in slot t. Hence, $W_{i,j}[t]$ always increases by 1 if its corresponding file does not leave the system, and resets to 0 otherwise. More precisely, the evolution of $W_{i,j}[t]$ can be written as

$$W_{i,j}[t+1] = \left(W_{i,j}[t]+1\right)\left(1-\mathbb{1}_{\{R_{i,j}[t+1]=0\}}\right), \tag{3}$$

where $\mathbb{1}_{\{\cdot\}}$ is an indicator function and we recall that $R_{i,j}[t]$ is the residual size of file j at link i in time slot t.

Lemma 3.3. *Under the TSLS-based Algorithm, for each link i, given any $\delta_i \in (0,1)$, there exists a $G_i > 0$ such that*

$$\mathbb{E}\left[f\left(W_i^{\max}[t]\right)\right] \le \frac{1}{1-\delta_i}\mathbb{E}\left[f\left(T_i^{\max}[t]\right)\right] + G_i, \forall t \ge 0, \tag{4}$$

where $W_i^{\max}[t] \triangleq \max_{j\in\mathcal{N}_i[t]} W_{i,j}[t]$ and $T_i^{\max}[t] \triangleq \max_{j\in\mathcal{N}_i[t]} T_{i,j}[t]$ denote the maximum age and maximum TSLS of files at link i in time slot t, respectively.

Proof. Since there are at most $A_i^{\max}$ files arriving at link i in each time slot, at most $A_i^{\max}+1$ files have the same TSLS value at link i, which implies that

$$|\mathcal{N}_i[t]| \le (A_i^{\max}+1)T_i^{\max}[t], \forall t \ge 0, \forall i = 1,2,\ldots,K, \tag{5}$$

hold for any sample path, where $|\mathcal{A}|$ denotes the cardinality of set $\mathcal{A}$.

In addition, files are served in round robin order within each link under the TSLS-based algorithm, and thus for any sample path, files with the size of $F_i^{\max}$ that came after time slot $\tau = t - W_i^{\max}[t]$ do not leave link i in time slot t, i.e.,

$$\sum_{\tau=t-W_i^{\max}[t]+1}^{t-1} \sum_{j=1}^{A_i[\tau]} \mathbb{1}_{\{F_{i,j}[\tau]=F_i^{\max}\}} \le |\mathcal{N}_i[t]|. \tag{6}$$

By combining (5) and (6), we have

$$\sum_{\tau=t-W_i^{\max}[t]+1}^{t-1} \sum_{j=1}^{A_i[\tau]} \mathbb{1}_{\{F_{i,j}[\tau]=F_i^{\max}\}} \le \left(A_i^{\max}+1\right) T_i^{\max}[t], \tag{7}$$

holding for any sample path.

Given $W_i^{\max}[t]$, $\sum_{j=1}^{A_i[\tau]} \mathbb{1}_{\{F_{i,j}[\tau]=F_i^{\max}\}}$, $\tau = t-W_i^{\max}[t]+1, \ldots, t-1$, are i.i.d, and thus according to the Law of Large Numbers, given any $\delta_i \in (0,1)$, $\exists H_i > 1$ such that for any $W_i^{\max}[t] > H_i$, we have

$$\Pr\left\{\sum_{\tau=t-W_i^{\max}[t]+1}^{t-1} \sum_{j=1}^{A_i[\tau]} \mathbb{1}_{\{F_{i,j}[\tau]=F_i^{\max}\}} \ge \frac{\lambda_i q_i}{2}\left(W_i^{\max}[t]-1\right)\right\}$$
$$\ge 1-\delta_i, \tag{8}$$

where we recall that $\lambda_i \triangleq \mathbb{E}[A_i[t]] > 0$ and $q_i \triangleq \Pr\{F_{i,j}[t] = F_i^{\max}\} > 0$.

By combining (7) and (8), we have

$$\Pr\left\{\frac{\lambda_i q_i}{2}\left(W_i^{\max}[t]-1\right)\mathbb{1}_{\{W_i^{\max}[t]>H_i\}} \le \left(A_i^{\max}+1\right)T_i^{\max}[t]\middle| W_i^{\max}[t]\right\} \ge 1-\delta_i. \quad (9)$$

If $\frac{\lambda_i q_i}{2}\left(W_i^{\max}[t]-1\right)\mathbb{1}_{\{W_i^{\max}[t]>H_i\}} \le \left(A_i^{\max}+1\right)T_i^{\max}[t]$, we have

$$W_i^{\max}[t]\mathbb{1}_{\{W_i^{\max}[t]>H_i\}} \le \frac{2\left(A_i^{\max}+1\right)}{\lambda_i q_i}T_i^{\max}[t]+1, \quad (10)$$

which implies

$$\begin{aligned} W_i^{\max}[t] =& W_i^{\max}[t]\mathbb{1}_{\{W_i^{\max}[t]>H_i\}} + W_i^{\max}[t]\mathbb{1}_{\{W_i^{\max}[t]\le H_i\}} \\ \le& \frac{2\left(A_i^{\max}+1\right)}{\lambda_i q_i}T_i^{\max}[t]+1+H_i. \end{aligned} \quad (11)$$

Note that $\frac{(A_i^{\max}+1)}{\lambda_i q_i} \ge \frac{A_i^{\max}}{\lambda_i} \ge 1$. Since $f \in \mathcal{G}$, there exists a constant $G_i \ge 0$ such that

$$f\left(W_i^{\max}[t]\right) \le f\left(T_i^{\max}[t]\right)+G_i. \quad (12)$$

Hence, we have

$$\begin{aligned} &\Pr\left\{f\left(W_i^{\max}[t]\right) \le f\left(T_i^{\max}[t]\right)+G_i\middle|W_i^{\max}[t]\right\} \\ \ge& \Pr\left\{\frac{\lambda_i q_i}{2}\left(W_i^{\max}[t]-1\right)\mathbb{1}_{\{W_i^{\max}[t]>H_i\}} \le \left(A_i^{\max}+1\right)T_i^{\max}[t]\middle|W_i^{\max}[t]\right\} \\ \ge& 1-\delta_i, \end{aligned} \quad (13)$$

which implies

$$\begin{aligned} &\mathbb{E}\left[f\left(T_i^{\max}[t]\right)\middle|W_i^{\max}[t]\right] \\ \ge& \mathbb{E}\left[f\left(T_i^{\max}[t]\right)\middle|f\left(T_i^{\max}[t]\right) \ge f\left(W_i^{\max}[t]\right)-G_i, W_i^{\max}[t]\right](1-\delta_i) \\ \ge& \left(f\left(W_i^{\max}[t]\right)-G_i\right)(1-\delta_i) \end{aligned} \quad (14)$$

Hence, we have

$$f\left(W_i^{\max}[t]\right) \le \frac{1}{1-\delta_i}\mathbb{E}\left[f\left(T_i^{\max}[t]\right)\middle|W_i^{\max}[t]\right]+G_i, \quad (15)$$

which implies the desired result by taking expectation on both sides. □

Next, we provide a simple inequality that will be useful in the main proof.

Lemma 3.4. *For any $f \in \mathcal{F}$, we have*

$$\sum_{m=M_L}^{M_U} f'(m) \le f(M_U)-f(M_L)+f'(1), \quad (16)$$

holding for any $M_U \ge M_L \ge 1$.

Proof. The proof basically follows from the Fundamental Theorem of Calculus and is similar to [14, Lemma 1]. □

We are ready to prove Proposition 3.2. As the Lyapunov function, we take the total sum of function of age of all files, which are currently in the system. Mathematically, we select the Lyapunov function $V(\mathbf{R},\mathbf{W}) \triangleq \sum_{i=1}^{K}\sum_{j\in\mathcal{N}_i} R_{i,j}f\left(W_{i,j}\right)$. Then, we have

$$\begin{aligned} \Delta V[t] \triangleq& V(\mathbf{R}[t+1],\mathbf{W}[t+1]) - V(\mathbf{R}[t],\mathbf{W}[t]) \\ =& \sum_{i=1}^{K}\left(\sum_{j\in\mathcal{N}_i[t+1]} R_{i,j}[t+1]f\left(W_{i,j}[t+1]\right)\right. \\ & \left. - \sum_{j\in\mathcal{N}_i[t]} R_{i,j}[t]f\left(W_{i,j}[t]\right)\right). \end{aligned} \quad (17)$$

For each link $i = 1, 2, \ldots, K$, we have

$$\begin{aligned} &\sum_{j\in\mathcal{N}_i[t+1]} R_{i,j}[t+1]f\left(W_{i,j}[t+1]\right) \\ =& \sum_{j\in\mathcal{N}_i[t]} R_{i,j}[t+1]f\left(W_{i,j}[t+1]\right) + \sum_{j=1}^{A_i[t]} R_{i,j}[t+1]f\left(W_{i,j}[t+1]\right) \\ \overset{(a)}{=}& \sum_{j\in\mathcal{N}_i[t]} f\left(\left(W_{i,j}[t]+1\right)\left(1-\mathbb{1}_{\{R_{i,j}[t+1]=0\}}\right)\right)R_{i,j}[t+1] \\ & + \sum_{j=1}^{A_i[t]} F_{i,j}[t]f(1) \\ \overset{(b)}{=}& \sum_{j\in\mathcal{N}_i[t]} f\left(W_{i,j}[t]+1\right)R_{i,j}[t+1]\left(1-\mathbb{1}_{\{R_{i,j}[t+1]=0\}}\right) \\ & + f(1)\sum_{j=1}^{A_i[t]} F_{i,j}[t] \\ =& \sum_{j\in\mathcal{N}_i[t]} f\left(W_{i,j}[t]+1\right)R_{i,j}[t+1] + f(1)\sum_{j=1}^{A_i[t]} F_{i,j}[t] \\ \overset{(c)}{=}& \sum_{j\in\mathcal{N}_i[t]} \left(f\left(W_{i,j}[t]\right)+f'\left(x_{i,j}\right)\right)\left(R_{i,j}[t]-C_i[t]S_{i,j}^*[t]\right) \\ & + f(1)\sum_{j=1}^{A_i[t]} F_{i,j}[t] \\ \overset{(d)}{\le}& \sum_{j\in\mathcal{N}_i[t]} R_{i,j}[t]f\left(W_{i,j}[t]\right) + \sum_{j\in\mathcal{N}_i[t]} R_{i,j}[t]f'\left(W_{i,j}[t]\right) \\ & - \sum_{j\in\mathcal{N}_i[t]} f\left(W_{i,j}[t]\right)C_i[t]S_{i,j}^*[t] + f(1)\sum_{j=1}^{A_i[t]} F_{i,j}[t], \end{aligned} \quad (18)$$

where step (a) uses equation (3) and the fact that the newly arriving files are not served in the current slot; (b) follows from the assumption that $f(0) = 0$; (c) uses the Mean Value Theorem for some $x_{i,j}$ between $W_{i,j}[t]$ and $W_{i,j}[t]+1$, and the fact that $S_{i,j}^*[t] \le 1$ and $R_{i,j}[t] \ge 1$, for any $j \in \mathcal{N}_i[t]$; (d) follows from the fact that $f'(y)$ is non-increasing and non-negative due to the function $f(y)$ being increasing and concave for any $y \ge 0$.

By substituting (18) into (17), we have

$$\begin{aligned}\Delta V[t] \leq & \sum_{i=1}^{K}\left(\sum_{j\in\mathcal{N}_i[t]} R_{i,j}[t]f'\left(W_{i,j}[t]\right) - \sum_{j\in\mathcal{N}_i[t]} f\left(W_{i,j}[t]\right)C_i[t]S^*_{i,j}[t]\right) \\ & + f(1)\sum_{i=1}^{K}\sum_{j=1}^{A_i[t]} F_{i,j}[t] \\ \leq & \sum_{i=1}^{K}\left(\sum_{j\in\mathcal{N}_i[t]} R_{i,j}[t]f'\left(W_{i,j}[t]\right) - \sum_{j\in\mathcal{N}_i[t]} f\left(T_{i,j}[t]\right)C_i[t]S^*_{i,j}[t]\right) \\ & + f(1)\sum_{i=1}^{K}\sum_{j=1}^{A_i[t]} F_{i,j}[t], \end{aligned} \tag{19}$$

where the last step uses the fact that the TSLS value is always not greater than the age of its associated flow, i.e., $T_{i,j}[t] \leq W_{i,j}[t]$ for any $j \in \mathcal{N}_i[t]$, $i = 1, 2, ..., K$ and $t \geq 0$. Thus, we have

$$\begin{aligned}\mathbb{E}\left[\Delta V[t]\right] \leq & \sum_{i=1}^{K}\mathbb{E}\Bigg[\sum_{j\in\mathcal{N}_i[t]} R_{i,j}[t]f'\left(W_{i,j}[t]\right) \\ & - \sum_{j\in\mathcal{N}_i[t]} f\left(T_{i,j}[t]\right)C_i[t]S^*_{i,j}[t]\Bigg] + f(1)\sum_{i=1}^{K}\rho_i. \end{aligned} \tag{20}$$

Next, we consider the term $\mathbb{E}\left[\sum_{j\in\mathcal{N}_i[t]} R_{i,j}[t]f'(W_{i,j}[t])\right]$. Note that all files at time t arrived after time $t - W_i^{\max}[t] - 1$, and files arriving after time $t - T_i^{\max}[t]$ are not served between $t - T_i^{\max}[t] + 1$ and $t - 1$, where we recall that $W_i^{\max}[t] \triangleq \max_{j\in\mathcal{N}_i[t]} W_{i,j}[t]$ and $T_i^{\max}[t] \triangleq \max_{j\in\mathcal{N}_i[t]} T_{i,j}[t]$ denote the maximum age and maximum TSLS of files at link i in time slot t. This implies that

$$\begin{aligned}& \mathbb{E}\left[\sum_{j\in\mathcal{N}_i[t]} R_{i,j}[t]f'\left(W_{i,j}[t]\right)\middle| T_i^{\max}[t], W_i^{\max}[t]\right] \\ \overset{(a)}{\leq} & \mathbb{E}\Bigg[A_i^{\max}F_i^{\max}\sum_{\tau=t-W_i^{\max}[t]}^{t-T_i^{\max}[t]} f'(t-\tau) \\ & + \sum_{\tau=t-T_i^{\max}[t]+1}^{t-1} f'(t-\tau)\sum_{j=1}^{A_i[\tau]} F_{i,j}[\tau]\middle| T_i^{\max}[t], W_i^{\max}[t]\Bigg] \\ \overset{(b)}{=} & \rho_i\sum_{\tau=t-T_i^{\max}[t]+1}^{t-1} f'(t-\tau) + A_i^{\max}F_i^{\max}\sum_{\tau=t-W_i^{\max}[t]}^{t-T_i^{\max}[t]} f'(t-\tau) \\ = & \rho_i\sum_{m=1}^{T_i^{\max}[t]-1} f'(m) + A_i^{\max}F_i^{\max}\sum_{m=T_i^{\max}[t]}^{W_i^{\max}[t]} f'(m), \end{aligned} \tag{21}$$

where step (a) follows from the fact that files arriving after time $t - T_i^{\max}[t]$ are not served until time t and the fact that the number of incoming packets at each link i in each time slot is not greater than $A_i^{\max}F_i^{\max}$; (b) uses the fact that files arriving after time $t - T_i^{\max}[t]$ are independent from $W_i^{\max}[t]$ and $T_i^{\max}[t]$.

By using Lemma 3.4, inequality (21) becomes

$$\begin{aligned}& \mathbb{E}\left[\sum_{j\in\mathcal{N}_i[t]} R_{i,j}[t]f'\left(W_{i,j}[t]\right)\middle| T_i^{\max}[t], W_i^{\max}[t]\right] \\ \leq & \rho_i\left(f\left(T_i^{\max}[t]-1\right) - f(1) + f'(1)\right) \\ & + A_i^{\max}F_i^{\max}\left(f\left(W_i^{\max}[t]\right) - f\left(T_i^{\max}[t]\right) + f'(1)\right) \\ \leq & \rho_i f\left(T_i^{\max}[t]\right) + A_i^{\max}F_i^{\max}\left(f\left(W_i^{\max}[t]\right) - f\left(T_i^{\max}[t]\right)\right) \\ & + \left(\rho_i + A_i^{\max}F_i^{\max}\right)f'(1), \end{aligned} \tag{22}$$

which implies that

$$\begin{aligned}& \mathbb{E}\left[\sum_{j\in\mathcal{N}_i[t]} R_{i,j}[t]f'\left(W_{i,j}[t]\right)\right] \\ \leq & \rho_i\mathbb{E}\left[f\left(T_i^{\max}[t]\right)\right] + A_i^{\max}F_i^{\max}\mathbb{E}\left[f\left(W_i^{\max}[t]\right) - f\left(T_i^{\max}[t]\right)\right] \\ & + \left(\rho_i + A_i^{\max}F_i^{\max}\right)f'(1), \end{aligned} \tag{23}$$

By substituting inequality (23) into (20), we have

$$\begin{aligned}& \mathbb{E}\left[\Delta V[t]\right] \\ \overset{(a)}{\leq} & \sum_{i=1}^{K}\rho_i\mathbb{E}\left[f\left(T_i^{\max}[t]\right)\right] - \sum_{i=1}^{K}\mathbb{E}\left[\sum_{j\in\mathcal{N}_i[t]} f\left(T_{i,j}[t]\right)C_i[t]S^*_{i,j}[t]\right] + B_1 \\ & + \sum_{i=1}^{K} A_i^{\max}F_i^{\max}\mathbb{E}\left[f\left(W_i^{\max}[t]\right) - f\left(T_i^{\max}[t]\right)\right] \\ \overset{(b)}{=} & \sum_{i=1}^{K}\rho_i f\left(T_i^{\max}[t]\right) - \sum_{i=1}^{K}\mathbb{E}\left[f\left(T_i^{\max}[t]\right)C_i[t]S_i^*[t]\right] + B_1 \\ & + \sum_{i=1}^{K} A_i^{\max}F_i^{\max}\mathbb{E}\left[f\left(W_i^{\max}[t]\right) - f\left(T_i^{\max}[t]\right)\right], \end{aligned} \tag{24}$$

where the step (a) is true for $B_1 \triangleq f'(1)\sum_{i=1}^{K}\left(\rho_i + A_i^{\max}F_i^{\max}\right) + f(1)\sum_{i=1}^{K}\rho_i$; and (b) follows from the fact that the served file under the TSLS-based Algorithm should have the largest TSLS at each link, since at most one file can be scheduled at each link in each time slot.

By using Lemma 3.3, we have

$$\begin{aligned}\mathbb{E}\left[\Delta V[t]\right] \leq & \sum_{i=1}^{K}\rho_i\mathbb{E}\left[f\left(T_i^{\max}[t]\right)\right] - \sum_{i=1}^{K}\mathbb{E}\left[f\left(T_i^{\max}[t]\right)C_i[t]S_i^*[t]\right] \\ & + \sum_{i=1}^{K} A_i^{\max}F_i^{\max}\frac{\delta_i}{1-\delta_i}\mathbb{E}\left[f\left(T_i^{\max}[t]\right)\right] + B_2 \end{aligned} \tag{25}$$

where $B_2 \triangleq B_1 + \sum_{i=1}^{K} A_i^{\max}F_i^{\max}G_i$.

Note that the capacity region Λ (see [23]) is also equivalent to a set of traffic intensity vectors $\boldsymbol{\rho} = (\rho_i)_{i=1}^{K}$ such that there exist non-negative numbers $\alpha(\mathbf{s})$ satisfying

$$\rho_i \leq \sum_{\mathbf{c}}\phi_{\mathbf{c}}\sum_{\mathbf{s}\in\Omega}\alpha(\mathbf{s})s_ic_i, \qquad \forall i = 1, 2, .., K, \tag{26}$$

where both $\mathbf{c} = (c_i)_{i=1}^{K}$ and $\mathbf{s} = (s_i)_{i=1}^{K}$ are zero-one vectors, and $\sum_{\mathbf{s}\in\Omega}\alpha(\mathbf{s}) = 1$. For any traffic intensity vector $\boldsymbol{\rho} \in \text{Int}(\Lambda)$, we can pick $\delta_i > 0$, $\forall i = 1, 2, ..., K$, sufficiently small enough such that the

vector $\left(\rho_i + A_i^{\max} F_i^{\max} \frac{\delta_i}{1-\delta_i}\right)_{i=1}^K \in \text{Int}(\Lambda)$. Hence, there exists an $\epsilon > 0$ such that

$$\rho_i + A_i^{\max} F_i^{\max} \frac{\delta_i}{1-\delta_i} \leq \sum_{\mathbf{c}} \phi_{\mathbf{c}} \sum_{\mathbf{s}\in\Omega} \alpha(\mathbf{s}) s_i c_i - \epsilon, \qquad \forall i = 1, 2, .., K. \tag{27}$$

Thus, we have

$$\begin{aligned}\mathbb{E}\left[\Delta V[t]\right] \leq & -\epsilon \sum_{i=1}^K \mathbb{E}\left[f\left(T_i^{\max}[t]\right)\right] + B_2 \\ & + \sum_{i=1}^K \sum_{\mathbf{c}} \phi_{\mathbf{c}} \sum_{\mathbf{s}\in\Omega} \alpha(\mathbf{s}) s_i c_i \mathbb{E}\left[f\left(T_i^{\max}[t]\right)\right] \\ & - \sum_{i=1}^K \mathbb{E}\left[f\left(T_i^{\max}[t]\right) C_i[t] S_i^*[t]\right]. \end{aligned} \tag{28}$$

Given any $\mathbf{T}^{\max}[t]$, according to the TSLS-based Algorithm, we have

$$\begin{aligned}& \sum_{i=1}^K \sum_{\mathbf{c}} \phi_{\mathbf{c}} \sum_{\mathbf{s}\in\Omega} \alpha(\mathbf{s}) s_i c_i f\left(T_i^{\max}[t]\right) \\ =& \sum_{\mathbf{c}} \phi_{\mathbf{c}} \sum_{\mathbf{s}\in\Omega} \alpha(\mathbf{s}) \sum_{i=1}^K f\left(T_i^{\max}[t]\right) c_i s_i \\ \leq & \sum_{\mathbf{c}} \phi_{\mathbf{c}} \sum_{\mathbf{s}\in\Omega} \alpha(\mathbf{s}) \max_{\mathbf{s}\in\Omega} \sum_{i=1}^K f\left(T_i^{\max}[t]\right) c_i s_i \\ =& \sum_{\mathbf{c}} \phi_{\mathbf{c}} \max_{\mathbf{s}\in\Omega} \sum_{i=1}^K f\left(T_i^{\max}[t]\right) c_i s_i \\ =& \mathbb{E}\left[\max_{\mathbf{s}\in\Omega} \sum_{i=1}^K f\left(T_i^{\max}[t]\right) C_i[t] s_i\right] \\ =& \mathbb{E}\left[\sum_{i=1}^K f\left(T_i^{\max}[t]\right) C_i[t] S_i^*[t]\right], \end{aligned} \tag{29}$$

which implies

$$\sum_{i=1}^K \sum_{\mathbf{c}} \phi_{\mathbf{c}} \sum_{\mathbf{s}\in\Omega} \alpha(\mathbf{s}) s_i c_i f\left(T_i^{\max}[t]\right) \leq \sum_{i=1}^K \mathbb{E}\left[f\left(T_i^{\max}[t]\right) C_i[t] S_i^*[t]\right].$$

By substituting the above inequality into (28), we have

$$\begin{aligned}\mathbb{E}\left[\Delta V[t]\right] \leq & -\epsilon \sum_{i=1}^K \mathbb{E}\left[f\left(T_i^{\max}[t]\right)\right] + B_2 \\ \leq & -\epsilon \sum_{i=1}^K (1-\delta_i) \mathbb{E}\left[f\left(W_i^{\max}[t]\right)\right] + B, \end{aligned} \tag{30}$$

where the last step utilizes Lemma 3.3 and $B \triangleq \epsilon \sum_{i=1}^K G_i(1-\delta_i) + B_2 > 0$.

By summing the above inequality over $t = 0, 1, \ldots, M$, we have

$$\limsup_{M\to\infty} \frac{1}{M} \sum_{t=0}^{M-1} \sum_{i=1}^K \mathbb{E}\left[f(W_i^{\max}[t])\right] \leq \frac{B}{\gamma}, \tag{31}$$

where $\gamma \triangleq \epsilon(1-\delta_{\max}) > 0$ and $\delta_{\max} \triangleq \max_{i=1,2,\ldots,K} \delta_i$.

Since all files at time t arrived at the system after time $t - W_i^{\max}[t]$, we have

$$\sum_{j\in\mathcal{N}_i[t]} R_{i,j}[t] \leq A_i^{\max} F_i^{\max} W_i^{\max}[t], \forall i = 1, 2, \ldots, K, \tag{32}$$

which implies that for any $f \in \mathcal{G}$

$$\begin{aligned} f\left(\sum_{j\in\mathcal{N}_i[t]} R_{i,j}[t]\right) & \overset{(a)}{\leq} f\left(A_i^{\max} F_i^{\max} W_i^{\max}[t]\right) \\ & \overset{(b)}{\leq} f\left(W_i^{\max}[t]\right) + G_i' \\ & \overset{(c)}{\leq} f\left(W_i^{\max}[t]\right) + G_{\max}', \end{aligned} \tag{33}$$

where step (a) follows from that the fact that f is increasing; (b) follows the definition of $f \in \mathcal{G}$ and is true for some $G_i' > 0$; (c) is true for $G_{\max}' \triangleq \max_{i=1,2,\ldots,K} G_i' > 0$.

By substituting (33) into (31), we have

$$\limsup_{M\to\infty} \frac{1}{M} \sum_{t=0}^{M-1} \sum_{i=1}^K \mathbb{E}\left[f\left(\sum_{j\in\mathcal{N}_i[t]} R_{i,j}[t]\right)\right] \leq \frac{B}{\gamma} + K G_{\max}' < \infty.$$

This implies stability-in-the-mean property and thus the underlying Markov Chain is positive recurrent [12].

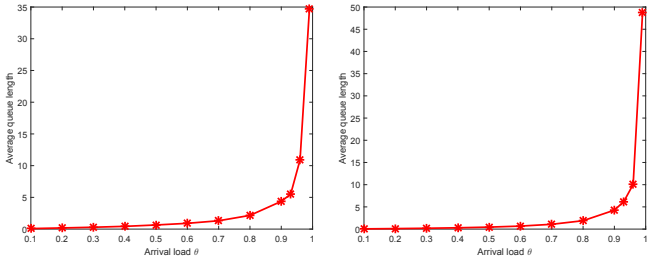
4 SIMULATIONS

In this section, we provide simulation results for our proposed TSLS-based algorithm with logarithmic function (i.e., $f(x) = \log(x+1)$) and compare its performance to both queue-length-based and age-based algorithms. In particular, we investigate the throughput (cf. Section 4.1), mean file delay and service regularity (cf. Section 4.2) performance of our policy in a fully-connected ON-OFF fading network with $K = 5$ links and a 3×3 switch. While the algorithm was described for wireless networks, it also applies to other communication networking applications where interference-type constraints exists, such as a switch. Since switches are widely used in data centers and the Internet, we demonstrate our algorithm in that context as well. The number of arriving files at each link in each time slot follows a Bernoulli distribution. The file size at link i has the following distribution: it is equal to M with probability $(\eta_i - 1)/(M-1)$ and 1 otherwise, where $M \geq 2$ is some parameter that measures the burstiness of the files and $\eta_i > 1$ is the mean file size at link i. Indeed, it is easy to calculate that the variance of file size at link i is equal to $(M - \eta_i)(\eta_i - 1)$, which linearly increases with the parameter M. In the fully-connected network, the probability vector of channel being ON is $\mathbf{p} = [0.1, 0.1, 0.9, 0.9, 0.9]$, mean file size vector is $\boldsymbol{\eta} = [5, 10, 20, 5, 15]$, and the traffic intensity at each link satisfies the following relationship: $\rho_1 = \rho_2 = \rho_3/3 = \rho_4/3 = \rho_5/3$. According to [25, Lemma 1], the capacity region can be represented as $\boldsymbol{\rho} = \theta \times [0.0908, 0.0908, 0.2724, 0.2724, 0.2724]$, where $\theta \in (0, 1)$ is called the arrival load. In the case of the 3×3 switch, the mean file size matrix is $\boldsymbol{\eta} = [5, 2, 2; 3, 10, 5; 5, 4, 5]$, and capacity region is $\boldsymbol{\rho} = \theta \times [0.7, 0.2, 0.1; 0.15, 0.6, 0.25; 0.15, 0.2, 0.65]$, where $\theta \in (0, 1)$.

4.1 Throughput Performance

Fig. 2 shows the average queue-length under our proposed TSLS-based algorithm in both fully-connected fading networks and 3×3

switch when M is equal to 50. It can be observed from Fig. 2 that the TSLS-based algorithm stabilizes the system for any $\theta \in (0, 1)$ in the above network setups, which validate the throughput optimality of our proposed algorithm.



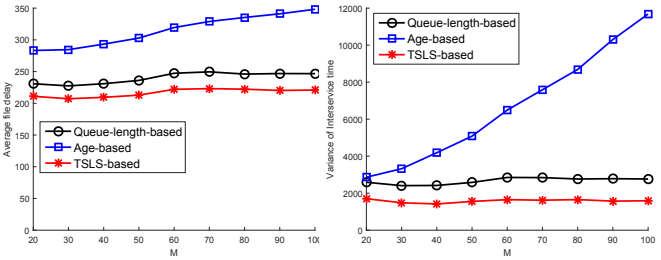
(a) Fully-connected network **(b) 3×3 switch**

Figure 2: The throughput of the TSLS-based Algorithm

4.2 Delay and Service Regularity Performance

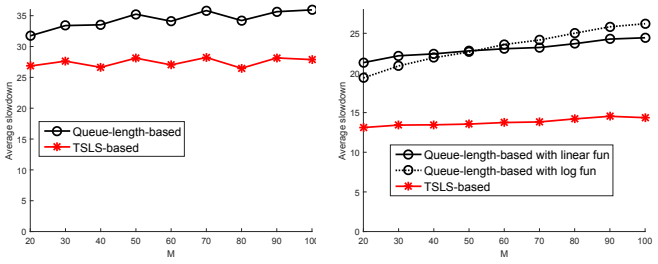
In this subsection, we investigate the delay and service regularity performance of our proposed TSLS-based Algorithm and compare it to both queue-length-based and age-based algorithms with both linear (i.e., $f(x) = x$) and logarithmic (i.e., $f(x) = \log(x + 1)$) functions. In particular, the queue-length-based algorithm means running MaxWeight across links based on the functions of number of files of each link and serving packets within each link in a round robin fashion. Similarly, the age-based algorithm schedules links via MaxWeight based on the functions of the maximum age of files at each link and serves packets in a round robin way at each link. Note that the MAC-layer queue-length-based algorithm proposed in [9] corresponds to the queue-length-based algorithm with the logarithmic function.

Fig. 3 shows the mean file delay and service regularity performance of our proposed TSLS-based algorithm as well as both queue-length-based and age-based algorithms with respect to the parameter M in a fully-connected network with ON-OFF fading channel when the arrival load is $\theta = 0.9$. Note that in fully connected networks, both queue-length-based and age-based algorithms are insensitive to the functional forms. In other words, both queue-length-based and age-based algorithms with any increasing functions correspond to the queue-length-based and age-based algorithms with the linear function, respectively. We can observe from Fig. 3 that both mean file delay and service regularity linearly increase with the parameter M under the age-based algorithm, while they almost keep the same under both queue-length-based and TSLS-based algorithms. Recall that the larger the M, the higher variance of file size. This indicates that the network performance under the age-based algorithm is quite sensitive to the variance of the file size, while they are independent of variance of the file size under both queue-length-based and TSLS-based algorithms. Although age-based algorithm serves packets in a round robin fashion, it schedules links based on the First-Come-First-Serve discipline and thus the different distributions of file size across links still have significant impact on its performance. In contrast, both queue-length-based and TSLS-based algorithms exhibit an insensitivity property in the sense that both mean file delay and service regularity under these two algorithms are independent of network traffic characteristics except the mean traffic intensity. However, by closely looking at the slowdown[1] performance in Fig. 4a, we can observe that the average slowdown under our proposed TSLS-based algorithm almost stays the same as the parameter M varies, while it slightly increases under the queue-length-based policies. This clearly shows the delay insensitivity property of our proposed algorithm. In addition to that, our proposed TSLS-based algorithm outperforms the queue-length-based algorithm in terms of both mean delay and service regularity performance and exhibits significant gains in service regularity performance. This is because that our proposed algorithm serves files in a round robin fashion not only within each link but also across the links. We can observe the similar phenomenon from Fig. 5 and Fig. 4b for the 3×3 switch.



(a) Average delay performance **(b) Service regularity performance**

Figure 3: Fully-connected network: arrival load $\theta = 0.9$



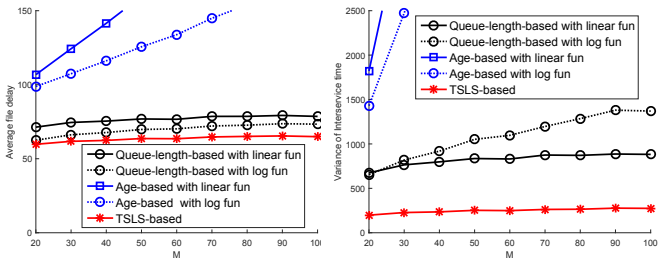
(a) Fully-connected network **(b) 3×3 switch**

Figure 4: Average slowdown: arrival load $\theta = 0.9$

Since our TSLS-based algorithm outperforms queue-length-based algorithm for a fixed arrival load $\theta = 0.9$, it is also useful to know if this observation holds for all traffic loads. To that end, we further compare their performance in terms of traffic load when M is fixed to 60. We can observe from Fig. 6 that the TSLS-based algorithm always performs better than the queue-length-based algorithm in terms of both mean file delay and service regularity performance

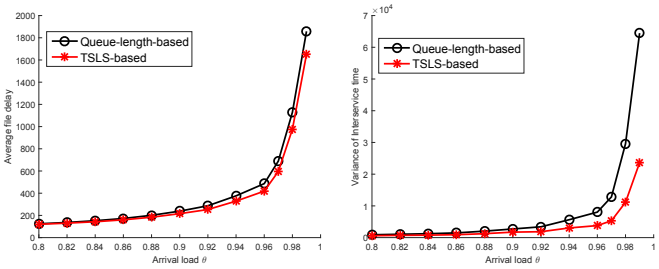
[1]The slowdown of a file is its file delay divided by its size. If the mean slowdown is a constant independent of network traffic characteristic other than its mean traffic intensity, then the mean file delay is said to be insensitive.

in fully-connected networks, and shows large performance gains in service regularity. Interestingly, such a performance improvement increases as the arrival load increases. Similar observations can be made in the 3×3 switch, as shown in Fig. 7.



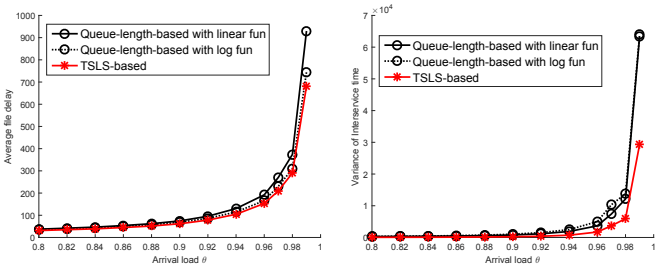
(a) Average delay performance **(b) Service regularity performance**

Figure 5: 3×3 switch: arrival load $\theta = 0.9$



(a) Average delay performance **(b) Service regularity performance**

Figure 6: Fully-connected network: $M = 60$



(a) Average delay performance **(b) Service regularity performance**

Figure 7: 3×3 switch: $M = 60$

5 CONCLUSION

In this work, we developed a round-robin-like algorithm that has attractive throughput, mean delay and service regularity performance as the round robin algorithm in wireline networks. We maintained a time-since-last-service (TSLS) counter for each file, and established a tight relationship between the mean TSLS and the second moment of inter-service time in the presence of flow dynamics. Based on this, we proposed a TSLS-based scheduling algorithm that nicely balances the weight between the TSLS and the channel rate. We established the throughput optimality property of our proposed algorithm, and showed its excellent delay and service regularity in comparison to various other alternatives through extensive simulations.

ACKNOWLEDGMENTS

This work is supported by the NSF grants: CCSS-EARS-1444026, CNS-NeTS-1514127, CNS-NeTS-1717045, CMMI-SMOR-1562065, CNS-WiFiUS-1456806, CNS-ICN-WEN-1719371, CIF-1409106, and ECCS 16-09370; the DTRA grant HDTRA1-15-1-0003; the QNRF Grant NPRP 7-923-2-344; the ARO grants: MURI W911NF-12-1-0385 and W911NF-16-1-0259.

REFERENCES

[1] M. Andrews, K. Kumaran, K. Ramanan, A. Stolyar, R. Vijayakumar, and P. Whiting. 2004. Scheduling in a queueing system with asynchronously varying service rates. *Probability in the Engineering and Informational Sciences archive* 18, 2 (April 2004), 191–217.

[2] T. Bonald, S. Borst, and A. Proutière. 2004. How mobility impacts the flow-level performance of wireless data systems. In *Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*. Hong Kong, China.

[3] T. Bonald and M. Feuillet. 2010. On the Stability of Flow-Aware CSMA. *Performance Evaluation* 67, 11 (November 2010), 1219–1229.

[4] S. Borst. 2005. User-level performance of channel-aware scheduling algorithms in wireless data networks. *IEEE/ACM Transactions on Networking (TON)* 13, 3 (2005), 636–647.

[5] S. Borst. 2008. Flow-level performance and user mobility in wireless data networks. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences* 366, 1872 (2008), 2047–2058.

[6] A. Demers, S. Keshav, and S. Shenker. 1989. Analysis and simulation of a fair queueing algorithm. In *ACM SIGCOMM Computer Communication Review*, Vol. 19. 1–12.

[7] A. Eryilmaz and R. Srikant. 2012. Asymptotically Tight Steady-state Queue Length Bounds Implied By Drift Conditions. *Queueing Systems* 72 (2012), 311–359.

[8] R. Gallager. 2013. *Stochastic processes: theory for applications.* Cambridge University Press.

[9] J. Ghaderi, T. Ji, and R. Srikant. 2014. Flow-level stability of wireless networks: Separation of congestion control and scheduling. *IEEE Trans. Automat. Control* 59, 8 (2014), 2052–2067.

[10] M. Harchol-Balter. 2013. *Performance Modeling and Design of Computer Systems: Queueing Theory in Action.* Cambridge University Press.

[11] A. Jalali, R. Padovani, and R. Pankaj. 2000. Data throughput of CDMA-HDR a high efficiency-high data rate personal communication wireless system. In *IEEE Vehicular technology conference proceedings.*

[12] P.R. Kumar and S.P. Meyn. 1995. Stability of Queueing Networks and Scheduling Policies. *IEEE Trans. Automat. Control* 40 (1995), 251–260.

[13] B. Li and A. Eryilmaz. 2013. Optimal Distributed Scheduling under Time-varying Conditions: A Fast-CSMA Algorithm with Applications. *IEEE Transactions on Wireless Communications* 12, 7 (2013), 3278–3288.

[14] B. Li, A. Eryilmaz, and R. Srikant. 2015. On the universality of age-based scheduling in wireless networks. In *IEEE Conference on Computer Communications (INFOCOM)*. IEEE, 1302–1310.

[15] B. Li, R. Li, and A. Eryilmaz. 2015. Throughput-optimal scheduling design with regular service guarantees in wireless networks. *IEEE/ACM Transactions on Networking (ToN)* 23, 5 (2015), 1542–1552.

[16] B. Li, R. Li, and A. Eryilmaz. 2016. Wireless Scheduling Design for Optimizing Both Service Regularity and Mean Delay in Heavy-Traffic Regimes. *IEEE/ACM Transactions on Networking (ToN)* 24, 3 (2016), 1867–1880.

[17] S. Lu, V. Bharghavan, and R. Srikant. 1999. Fair Scheduling in Wireless Packet Networks. *IEEE/ACM Transactions on Networking* 7, 4 (August 1999), 473–489.

[18] N. McKeown, A. Mekkittikul, V. Anantharam, and J. Walrand. 1999. Achieving 100% throughput in an input-queued switch. *IEEE Transactions on Communications* 47, 8 (August 1999), 1260 – 1267.

[19] N.Walton. 2015. Concave Switching in Single and Multihop networks. *Queueing Systems* 81, 2 (2015), 265–299.

[20] I. Rhee, A. Warrier, J. Min, and L. Xu. 2009. DRAND: Distributed Randomized TDMA Scheduling for Wireless Ad Hoc Networks. *IEEE Transactions on Mobile Computing* 8, 10 (2009), 1384–1396.
[21] S. Shakkottai and A. Stolyar. 2002. Scheduling for multiple flows sharing a time-varying channel: The exponential rule. *American Mathematical Society Translations, Series 2* 207 (2002), 185–202.
[22] A. Stolyar. 2004. Maxweight Scheduling in a Generalized Switch: State Space Collapse and Workload Minimization in Heavy Traffic. *The Annals of Applied Probability* 14, 1 (2004), 1–53.
[23] L. Tassiulas. 1997. Scheduling and Performance Limits of Networks with Constantly Varying Topology. *IEEE Transactions on Information Theory* 43 (1997), 1067–1073.
[24] L. Tassiulas and A. Ephremides. 1992. Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. *IEEE Trans. Automat. Control* 36, 12 (1992), 1936–1948.
[25] L. Tassiulas and A. Ephremides. 1993. Dynamic server allocation to parallel queues with randomly varying connectivity. *IEEE Transactions on Information Theory* 39 (1993), 466–478.
[26] L. Tassiulas and S. Sarkar. 2002. Maxmin fair scheduling in wireless networks. In *Twenty-first Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*. New York City, New York, USA.
[27] N. Walton. 2014. Store-forward and its implications for proportional scheduling. In *Communication, Control, and Computing (Allerton), 2014 52nd Annual Allerton Conference on*. IEEE, 1174–1181.

A PROOF OF LEMMA 3.1

Let $s_{i,j}$ and $d_{i,j}$ be the arrival and departure time of file j at link i, respectively. We use $m_{i,j}$ to denote the number of times that file j at link i has been served by time t. Recall that $I_{i,j}[k]$ denotes the inter-service time between $(k-1)^{th}$ and k^{th} service of file j at link i. Then, we have

$$\sum_{t=s_{i,j}}^{\min\{t,d_{i,j}\}} T_{i,j}[t] = \sum_{t=s_{i,j}+1}^{s_{i,j}+I_{i,j}[1]} T_{i,j}[t] + \sum_{t=s_{i,j}+I_{i,j}[1]+1}^{s_{i,j}+I_{i,j}[1]+I_{i,j}[2]} T_{i,j}[t] + \ldots + \sum_{t=s_{i,j}+I_{i,j}[1]+\ldots+I_{i,j}[m_{i,j}]+1}^{\min\{t,d_{i,j}\}} T_{i,j}[t] \quad (34)$$

We observe the following fact: assume file j at link i receives its $(k-1)^{th}$ and k^{th} service at time slot t_1 and t_2, respectively, where $t_2 > t_1$. Then, by definition, $I_{i,j}[k] = t_2 - t_1$ and $T_{i,j}[t] = t - t_1 - 1$, $\forall t_1 < t \le t_2$. Using this fact, we know the k^{th} summation on the right hand side of (34) gives $I_{i,j}[k](I_{i,j}[k]-1)/2$, except for the last one. Therefore, we have

$$\sum_{k=1}^{m_{i,j}} \frac{I_{i,j}[k](I_{i,j}[k]-1)}{2} \le \sum_{t=s_{i,j}}^{\min\{t,d_{i,j}\}} T_{i,j}[t] \le \sum_{k=1}^{m_{i,j}+\mathbb{1}_{\{d_{i,j}>t\}}} \frac{I_{i,j}[k](I_{i,j}[k]-1)}{2}. \quad (35)$$

By summing all the files coming before time t, we have

$$\begin{aligned} &\sum_{j:s_{i,j}\le t} \sum_{k=1}^{m_{i,j}} \frac{I_{i,j}[k](I_{i,j}[k]-1)}{2} \\ \le &\sum_{j:s_{i,j}\le t} \sum_{t=s_{i,j}}^{\min\{t,d_{i,j}\}} T_{i,j}[t] \\ \le &\sum_{j:s_{i,j}\le t} \sum_{k=1}^{m_{i,j}+\mathbb{1}_{\{d_{i,j}>t\}}} \frac{I_{i,j}[k](I_{i,j}[k]-1)}{2}, \end{aligned} \quad (36)$$

This combines with the following fact that

$$\sum_{k=1}^{m_{i,j}} I_{i,j}[k] \le \min\{t, d_{i,j}\} - s_{i,j} + 1 \le \sum_{k=1}^{m_{i,j}+\mathbb{1}_{\{d_{i,j}>t\}}} I_{i,j}[k],$$

and implies that

$$\begin{aligned} &\frac{\sum_{j:s_{i,j}\le t} \sum_{k=1}^{m_{i,j}} \frac{I_{i,j}[k](I_{i,j}[k]-1)}{2}}{\sum_{j:s_{i,j}\le t} \sum_{k=1}^{m_{i,j}+\mathbb{1}_{\{d_{i,j}>t\}}} I_{i,j}[k]} \\ \le &\frac{\sum_{j:s_{i,j}\le t} \sum_{t=s_{i,j}}^{\min\{t,d_{i,j}\}} T_{i,j}[t]}{\sum_{j:s_{i,j}\le t} \left(\min\{t,d_{i,j}\} - s_{i,j} + 1\right)} \\ \le &\frac{\sum_{j:s_{i,j}\le t} \sum_{k=1}^{m_{i,j}+\mathbb{1}_{\{d_{i,j}>t\}}} \frac{I_{i,j}[k](I_{i,j}[k]-1)}{2}}{\sum_{j:s_{i,j}\le t} \sum_{k=1}^{m_{i,j}} I_{i,j}[k]}. \end{aligned} \quad (37)$$

Note that

$$\begin{aligned} &\lim_{t\to\infty} \frac{\sum_{j:s_{i,j}\le t} \sum_{k=1}^{m_{i,j}} \frac{I_{i,j}[k](I_{i,j}[k]-1)}{2}}{\sum_{j:s_{i,j}\le t} \sum_{k=1}^{m_{i,j}+\mathbb{1}_{\{d_{i,j}>t\}}} I_{i,j}[k]} \\ = &\lim_{t\to\infty} \frac{\sum_{j:s_{i,j}\le t} m_{i,j}}{\sum_{j:s_{i,j}\le t} \left(m_{i,j} + \mathbb{1}_{\{d_{i,j}>t\}}\right)} \\ &\cdot \frac{\frac{1}{\sum_{j:s_{i,j}\le t} m_{i,j}} \sum_{j:s_{i,j}\le t} \sum_{k=1}^{m_{i,j}} \frac{I_{i,j}[k](I_{i,j}[k]-1)}{2}}{\frac{1}{\sum_{j:s_{i,j}\le t}(m_{i,j}+\mathbb{1}_{\{d_{i,j}>t\}})} \sum_{j:s_{i,j}\le t} \sum_{k=1}^{m_{i,j}+\mathbb{1}_{\{d_{i,j}>t\}}} I_{i,j}[k]} \\ = &\frac{\mathbb{E}\left[\bar{I}_i\left(\bar{I}_i - 1\right)\right]}{2\mathbb{E}\left[\bar{I}_i\right]}, \end{aligned}$$

where the second last step uses the fact that $\sum_{j:s_{i,j}\le t} m_{i,j} \to \infty$ as $t \to \infty$ and $\sum_{j:s_{i,j}\le t} \mathbb{1}_{\{d_{i,j}>t\}} < \infty$ hold for almost all sample paths.

Similarly, we can show that

$$\lim_{t\to\infty} \frac{\sum_{j:s_{i,j}\le t} \sum_{k=1}^{m_{i,j}+\mathbb{1}_{\{d_{i,j}>t\}}} \frac{I_{i,j}[k](I_{i,j}[k]-1)}{2}}{\sum_{j:s_{i,j}\le t} \sum_{k=1}^{m_{i,j}} I_{i,j}[k]} = \frac{\mathbb{E}\left[\bar{I}_i\left(\bar{I}_i - 1\right)\right]}{2\mathbb{E}\left[\bar{I}_i\right]}.$$

On the other hand, we have

$$\lim_{t\to\infty} \frac{\sum_{j:s_{i,j}\le t} \sum_{t=s_{i,j}}^{\min\{t,d_{i,j}\}} T_{i,j}[t]}{\sum_{j:s_{i,j}\le t} \left(\min\{t,d_{i,j}\} - s_{i,j} + 1\right)} = \mathbb{E}\left[\overline{T}_i\right].$$

Thus, we have the desired result.