

Throughput-Optimal Opportunistic Scheduling in the Presence of Flow-Level Dynamics

Shihuan Liu
ECE

Iowa State University
liush08@iastate.edu;

Lei Ying
ECE

Iowa State University
leiying@iastate.edu

R. Srikant
ECE and CSL

University of Illinois
rsrikant@illinois.edu

Abstract—We consider multiuser scheduling in wireless networks with channel variations and flow-level dynamics. Recently, it has been shown that the MaxWeight algorithm, which is throughput-optimal in networks with a fixed number of users, fails to achieve the maximum throughput in the presence of flow-level dynamics. In this paper, we propose a new algorithm, called *workload-based scheduling with learning*, which is provably throughput-optimal, requires no prior knowledge of channels and user demands, and performs significantly better than previously suggested algorithms.

I. INTRODUCTION

Multiuser scheduling is one of the core challenges in wireless communications. Due to channel fading and wireless interference, scheduling algorithms need to dynamically allocate resources based on both the demands of the users and the channel states to maximize network throughput. The celebrated MaxWeight algorithm developed in [2] for exploiting channel variations works as follows. Consider a network with a single base station and n users, and further assume that the base station can transmit to only one user in each time slot. The MaxWeight algorithm computes the product of the queue length and current channel rate for each user, and chooses to transmit to that user which has the largest product; ties can be broken arbitrarily. The throughput-optimality property of the MaxWeight algorithm was first established in [2], and the results were later extended to more general channel and arrival models in [3]–[5]. The MaxWeight algorithm should be contrasted with other opportunistic scheduling such as [6], [7] which exploit channel variations to allocate resources fairly assuming continuously backlogged users, but which are not throughput-optimal when the users are not continuously backlogged.

While the results in [2]–[4] demonstrate the power of MaxWeight-based algorithms, they were obtained under the assumptions that *the number of users in the network is fixed and the traffic flow generated by each user is long-lived, i.e., each user continually injects new bits into the network*. However, practical networks have flow-level dynamics: users arrive to transmit data and leave the network after the data are fully transmitted. In a recent paper [1], the authors showed that the MaxWeight algorithm is in fact *not throughput optimal* in networks with flow-level dynamics by providing a clever example showing the instability of the MaxWeight scheduling. The intuition is as follows: if a long-lived flow does not

receive enough service, its backlog builds up, which forces the MaxWeight scheduler to allocate more service to the flow. This interaction between user backlogs and scheduling guarantees the correctness of the resource allocation. However, if a flow has only a finite number of bits, its backlog does not build up over time and it is possible for the MaxWeight to stop serving such a flow and thus, the flow may stay in the network forever. Thus, in a network where finite-size flows continue to arrive, the number of flows in the network could increase to infinity. One may wonder why flow-level instability is important since, in real networks, base stations limit the number of simultaneously active flows in the network by rejecting new flows when the number of existing flows reaches a threshold. The reason is that, if a network model without such upper limits is unstable in the sense that the number of flows grows unbounded, then the corresponding real network with an upper limit on the number of flows will experience high flow blocking rates. This fact is demonstrated in our simulations later.

In [1], the authors addressed this instability issue of MaxWeight-based algorithms, and established necessary and sufficient conditions for the stability of networks with flow-level dynamics. The authors also proposed throughput-optimal scheduling algorithms. However, as the authors mentioned in [1], the proposed algorithms require prior knowledge of channel distribution and traffic distribution, which is difficult and sometimes impossible to obtain in practical systems, and further, the performance of the proposed algorithms is also not ideal.

Since flow arrivals and departures are common in reality, we are interested in developing practical scheduling algorithms that are throughput-optimal *under flow-level dynamics*. We consider a wireless system with a single base station and multiple users (flows). The network contains both long-lived flows, which keep injecting bits into the network, and short-lived flows, which have a finite number of bits to transmit. The main contributions of this paper include the following:

- We obtain the necessary conditions for flow-level stability of networks with both long-lived flows and short-lived flows. This generalizes the result in [1], where only short-lived flows are considered.
- We propose a simple algorithm for networks with short-lived flows only. Under this algorithm, each flow keeps track of the best channel condition that it has seen so far.

Each flow whose current channel condition is equal to the best channel condition that it has seen during its lifetime is eligible for transmission. It is shown that an algorithm which uniformly and randomly chooses a flow from this set of eligible flows for transmission is throughput-optimal. Note that the algorithm is a purely opportunistic algorithm in that it selects users for transmission when they are in the best channel state that they have seen so far, without considering their backlogs.

- Based on an optimization framework, we propose to use the estimated *workload*, the number of time slots required to transmit the remainder of a flow based on the best channel condition seen by the flow so far, to measure the backlog of short-lived flows. By comparing this short-lived flow backlog to the queue lengths and channel conditions of the long-lived flows, we develop a new algorithm, named workload-based scheduling with learning, which is throughput-optimal under flow-level dynamics. The term "learning" refers to the fact that the algorithm learns the best channel condition for each short-lived flow and attempts to transmit when the channel condition is the best.
- We use simulations to evaluate the performance of the proposed scheduling algorithm, and observe that the workload-based scheduling with learning performs significantly better than the MaxWeight scheduling in various settings.

The terminology of long-lived and short-lived flows above has to be interpreted carefully in practical situations. In practice, each flow has a finite size and thus, all flows eventually will leave the system if they receive sufficient service. Thus, all flows are short-lived flows in reality. Our results suggest that transmitting to users who are individually in their best estimated channel state so far is thus, throughput optimal. On the other hand, it is also well known that real network traffic consists of many flows with only a few packets and a few flows with a huge number of packets. If one considers the time scales required to serve the small-sized flows, the large-sized flows will appear to be long-lived (i.e., persistent forever) in the terminology above. Thus, if one is interested in performance over short time-scales, an algorithm which considers flows with a very large number of packets as being long-lived may lead to better performance and hence, we consider the more general model which consists of both short-lived flows and long-lived flows. Our simulations in [12] confirm the fact that the algorithm which treats some flows as being long-lived leads to better performance although through-optimality does not require such a model. In addition, long-lived flows partially capture the scenario where all bits from a flow do not arrive at the base station all at once. This fact is also exploited in our simulation experiments.

II. BASIC MODEL

Network Model: We consider a discrete-time wireless downlink network with a single base station and many flows,

each flow associates with a distinct mobile user. The base station can serve only one flow at a time.

Traffic Model: The network consists of the following two types of flows:

- **Long-lived flows:** Long-lived flows are traffic streams that are always in the network and continually generate bits to be transmitted.
- **Short-lived flows:** Short-lived flows are flows that have a finite number of bits to transmit. A short-lived flow enters the network at a certain time, and leaves the system after all bits are transmitted.

We assume that the set of long-lived flows is fixed, and short-lived flows arrive and depart. We let l be the index for long-lived flows, \mathcal{L} be the set of long-lived flows, and L be the number of long-lived flows, i.e., $L = |\mathcal{L}|$. Furthermore, we let $X_l(t)$ be the number of new bits injected by long-lived flow l in time slot t , where $X_l(t)$ is a discrete random variable with finite support, and independently and identically distributed (i.i.d.) across time slots. We also assume $\mathbf{E}[X_l(t)] = x_l$ and $X_l(t) \leq X^{\max}$ for all l and t .

Similarly, we let i be the index for short-lived flows, $\mathcal{I}(t)$ be the set of short-lived flows in the network at time t , and $I(t)$ be the number of short-lived flows at time t , i.e., $I(t) = |\mathcal{I}(t)|$. We denote by f_i the size (total number of bits) of short-lived flow i , and assume $f_i \leq F^{\max}$ for all i .

It is important to note that we allow different short-lived flows to have different maximum link rates. A careful consideration of our proofs will show the reader that the learning algorithm is not necessary if all users have the same maximum rate and that one can simply transmit to the user with the best channel state if it is assumed that all users have the same maximum rate. However, we do not believe that this is a very realistic scenario since SNR variations will dictate different maximum rates for different users.

Residual Size and Queue Length: For a short-lived flow i , let $Q_i(t)$, which we call the residual size, denote the number of bits still remaining in the system at time t . For a long-lived flow l , let $Q_l(t)$ denote the number of bits stored at the queue at the base station.

Channel Model: There is a wireless link between each user and the base station. Denote by $R_i(t)$ the state of the link between short-lived flow i and the base station at time t (i.e., the maximum rate at which the base station can transmit to short-lived flow i at time t), and $R_l(t)$ the state of the link between long-lived flow l and the base station at time t . We assume that $R_i(t)$ and $R_l(t)$ are discrete random variables with finite support. Define R_i^{\max} and R_l^{\max} to be the largest values that these random variables can take, i.e., $\Pr(R_j(t) > R_j^{\max}) = 0$ for each $j \in \mathcal{L} \cup (\bigcup_t \mathcal{I}(t))$. Further, assume that there exist $p_s^{\max} > 0$ and $R^{\max} > 0$ such that

$$\Pr(R_i(t) = R_i^{\max}) \geq p_s^{\max} \quad \forall i, t$$

$$\max \{ \max_i R_i^{\max}, \max_l R_l^{\max} \} \leq R^{\max}.$$

The states of wireless links are assumed to be independent across flows and time slots (but not necessarily identically

distributed across flows). The independence assumption across time slots can be relaxed easily but at the cost of more complicated proofs.

III. WORKLOAD-BASED SCHEDULING WITH LEARNING

In this section, we introduce a new scheduling algorithm called Workload-based Scheduling with Learning (WSL).

Workload-based Scheduling with Learning: For a short-lived flow i , we define

$$\tilde{R}_i^{\max}(t) = \max_{\max\{t-D, b_i\} \leq s \leq t} R_i(s),$$

where b_i is the time short-lived flow i joins the network and $D > 0$ is called the learning period. A key component of this algorithm is to use \tilde{R}_i^{\max} to evaluate the workload of short-lived flows (the reason will be explained in a detail in Section V). However, R_i^{\max} is in general unknown, so the scheduling algorithm uses $\tilde{R}_i^{\max}(t)$ as an estimate of R_i^{\max} .

During each time slot, the base station first checks the following inequality:

$$\alpha \sum_{i \in \mathcal{I}(t)} \left[\frac{Q_i(t)}{\tilde{R}_i^{\max}(t)} \right] > \max_{l \in \mathcal{L}} Q_l(t) R_l(t), \quad (1)$$

where $\alpha > 0$.

- If inequality (1) holds, then the base station serves a short-lived flow as follows: if at least one short-lived flow (say flow i) satisfies $R_i(t) \geq Q_i(t)$ or $R_i(t) = \tilde{R}_i^{\max}(t)$, then the base station uniformly and randomly selects such a flow for transmission¹; otherwise, the base station picks an arbitrary short-lived flow to serve.
- If inequality (1) does not hold, then the base station serves a long-lived flow l^* such that

$$l^* \in \arg \max_{l \in \mathcal{L}} Q_l(t) R_l(t)$$

(ties are broken arbitrarily). \square

Remark 1: The α in inequality (1) is a parameter balancing the performance of long-lived flows and short-lived flows. A large α will lead to a small number of short-lived flows but large queue-lengths of long-lived flows, and vice versa.

Remark 2: In Theorem 3, we will prove that WSL is throughput optimal when D is sufficiently large. From purely throughput-optimality considerations, it is then natural to choose $D = \infty$. However, in practical systems, if we choose D too large, such as ∞ , then it is possible that a flow may stay in the system for a very long time if its best channel condition occurs extremely rarely. Thus, it is perhaps best to choose a finite D to tradeoff between performance and throughput. This fact is explored in simulations in [12].

¹Other tie-breaking rules also may work. In fact, any tie-breaking rule which leads to Proposition 4 (in the appendix) being satisfied will work. For example, breaking ties in favor of flows which have been in the network the longest also works. In simulations, when the traffic load is close to the capacity limit, this oldest-first rule performs significantly better than the uniformly random rule stated above [12].

Remark 3: If all flows are short-lived, then the algorithm simplifies as follows: If at least one short-lived flow (say flow i) satisfies $R_i(t) \geq Q_i(t)$ or $R_i(t) = \tilde{R}_i^{\max}(t)$, then the base station uniformly and randomly selects such a flow for transmission; otherwise, the base station picks an arbitrary short-lived flow to serve. Simply stated, the algorithm serves one of the flows which can be completely transmitted or sees its best channel state, where the best channel state is an estimate based on past observations. If no such flow exists, any flow can be served. We do not separately prove the throughput optimality of this scenario since it is a special case of the scenario considered here. But it is useful to note that, in the case of short-lived flows only, the algorithm does not consider backlogs at all in making scheduling decisions.

We will prove that WSL (with any $\alpha > 0$) is throughput-optimal in the following sections, i.e., the scheduling policy can support any set of traffic flows that are supportable by any other algorithm. In the next section, we first present the necessary conditions for the stability, which also define the network throughput region.

IV. NECESSARY CONDITIONS FOR STABILITY

In this section, we establish the necessary conditions for the stability of networks with flow-level dynamics. To get the necessary conditions, we need to classify the short-lived flows into different classes:

- A short-lived flow class is defined by a pair of random variables (\hat{R}, \hat{F}) . Class- k is associated with random variables \hat{R}_k and \hat{F}_k .² A short-lived flow i belongs to class k if $R_i(t)$ has the same distribution as \hat{R}_k and the size of flow i (f_i) has the same distribution as \hat{F}_k . We let $\Lambda_k(t)$ denote the number of class- k flows joining the network at time t , where $\Lambda_k(t)$ are i.i.d. across time slots and $\mathbf{E}[\Lambda_k(t)] = \lambda_k$. Denote by \mathcal{K} the set of distinct classes. We assume that \mathcal{K} is finite, $|\mathcal{K}| = K$, and $\Lambda_k[t] \leq \lambda^{\max}$ for all t and $k \in \mathcal{K}$.
- Let \mathbf{c} denote an L -dimensional vector describing the state of the channels of the long-lived flows. In state \mathbf{c} , $R_{\mathbf{c},l}$ is the service rate that long-lived flow l can receive if it is scheduled. We denote by \mathcal{C} the set of all possible states.
- Let $\mathbf{C}(t)$ denote the state of the long-lived flows at time t , and $\pi_{\mathbf{c}}$ denote the probability that $\mathbf{C}(t)$ is in state \mathbf{c} .
- Let $p_{\mathbf{c},l}$ be the probability that the base station serves flow l when the network is in state \mathbf{c} . Clearly, for any \mathbf{c} , we have $\sum_{l \in \mathcal{L}} p_{\mathbf{c},l} \leq 1$. Note that the sum could be less than 1 if the base station schedules a short-lived flow in this state.
- Let $\mu_{\mathbf{c},s}$ be the probability that the base station serves a short-lived flow when the network is in state \mathbf{c} .
- Let $\Theta_{k,\beta}(t)$ denote the number of short-lived flows that belong to class- k and have residual size $Q(t) = \beta$. Note that β can only take on a finite number of values.

Theorem 1: Consider traffic parameters $\{x_l\}$ and $\{\lambda_k\}$, and

²We use $\hat{\cdot}$ to indicate that the notation is associated with a class of short-lived flows instead of an individual short-lived flow.

suppose that there exists a scheduling policy guaranteeing

$$\lim_{t \rightarrow \infty} \mathbf{E} \left[\sum_{l \in \mathcal{L}} Q_l(t) + \sum_{k \in \mathcal{K}} \sum_{\beta=1}^{F^{\max}} \Theta_{k,\beta}(t) \right] < \infty.$$

Then there exist $p_{c,l}$ and $\mu_{c,s}$ such that the following inequalities hold:

$$x_l \leq \sum_{c \in \mathcal{C}} \pi_c p_{c,l} R_{c,l} \quad \forall l \in \mathcal{L} \quad (2)$$

$$\sum_{k \in \mathcal{K}} \lambda_k \mathbf{E} \left[\left[\frac{\hat{F}_k}{\hat{R}_k^{\max}} \right] \right] \leq \sum_{c \in \mathcal{C}} \mu_{c,s} \pi_c. \quad (3)$$

$$\left(\sum_{l \in \mathcal{L}} p_{c,l} \right) + \mu_{c,s} \leq 1 \quad \forall c \in \mathcal{C}. \quad (4)$$

Inequality (2) and (3) state that the service allocated should be no less than the user requests if the flows are supportable. Inequality (4) states that the overall time used to serve long-lived and short-lived flows should be no more than the time available. The complete proof is based on the Strict Separation Theorem and is along the lines of a similar proof in [4].

V. THROUGHPUT OPTIMALITY OF WSL

First, we provide some intuition into how one can derive the WSL algorithm from optimization decomposition considerations. Then, we will present our main throughput optimality results. Given traffic parameters $\{x_l\}$ and $\{\lambda_k\}$, the necessary conditions for the supportability of the traffic is equivalent to the feasibility of the following constraints:

$$\begin{aligned} x_l &\leq \sum_{c \in \mathcal{C}} \pi_c p_{c,l} R_{c,l} && \forall l \\ \sum_{k \in \mathcal{K}} \lambda_k \mathbf{E} \left[\left[\frac{\hat{F}_k}{\hat{R}_k^{\max}} \right] \right] &\leq \sum_{c \in \mathcal{C}} \mu_{c,s} \pi_c && (5) \\ \sum_{l \in \mathcal{L}} p_{c,l} + \mu_{c,s} &\leq 1 && \forall c. \end{aligned}$$

For convenience, we view the feasibility problem as an optimization problem with the objective $\max A$, where A is some constant. While we have not explicitly stated that the x 's and μ 's are non-negative, this is assumed throughout.

Partially augmenting the objective using Lagrange multipliers, we get

$$\begin{aligned} &\max A - \sum_{l \in \mathcal{L}} q_l (x_l - \sum_{c \in \mathcal{C}} \pi_c p_{c,l} R_{c,l}) - \\ & q_s \left(\sum_{k \in \mathcal{K}} \lambda_k \mathbf{E} \left[\left[\frac{\hat{F}_k}{\hat{R}_k^{\max}} \right] \right] - \sum_{c \in \mathcal{C}} \mu_{c,s} \pi_c \right) \\ \text{s.t.} \quad &\sum_{l \in \mathcal{L}} p_{c,l} + \mu_{c,s} \leq 1 \quad \forall c. \end{aligned}$$

For the moment, let us assume Lagrange multipliers q_l and q_s are given. Then the maximization problem above can be decomposed into a collection of optimization problems, one for each c :

$$\begin{aligned} &\max_{p_{c,l}, \mu_{c,s}} \sum_{l \in \mathcal{L}} q_l R_{c,l} p_{c,l} + q_s \mu_{c,s} \\ \text{s.t.} \quad &\sum_{l \in \mathcal{L}} p_{c,l} + \mu_{c,s} \leq 1. \end{aligned}$$

It is easy to verify that one optimal solution to the optimization problem above is:

- if $q_s > \max_{l \in \mathcal{L}} q_l R_{c,l}$, then $\mu_{c,s} = 1$ and $p_{c,l} = 0 (\forall l)$;
- otherwise, $\mu_{c,s} = 0$, and $p_{c,l^*} = 1$ for some $l^* \in \arg \max q_l R_{c,l}$ and $p_{c,l} = 0$ for other l .

The complementary slackness conditions give

$$q_l \left(x_l - \sum_{c \in \mathcal{C}} \pi_c p_{c,l} R_{c,l} \right) = 0.$$

Since x_l is the mean arrival rate of long-lived flow l and $\sum_{c \in \mathcal{C}} \pi_c p_{c,l} R_{c,l}$ is the mean service rate, the condition on q_l says that if the mean arrival rate is less than the mean service rate, q_l is equal to zero. Along with the non-negativity condition on q_l , this suggests that perhaps q_l behaves like a queue with these arrival and service rates. Indeed, it turns out that the mean of the queue lengths are proportional to Lagrange multipliers (see the surveys in [8]–[10]). For long-lived flow l , we can treat the queue-length $Q_l(t)$ as a time-varying estimate of Lagrange multiplier q_l . Similarly q_s can be associated with a queue whose arrival rate is $\sum_{k \in \mathcal{K}} \lambda_k \mathbf{E} \left[\left[\frac{\hat{F}_k}{\hat{R}_k^{\max}} \right] \right]$, which is the mean rate at which workload arrives where workload is measured by the number of slots needed to serve a short-lived flow if it is served when its channel condition is the best. The service rate is $\sum_{c \in \mathcal{C}} \mu_{c,s} \pi_c$ which is the rate at which the workload can potentially decrease when a short-lived flow is picked for scheduling by the base station. Thus, the workload in the system can serve as a dynamic estimate of q_s .

Letting

$$W_s(t) = \left(\sum_{i \in \mathcal{I}(t)} \left[\frac{Q_i(t)}{R_i^{\max}} \right] \right)$$

and $\alpha W_s(t)$ ($\alpha > 0$) be an estimate of q_s , the observations above suggest the following workload-based scheduling algorithm if R_i^{\max} are known.

Workload-based Scheduling (WS): During each time slot, the base station checks the following inequality:

$$\alpha W_s(t) > \max_{l \in \mathcal{L}} Q_l(t) R_l(t). \quad (6)$$

- If inequality (6) holds, then the base station serves a short-lived flow as follows: if at least one short-lived flow (say flow i) satisfies $R_i(t) \geq Q_i(t)$ or $R_i(t) = R_i^{\max}$, then such a flow is selected for transmission (ties are broken arbitrarily); otherwise, the base station picks an arbitrary short-lived flow to serve.
- If inequality (6) does not hold, then the base station serves a long-lived flow l^* such that $l^* \in \arg \max_{l \in \mathcal{L}} Q_l(t) R_l(t)$ (ties are broken arbitrarily).
- The factor α can be obtained from the optimization formulation by multiplying constraint (5) by α on both sides. \square

However, this algorithm which was directly derived from dual decomposition considerations is not implementable since R_i^{\max} 's are unknown. So WSL uses $\hat{R}_i^{\max}(t)$ to approximate R_i^{\max} . Note that an inaccurate estimate of R_i^{\max} not only

affects the base station's decision on whether $R_i(t) = R_i^{\max}$, but also on its computation of $\left[\frac{Q_i(t)}{R_i^{\max}}\right]$. However, it is not difficult to see that the error in the estimate of the total workload is a small fraction of the total workload when the total workload is large. When the workload is very large, the total number of short-lived flows is large since their file sizes are bounded. Since the arrival rate of short-lived flows is also bounded, this further implies that the majority of short-lived flows must have arrived a long time ago, which means that with high probability, their estimate of their best channel conditions must be correct.

Next we will prove that both WS and WSL can stabilize any traffic x_l and λ_k such that $(1 + \epsilon)x_l$ and $(1 + \epsilon)\lambda_k$ are *supportable*, i.e., satisfying the conditions presented in Theorem 1. In other words, the number of short-lived flows in the network and the queues for long-lived flows are all bounded.

Theorem 2: Given any traffic x_l and λ_k such that $(1 + \epsilon)x_l$ and $(1 + \epsilon)\lambda_k$ are supportable, we have $\lim_{t \rightarrow \infty} \mathbf{E}[\sum_{l \in \mathcal{L}} Q_l(t) + \sum_{i \in \mathcal{I}(t)} Q_i(t)] < \infty$ under WS.

We skip the proof details due to space constraints. Interested readers can find the details in [12]. Drift conditions obtained as a part of this proof will be used to establish the stability of WSL (see equations (16) and (17) in the appendix).

We next study WSL, where R_i^{\max} is estimated from the history. We define $\Theta_{k,\beta,r}(t)$ to be the number of short-lived flows that belong to class- k , have a residual size of β , and have $\tilde{R}_i^{\max}(t) = r$. Furthermore, we define

$$\tilde{\mathbf{M}}(n) = \left(\begin{array}{c} \{Q_l(t)\}_{l \in \mathcal{L}}, \{\Theta_{k,\beta,r}(t)\}_{\substack{k \in \mathcal{K} \\ 1 \leq \beta \leq F^{\max} \\ 1 \leq r \leq \hat{R}_k^{\max}}} \\ (n-1)T+1 \leq t \leq nT \end{array} \right)$$

from some $T \geq D$. It is easy to see that $\tilde{\mathbf{M}}(n)$ is a *finite-dimensional* Markov chain under WSL.

Theorem 3: Given any traffic x_l and λ_k such that $(1 + \epsilon)x_l$ and $(1 + \epsilon)\lambda_k$ are supportable, there exists D_ϵ such that the Markov chain $\tilde{\mathbf{M}}(n)$ is *positive-recurrent* under any WSL with learning period $D \geq D_\epsilon$, and $\lim_{t \rightarrow \infty} \mathbf{E}[\sum_{l \in \mathcal{L}} Q_l(t) + \sum_{i \in \mathcal{I}(t)} Q_i(t)] < \infty$ under WSL.

The proof is presented in the appendix.

VI. SIMULATIONS

In this section, we use simulations to evaluate the performance of WSL and compare it to other scheduling policies. There are two types of flows used in the simulations:

- **S-flow:** An S-flow has a finite size, generated from a truncated exponential distribution with mean value 30 and maximum value 150. Non-integer values are rounded to integers.
- **L-flow:** An L-flow keeps injecting bits into the network and never leaves the network. The number of bits generated at each time slot follows a Poisson distribution with mean value 1.

Here S-flows represent short-lived flows that have finite sizes and whose bits arrive all at once, and L-flows represent long-lived flows that continuously inject bits and never leave the network.

We assume that the channel between each user and the base station is distributed according to one of the following two distributions:

- **G-link:** A G-link has five possible link rates $\{10, 20, 30, 40, 50\}$, and each of the states happens with probability 20%.
- **P-link:** A P-link has five possible link rates $\{5, 10, 15, 20, 25\}$, and each of the states happens with probability 20%.

The G and P stand for Good and Poor, respectively. In [12], we present more simulations with other channel state distributions as well. We set $\alpha = 50$ and $D = 16$ in our simulations.

In the introduction, we have pointed out that the MaxWeight is not throughput optimal under flow-level dynamics because the backlog of a short-lived queue does not build up even when it has not being served for a while. To overcome this, one could try to use the delay of the head-of-line packet, instead of queue-length, as the weight because the head-of-line delay will keep increasing if no service is received. In the case of long-lived flows only, this algorithm is known to be throughput-optimal [4]. We will show that this Delay-based scheduling does not solve the instability problem when there are short-lived flows.

Delay-based Scheduling: At each time slot, the base station selects a flow i such that $i \in \arg \max_i D_i(t)R_i(t)$, where $D_i(t)$ is the delay experienced so far by the head-of-line packet of flow i . □

Simulation I: Performance comparison of various algorithms

We first consider the case where all flows are S-flows, which arrive according to a truncated Poisson process with maximum value 100 and mean λ . An S-flow is assigned a G-link or a P-link equally likely.

Figure 1 shows the average file-transfer delay and average number of S-flows under different values of λ . We can see that WSL performs significantly better than the MaxWeight and Delay-based algorithms. Specifically, under MaxWeight and Delay-based algorithms, both the number of S-flows and file-transfer delay explode when $\lambda \geq 0.102$. WSL, on the other hand, performs well even when $\lambda = 0.12$.

Next, we consider the same scenario with three L-flows in the network. Two of the L-flows have G-links and one has a P-link. Figure 2 shows the number of short-lived flows and file-transfer delay under different values of λ . We can see that the MaxWeight becomes unstable even when *the arrival rate of S-flows is very small*. This is because the MaxWeight stops serving S-flows when the backlogs of F-flows are large, so S-flows stay in the network forever. The delay-based scheduling performs better than the MaxWeight, but significantly worse than WSL.

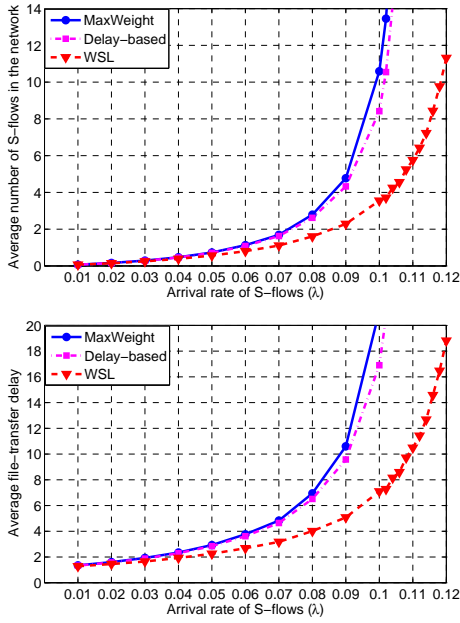


Fig. 1. The performance of the delay-based, MaxWeight, and WSL algorithms in a network without long-lived flows

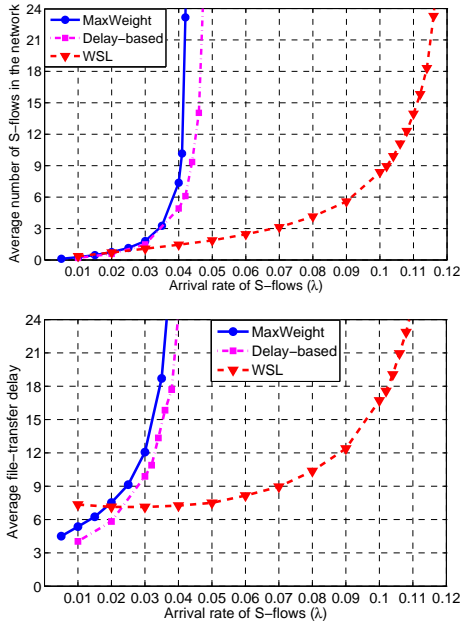


Fig. 2. The performance of the delay-based, MaxWeight, and WSL algorithms in a network with both S-flows and L-flows

Simulation II: Blocking probability of various algorithms

While our theory assumes that the number of flows in the network can be infinite, in reality, base stations limit the number of simultaneously active flows, and reject new flows when the number of existing flows above some threshold. In this simulation, we assume that the base station can support at most 20 S-flows. A new S-flow will be blocked if 20 S-

flows are already in the network. In this setting, the number of flows in the network is finite, so we compute the blocking probability, i.e., the fraction of S-flows rejected by the base station.

We consider the case where no long-lived flow is in the network and the case where both short-lived and long-lived flows are present in the network. The flows and channels are selected as in Simulation II. The results are shown in Figure 3 and 4. We can see that the blocking probability under WSL is substantially smaller than that under the MaxWeight or delay-based scheduling. Thus, this simulation demonstrates that instability under the assumption when the number of flows is allowed to unbounded implies high blocking probabilities for the practical scenario when the base station limits the number of flows in the network.

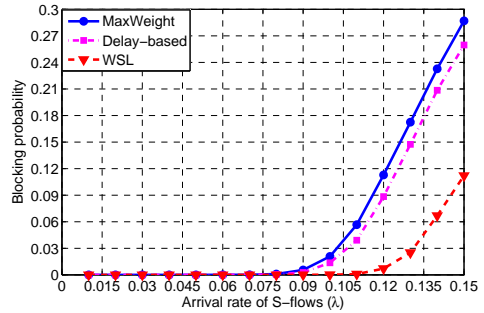


Fig. 3. The blocking probabilities of the delay-based, MaxWeight, and WSL in a network without L-flows

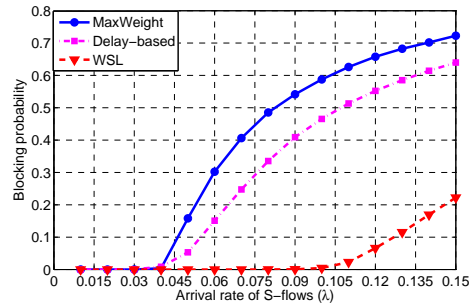


Fig. 4. The blocking probabilities of the delay-based, MaxWeight, and WSL in a network with L-flows

VII. CONCLUSIONS

In this paper, we studied multiuser scheduling in networks with flow-level dynamics. We first obtained necessary conditions for flow-level stability of networks with both long-lived flows and short-lived flows. Then based on an optimization framework, we proposed the workload-based scheduling with learning that is throughput-optimal under flow-level dynamics and requires no prior knowledge about channels and traffic. In the simulations, we evaluated the performance of the proposed scheduling algorithms, and demonstrated that the proposed algorithm performs significantly better than the MaxWeight algorithm and the Delay-based algorithm in various settings.

Acknowledgments: Research supported by NSF Grants 07-21286 and 08-31756, ARO MURI Subcontracts, and the DTRA grants HDTRA1-08-1-0016 and HDTRA1-09-1-0055.

APPENDIX A: PROOF OF THEOREM 3

Define $\mathcal{E}_{miss}(t)$ to be the event such that the tie-breaking rule selects a short-lived flow with $\tilde{R}_i^{\max}(t) \neq R_i^{\max}$. We first introduce the following lemma which states $\mathcal{E}_{miss}(t)$ occurs with a small probability when the workload is large.

Proposition 4: Given any $\epsilon < 0$, there exist N_ϵ and D_ϵ such that conditioned on $D \geq D_\epsilon$ and $W_s(t-D) \geq N_\epsilon$, $\Pr(\mathcal{E}_{miss}(t)) \leq \epsilon$ under the WSL. \square

The intuition behind the proof of this lemma is the following: if the workload of short-lived flows is large, then the number of short-lived flows must be large. Since the arrival rate of short-lived flows is bounded, a substantial number of the flows must have arrived a long time ago which means that their estimate of their individual best channel state must be correct with high probability. We skip the details which can be found in [12].

Consider the network that is operated under WSL, and define $\mathcal{H}(t)$ to be

$$\mathcal{H}(t) \triangleq \left\{ Q_l(t), R_l(t), Q_i(t), R_i(t), \tilde{R}_i^{\max}(t) \right\}.$$

Now given $\mathcal{H}(t)$, we define the following notations:

- Define $\mu_{2;l}(t) = R_l(t)$ if flow l is selected by WSL, and $\mu_{2;l}(t) = 0$ otherwise.
- Define $\mu_{2;i}(t) = 1$ if flow i is selected by WSL and the workload of flow i can be reduced by one, and $\mu_{2;i}(t) = 0$ otherwise.
- Define $\mu_{1;l}(t) = R_l(t)$ if flow l is selected by WS, and $\mu_{1;l}(t) = 0$ otherwise.
- Define $\mu_{1;i}(t) = 1$ if flow i is selected by WS and the workload of flow i can be reduced by one, and $\mu_{1;i}(t) = 0$ otherwise.

We remark that $\mu_{2;j}(t)$ is the action selected by the base station at time t under WSL and $\mu_{1;j}(t)$ is the action selected by the base station at time t under WS, assuming the same history $\mathcal{H}(t)$. We define

$$A_s(t) = \sum_{k \in \mathcal{K}} \sum_{i=1}^{\Lambda_k(t)} \left[\frac{f_i}{\tilde{R}_k^{\max}} \right],$$

which is the amount of new workload (from short-lived flows) injected in the network at time t , and $\mu_s(t)$ to be the decrease of the workload at time t , i.e., $\mu_s(t) = 1$ if the workload of short-lived flows is reduced by one and $\mu_s(t) = 0$ otherwise. Based on the notations above, the evolution of short-lived flows can be described as:

$$W_s(t+1) = W_s(t) + A_s(t) - \mu_s(t).$$

Further, the evolution of $Q_l(t)$ can be described as

$$Q_l(t+1) = Q_l(t) + X_l(t) - \mu_l(t) + u_l(t),$$

where $\mu_l(t)$ is the decrease of $Q_l(t)$ due to the service long-lived flow l receives at time t , and $u_l(t)$ is the unused service due to the lack of data in the queue. We further define

$$\bar{\lambda} = \left(\sum_{k \in \mathcal{K}} \lambda_k \mathbf{E} \left[\left[\frac{\hat{F}_k}{\tilde{R}_k^{\max}} \right] \right] \right),$$

where \tilde{R}_k^{\max} is the largest achievable link rate of class- k short-lived flows.

We define the Lyapunov function to be

$$V(n) = \alpha (W_s(nT))^2 + \sum_{l \in \mathcal{L}} (Q_l(nT))^2. \quad (7)$$

First, it is easy to verify that there exists U_1 independent of $\tilde{\mathbf{M}}(n)$ such that

$$\begin{aligned} & \mathbf{E}[V(n+1) - V(n) | \tilde{\mathbf{M}}(n)] \\ & < U_1 + 2\alpha \mathbf{E} \left[W_s(nT) \sum_{t=nT}^{(n+1)T-1} (A_s(t) - \mu_{2;s}(t)) \middle| \tilde{\mathbf{M}}(n) \right] \\ & + \sum_{l \in \mathcal{L}} 2\mathbf{E} \left[Q_l(nT) \sum_{t=nT}^{(n+1)T-1} (X_l(t) - \mu_{2;l}(t)) \middle| \tilde{\mathbf{M}}(n) \right]. \end{aligned}$$

Dividing the time into two segments $[nT, nT + D - 1]$ and $[nT + D, (n+1)T - 1]$ and noticing the fact that $|Q_l(t_1) - Q_l(t_2)|$ and $|W_k(t_1) - W_k(t_2)|$ are both bounded by some constants independent of $\tilde{\mathbf{M}}(n)$, we can obtain that

$$\begin{aligned} & \mathbf{E}[V(n+1) - V(n) | \tilde{\mathbf{M}}(n)] \\ & < \tilde{U} + 2\alpha W_s(nT) \bar{\lambda} D + 2 \sum_{l \in \mathcal{L}} Q_l(nT) x_l D \\ & + 2\mathbf{E} \left[\alpha \sum_{t=nT+D}^{(n+1)T-1} W_s(t) (A_s(t) - \mu_{2;s}(t)) \middle| \tilde{\mathbf{M}}(n) \right] \\ & + \sum_{l \in \mathcal{L}} 2\mathbf{E} \left[\sum_{t=nT+D}^{(n+1)T-1} Q_l(t) (X_l(t) - \mu_{2;l}(t)) \middle| \tilde{\mathbf{M}}(n) \right], \end{aligned}$$

where \tilde{U} is some constant independent of $\tilde{\mathbf{M}}(n)$. Now, by adding and subtracting $\mu_{1;s}(t)$, we obtain

$$\begin{aligned} & \mathbf{E}[V(n+1) - V(n) | \tilde{\mathbf{M}}(n)] \\ & \leq \tilde{U} + 2\alpha W_s(nT) \bar{\lambda} D + 2 \sum_{l \in \mathcal{L}} Q_l(nT) x_l D + \sum_{t=nT+D}^{(n+1)T-1} \text{Drift}(t), \end{aligned}$$

where

$$\begin{aligned} & \text{Drift}(t) \\ & = 2\mathbf{E} \left[\alpha W_s(t) A_s(t) + \sum_{l \in \mathcal{L}} Q_l(t) X_l(t) \middle| \tilde{\mathbf{M}}(n) \right] \quad (8) \end{aligned}$$

$$- 2\mathbf{E}[\alpha W_s(t) \mu_{1;s}(t) + \sum_{l \in \mathcal{L}} Q_l(t) \mu_{1;l}(t) | \tilde{\mathbf{M}}(n)] \quad (9)$$

$$+ \sum_{l \in \mathcal{L}} 2\mathbf{E}[Q_l(t) (\mu_{1;l}(t) - \mu_{2;l}(t)) | \tilde{\mathbf{M}}(n)] \quad (10)$$

$$+ 2\mathbf{E}[\alpha W_s(t) (\mu_{1;s}(t) - \mu_{2;s}(t)) | \tilde{\mathbf{M}}(n)]. \quad (11)$$

Note that (10)+(11) is the difference between the WS and the WSL. In the following analysis, we will prove that this difference is small compared to the absolute value of (8)+(9). We define

$$\begin{aligned} \text{Diff}(t) &= \alpha W_s(t) (\mu_{1;s}(t) - \mu_{2;s}(t)) \\ &\quad + \sum_{l \in \mathcal{L}} Q_l(t) (\mu_{1;l}(t) - \mu_{2;l}(t)), \end{aligned}$$

and

$$\tilde{W}_s(t) = \sum_{i \in \mathcal{I}(t)} \left\lceil \frac{Q_i(t)}{\tilde{R}_i^{\max}(t)} \right\rceil.$$

Next we compute its value in three different situations:

- **Situ-A:** Consider the situation in which $\alpha \tilde{W}_s(t) \leq \max_{l \in \mathcal{L}} Q_l(t) R_l(t)$. We note that $\tilde{W}_s(t) \geq W_s(t)$ since $\tilde{R}_i^{\max}(t) \leq R_i^{\max}$ for all t and i . Therefore, given $\alpha W_s(t) \leq \sum_{l \in \mathcal{L}} Q_l(t)$, both WS and WSL will select a long-lived flow. In this case, we can conclude that $\mu_{1;l}(t) = \mu_{2;l}(t)$ and $\mu_{1;s}(t) = \mu_{2;s}(t) = 0$, which implies that $\text{Diff}(t) = 0$.
- **Situ-B:** Consider the situation in which $\alpha W_s(t) > \max_{l \in \mathcal{L}} Q_l(t) R_l(t)$. In this case, both WS and WSL will select a short-lived flow, which implies that $\mu_{1;l}(t) = \mu_{2;l}(t) = 0$, and

$$\begin{aligned} \text{Diff}(t) &= \alpha W_s(t) (\mu_{1;s}(t) - \mu_{2;s}(t)) \\ &\leq \alpha W_s(t) (1 - \mu_{2;s}(t)). \end{aligned}$$

- **Situ-C:** Consider the situation in which $\alpha \tilde{W}_s(t) > \max_{l \in \mathcal{L}} Q_l(t) R_l(t) \geq \alpha W_s(t)$. In this case, WS will select a long-lived flow and WSL will select a short-lived flow. We hence have $\mu_{1;l}(t) > 0$ and $\mu_{1;s}(t) = \mu_{2;l}(t) = 0$, which implies that

$$\begin{aligned} \text{Diff}(t) &= \max_{l \in \mathcal{L}} Q_l(t) R_l(t) - \alpha W_s(t) \mu_{2;s}(t) \\ &\leq \alpha \tilde{W}_s(t) - \alpha W_s(t) \mu_{2;s}(t) \end{aligned}$$

According to the analysis above, we have that

$$\begin{aligned} &\mathbf{E}[\text{Diff}(t) | \tilde{\mathbf{M}}(n)] \\ &\leq \mathbf{E} \left[\alpha W_s(t) | \text{Situ-B}, \mu_{2;s} = 0, \tilde{\mathbf{M}}(n) \right] \times \\ &\quad \Pr \left(\text{Situ-B}, \mu_{2;s} = 0 | \tilde{\mathbf{M}}(n) \right) \\ &+ \mathbf{E} \left[\alpha \tilde{W}_s(t) | \text{Situ-C}, \mu_{2;s} = 0, \tilde{\mathbf{M}}(n) \right] \times \\ &\quad \Pr \left(\text{Situ-C}, \mu_{2;s} = 0 | \tilde{\mathbf{M}}(n) \right) \\ &+ \mathbf{E} \left[\alpha \tilde{W}_s(t) - \alpha W_s(t) | \text{Situ-C}, \mu_{2;s} = 1, \tilde{\mathbf{M}}(n) \right] \times \\ &\quad \Pr \left(\text{Situ-C}, \mu_{2;s} = 1 | \tilde{\mathbf{M}}(n) \right). \end{aligned}$$

Next we define a finite set $\tilde{\Upsilon}$. We first introduce some constants:

- $\epsilon_1 = \min \left\{ \frac{\bar{\lambda}\epsilon}{32}, \frac{\epsilon \min_l x_l}{8R^{\max}} \right\}$.
- $\epsilon_2 = \frac{\bar{\lambda}\epsilon}{32R^{\max}}$, and D_{ϵ_2} and N_{ϵ_2} are the numbers that guarantee $\Pr(\mathcal{E}_{miss}(t)) \leq \epsilon_2$, which are defined by the goodness of the tie-breaking rule.

- $\lambda_W^{\max} = K\lambda^{\max}F^{\max}$, which is the maximum number of bits of short-lived flows injected in one time slot, and also the upper bound on the new workload injected in the network in one time slot.

We define a set $\tilde{\Upsilon}$ such that

$$\tilde{\Upsilon} = \left\{ \tilde{\mathbf{M}}(n) : \begin{array}{l} W_s(nT) \leq \tilde{U}_W + 2T + \frac{2\sum_l x_l R^{\max} T}{\alpha\bar{\lambda}} \\ Q_l(nT) \leq \tilde{U}_Q + \frac{2\alpha\bar{\lambda}T}{\min_l x_l} + \frac{2TR^{\max}\sum_l x_l}{\min_l x_l} \forall l \end{array} \right\}.$$

In this definition, \tilde{U}_W is a positive integer satisfying that

$$(1 - p_s^{\max}) \frac{\tilde{U}_W}{F^{\max}} \leq \epsilon_1, \quad (12)$$

$$\tilde{U}_W \geq \frac{8\tilde{U} + 16\epsilon_2\alpha\lambda_W^{\max}T + 8\alpha DR^{\max} + 16\epsilon_2\alpha R^{\max}T + 8\lambda_W^{\max}D}{\epsilon\alpha\bar{\lambda}} \quad (13)$$

$$\frac{\tilde{U}_W}{F^{\max}} \geq N_{\epsilon_2}, \quad (14)$$

and \tilde{U}_Q is a positive integer satisfying

$$\tilde{U}_Q \geq \frac{8\tilde{U} + 12\alpha R^{\max}(\tilde{U}_W + \frac{2\sum_l x_l R^{\max} T}{\alpha\bar{\lambda}}) + (\lambda_W^{\max} + 2)T}{\epsilon \min_l x_l}. \quad (15)$$

Since the changes of $W_s(t)$ and $Q_l(t)$ during each time slot is bounded by some constants independent of $\tilde{\mathbf{M}}(n)$, it is easy to verify that $\tilde{\Upsilon}$ is a set of a finite number of elements.

Note that (8)+(9) is the drift under the WS. The WS policy has been analyzed in the proof of Theorem 2, which can be found in [12]. Here, we list two facts that will be used later:

- Given that $W_s(nT) > \tilde{U}_W + 2T + \frac{2\sum_l x_l R^{\max} T}{\alpha\bar{\lambda}}$, the following inequality holds:

$$\begin{aligned} \mathbf{E}[(8) + (9) | \tilde{\mathbf{M}}(n)] &\leq \mathbf{E} \left[2\epsilon_1 \left(\alpha W_s(t) + R^{\max} \max_{l \in \mathcal{L}} Q_l(t) \right) \right. \\ &\quad \left. - 2\epsilon\alpha W_s(t)\bar{\lambda} - 2\epsilon \sum_{l \in \mathcal{L}} Q_l(t)x_l \middle| \tilde{\mathbf{M}}(n) \right] \quad (16) \end{aligned}$$

- Given that $W_s(nT) < \tilde{U}_W + 2T + \frac{2\sum_l x_l R^{\max} T}{\alpha\bar{\lambda}}$ and $Q_l(nT) > \tilde{U}_Q + \frac{2\alpha\bar{\lambda}T}{\min_l x_l} + \frac{2TR^{\max}\sum_l x_l}{\min_l x_l}$ for some l , the following inequality holds:

$$\begin{aligned} &\mathbf{E}[(8) + (9) | \tilde{\mathbf{M}}(n)] \\ &\leq \mathbf{E} \left[4\alpha W_s(t) - 2\epsilon\alpha W_s(t)\bar{\lambda} - 2\epsilon \sum_{l \in \mathcal{L}} Q_l(t)x_l \middle| \tilde{\mathbf{M}}(n) \right] \quad (17) \end{aligned}$$

Next, we analyze the drift of Lyapunov function case by case assuming that $D > \left\lceil \frac{\log \bar{\lambda}\epsilon - \log 16 - \log R^{\max}}{\log(1 - \rho_s^{\max})} \right\rceil$ and $T > \left\lceil \frac{(4+\epsilon)D}{\epsilon} \right\rceil$.

- **Case I:** Assume that $\tilde{M}(n) \in \tilde{\Upsilon}$. In this case, it is easy to verify that $\mathbf{E}[V(n+1) - V(n) | \tilde{\mathbf{M}}(n)]$ is bounded by some constant \tilde{U}_d .
- **Case II:** Assume that

$$W_s(nT) > \tilde{U}_W + 2T + \frac{2\sum_l x_l R^{\max} T}{\alpha\bar{\lambda}} \geq \tilde{U}_W + T.$$

Recall that $\mathcal{E}_{miss}(t)$ is the event such that the tie-breaking rule selects a short-lived flow with $\tilde{R}_i^{\max}(t) \neq R_i^{\max}$. Note that $\mu_{2;s}(t) = 0$ implies that $\mathcal{E}_{miss}(t)$ occurs. Also note the following facts:

- For any $nT \leq t \leq (n+1)T$, we have $W(t) \leq W(nT) + \lambda_W^{\max} T$,
- Given $W_s(nT) \geq \tilde{U}_W + T$, we have $W_s(t) \geq \tilde{U}_W$ for all $nT \leq t \leq (n+1)T - 1$. Then according to the definition of ϵ_2 and \tilde{U}_W and assumption that the tie-breaking rule is good, we have $\Pr(\mathcal{E}_{miss}(t)) \leq \epsilon_2$ for all $nT + D \leq t \leq (n+1)T - 1$.
- Given any $\tilde{\mathbf{M}}(n)$ and any $nT + D \leq t \leq (n+1)T - 1$, we have

$$\begin{aligned}
& \mathbf{E} \left[\alpha \tilde{W}_s(t) - \alpha W_s(t) \mid \text{Situ-C}, \mu_{2;s} = 1, \tilde{\mathbf{M}}(n) \right] \times \\
& \Pr \left(\text{Situ-C}, \mu_{2;s} = 1 \mid \tilde{\mathbf{M}}(n) \right) \\
& \leq \mathbf{E} \left[\alpha \tilde{W}_s(t) - \alpha W_s(t) \mid \tilde{\mathbf{M}}(n) \right] \\
& = \mathbf{E} \left[\mathbf{E} \left[\alpha \tilde{W}_s(t) - \alpha W_s(t) \mid W_s(t-D) \right] \mid \tilde{\mathbf{M}}(n) \right] \\
& \leq \alpha \mathbf{E} \left[(1 - p_s^{\max})^D W_s(t-D) R^{\max} + \lambda_W^{\max} D \mid \tilde{\mathbf{M}}(n) \right] \\
& \leq \alpha \mathbf{E} \left[(1 - p_s^{\max})^D (W_s(t) + D) R^{\max} + \lambda_W^{\max} D \mid \tilde{\mathbf{M}}(n) \right], \tag{18}
\end{aligned}$$

where the inequality (18) holds because at most $\lambda_W^{\max} D$ bits belonging to short-lived flows are in the network for less than D time slots at time t , and a flow having been in the network for at least D time slots can estimate correctly its workload with a probability at least $1 - (1 - p_s^{\max})^D$.

Now according to the observations above, we can obtain that

$$\begin{aligned}
& \mathbf{E}[\text{Diff}(t) \mid \tilde{\mathbf{M}}(n)] \\
& \leq \epsilon_2 \alpha (W_s(nT) + \lambda_W^{\max} T) + \epsilon_2 \alpha (R^{\max} W_s(nT) + \lambda_W^{\max} T) \\
& + \mathbf{E} \left[\alpha (1 - p_s^{\max})^D (W_s(t) + D) R^{\max} + \alpha \lambda_W^{\max} D \mid \tilde{\mathbf{M}}(n) \right].
\end{aligned}$$

Combining with inequality (16), we conclude that

$$\begin{aligned}
& \text{Drift}(t) \\
& \leq 2\mathbf{E} \left[\epsilon_1 \left(\alpha W_s(t) + R^{\max} \max_{l \in \mathcal{L}} Q_l(t) \right) \right. \\
& \quad \left. - \epsilon \alpha W_s(t) \bar{\lambda} - \epsilon \sum_{l \in \mathcal{L}} Q_l(t) x_l \right. \\
& \quad \left. + \epsilon_2 \alpha (W_s(nT) + \lambda_W^{\max} T) \right. \\
& \quad \left. + \epsilon_2 \alpha (R^{\max} W_s(nT) + \lambda_W^{\max} T) \right. \\
& \quad \left. + \alpha (1 - p_s^{\max})^D (W_s(t) + D) R^{\max} + \alpha \lambda_W^{\max} D \mid \tilde{\mathbf{M}}(n) \right] \\
& \leq \mathbf{E} \left[-\epsilon \left(\alpha \bar{\lambda} W_s(t) + \sum_{l \in \mathcal{L}} x_l Q_l(t) \right) \mid \tilde{\mathbf{M}}(n) \right],
\end{aligned}$$

where the last inequality holds due to (13).

- **Case III:** Assume that $W_s(nT) < \tilde{U}_W + 2T + \frac{2 \sum_l x_l R^{\max} T}{\alpha \bar{\lambda}}$ and $Q_l(nT) > \tilde{U}_Q + \frac{2\alpha \bar{\lambda} T}{\min_l x_l} + \frac{2TR^{\max} \sum_l x_l}{\min_l x_l} > \tilde{U}_Q$ for some l . In this case, we have

$$\text{Diff}(t) \leq \alpha \tilde{W}_s(t) \leq \alpha R^{\max} W_s(t).$$

Combining with inequality (17), we have that

$$\text{Drift}(t) \leq \mathbf{E} \left[-\epsilon \left(\alpha \bar{\lambda} W_s(t) + \sum_{l \in \mathcal{L}} x_l Q_l(t) \right) \mid \tilde{\mathbf{M}}(n) \right],$$

where the last inequality holds due to (15). \square

Now, combining case II and case III, we can obtain that

$$\begin{aligned}
& \mathbf{E}[V(n+1) - V(n) \mid \tilde{\mathbf{M}}(n)] \\
& \leq \tilde{U} + 2\alpha W_s(nT) \bar{\lambda} D + 2 \sum_{l \in \mathcal{L}} Q_l(nT) x_l D \\
& + \sum_{t=nT+D}^{(n+1)T-1} \mathbf{E} \left[-\epsilon \left(\alpha \bar{\lambda} W_s(t) + \sum_{l \in \mathcal{L}} x_l Q_l(t) \right) \mid \tilde{\mathbf{M}}(n) \right] \\
& \leq -\tilde{U} - \sum_{t=nT+D}^{(n+1)T-1} \mathbf{E} \left[\frac{\epsilon}{2} \left(\alpha \bar{\lambda} W_s(t) + \sum_{l \in \mathcal{L}} x_l Q_l(t) \right) \mid \tilde{\mathbf{M}}(n) \right],
\end{aligned}$$

where the last inequality yields from the definition of \tilde{U}_W and \tilde{U}_Q , and the fact $\tilde{\mathbf{M}}(n) \notin \tilde{\mathcal{Y}}$. Finally, we can conclude the theorem from the Foster's Criterion and $\lim_{t \rightarrow \infty} \mathbf{E}[\sum_{l \in \mathcal{L}} Q_l(t) + \sum_{i \in \mathcal{I}(t)} Q_i(t)] < \infty$ [13].

REFERENCES

- [1] P. van de Ven, S. Borst, and S. Shneer, "Instability of MaxWeight scheduling algorithms," in *Proc. IEEE Infocom.*, Rio de Janeiro, Brazil, April 2009, pp. 1701 – 1709.
- [2] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks", *IEEE Transactions on Automatic Control*, Vol. 37, No. 12, pp. 1936-1949, December 1992.
- [3] M. Andrews, K. Kumaran, K. Ramanan, A. Stolyar, R. Vijayakumar, and P. Whiting, "Scheduling in a queueing system with asynchronously varying service rates," *Probability in the Engineering and Informational Sciences*, vol. 18, pp. 191–217, 2004.
- [4] A. Eryilmaz, R. Srikant, and J. R. Perkins, "Stable scheduling policies for fading wireless channels," *IEEE/ACM Trans. Network.*, vol. 13, no. 2, pp. 411–424, 2005.
- [5] M. J. Neely, E. Modiano, and C. E. Rohrs, "Dynamic Power Allocation and Routing for Time Varying Wireless Networks," *IEEE Journal on Selected Areas in Communications*, Special Issue on Wireless Ad-Hoc Networks, vol. 23, no. 1, pp. 89-103, Jan. 2005.
- [6] X. Liu, E. Chong, and N. Shroff, "Opportunistic Transmission Scheduling with Resource-Sharing Constraints in Wireless Networks," *IEEE Journal on Selected Areas in Communications*, vol. 19, no. 10, pp. 2053-2064, October, 2001.
- [7] P. Viswanath, D. Tse and R. Laroia, "Opportunistic Beamforming using Dumb Antennas," *IEEE Transactions on Information Theory*, Vol. 48, No. 6, pp. 1277-1294, June 2002.
- [8] X. Lin, N. B. Shroff and R. Srikant, "A Tutorial on Cross-Layer Optimization in Wireless Networks," in *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 8, August 2006.
- [9] L. Georgiadis, M. Neely and L. Tassiulas, *Resource Allocation and Cross Layer Control in Wireless Networks*, NoW publishers, 2006.
- [10] S. Shakkottai and R. Srikant, *Network Optimization and Control*. NoW publishers, 2007.
- [11] S. Asmussen. *Applied Probability and the Theory of Queues*. Springer, 2003.
- [12] S. Liu, L. Ying and R. Srikant, "Throughput-Optimal Opportunistic Scheduling in the Presence of Flow-Level Dynamics, *Technical Report*, 2009. <http://arxiv.org/abs/0907.3977>
- [13] MEYN, S., AND TWEEDIE, R. L. *Markov chains and stochastic stability*. Cambridge University Press, 2009.