Head-of-Line Access Delay-Based Scheduling Algorithm for Flow-Level Dynamics

Yi Chen, *Student Member, IEEE*, Xuan Wang, and Lin Cai, *Senior Member, IEEE*

***Abstract*—Scheduling algorithm design is a critical and challenging issue in multiuser wireless networks with dynamic flows. The well-known Queue-length-based MaxWeight (QMW) scheduling algorithm can achieve throughput-optimality if there only exist persistent flows that are long-lived with infinite traffic arrival. In this paper, we propose a head-of-line access delay (HAD)-based scheduling algorithm and show that it is throughput-optimal when the flows are dynamic, i.e., they are short-lived with finite data to transmit. HAD is an online algorithm and does not require prior knowledge of the statistics of the arrival traffic and channel information. We also develop the Markov analytic model to study system performance and reveal important properties of the proposed HAD scheduling algorithm. To reduce the complexity of the analysis, we further study two approximation methods corresponding to different arrival traffic intensity. Performance evaluation shows that the HAD scheduling algorithm can outperform the classic QMW and stabilize the system at the presence of flow-level dynamics. Compared to the other existing algorithms, HAD is practical to implement with a better delay performance.**

***Index Terms*—Communication system traffic control, cross layer design, wireless networks.**

I. Introduction

Scheduling is one of the core problems in the design of wireless networks to maintain efficient and high-quality wireless communications. Given a time-slotted system, a scheduling algorithm determines which user is allowed to transmit in each time slot. The scheduling algorithm design has been investigated in various topics such as the routing algorithm design [2], [3], the tradeoff analysis between throughput, energy and delay in wireless networks [4], [5], etc. Among various scheduling algorithms, throughput-optimal scheduling algorithms have been widely investigated, thanks to their capability of maintaining the queueing stability in the system as long as the arrival rate lies within the system capacity region [6]–[8]. The Queue-length-based MaxWeight (QMW) scheduling algorithm has been proposed in the pioneering works of Tassiulas and Ephremides [9]–[11] and then was thoroughly investigated in the literature [12]–[16]. QMW considers both the multiuser diversity gain and the real-time user demand in each time slot by prioritizing the flow, which has the greatest product of the current transmission rate and the user's backlog and has proven to be throughput-optimal if the data flow of each user is long-lived, i.e., the per-user flow is infinite (which we consider as a persistent flow), and the number of users in the system does not change over time. Besides throughput, other metrics were studied in the subsequent works such as fairness [17]–[19] and energy efficiency [20], [21].

However, in most of the practical communication networks, there are short-lived dynamic flows that arrive in the system, transmit a finite amount of data, and then leave the system once the demanded service is complete. It has been shown that the QMW scheduling algorithm and its variants are no longer throughput-optimal if some or all of the users are dynamic in the system [22], including the exponential rule [23] and the log rule [24]–[27], which were proposed to reduce the maximum and average flow delay. The reason of the instability can be briefly explained as follows. Consider a flow with the last packet waiting for transmission. By adopting QMW, without new traffic arrivals, the queue length of this flow will remain small, which will result in a long (or possibly infinite) delay of the flow, since the other flows with a larger queue length will have a higher priority. This problem is often referred as the "last packet problem," which will cause instability with flow-level dynamics. As a result, the scheduling algorithm design is a critical and challenging issue in real-world multiuser wireless networks with flow-level dynamics [28], which motivates this work.

The main contributions of this paper are threefold. First, we obtain the condition of queue stability for flow-level dynamics. Second, we propose our simple online scheduling algorithm, the head-of-line access delay (HAD)-based MaxWeight scheduling, and discuss its throughput-optimality with flow-level dynamics. We also explain the advantage of HAD over the existing works. Third, we evaluate the performance of our HAD algorithm regarding the number of users in the system and queueing delay by both analysis and simulation. We observe that HAD is able to stabilize the system, and our analytical results well match the simulation ones.

The rest of this paper is organized as follows. In Section II, the related work is introduced. Section III presents the system model, including the traffic arrival and departure models, and the definition of system capacity region. In Section IV, the HAD scheduling algorithm is proposed and the throughput-

Manuscript received February 15, 2016; revised August 6, 2016; accepted October 25, 2016. Date of publication November 4, 2016; date of current version June 16, 2017. This work was supported in part by the Natural Sciences and Engineering Research Council of Canada. Part of this paper was presented at IEEE GLOBECOM 2014, Austin, TX, USA. The review of this paper was coordinated by Prof. J. Liu.

Y. Chen and L. Cai are with the University of Victoria, Victoria, BC V8W 2Y2, Canada (e-mail: chenyi@uvic.ca; cai@ece.uvic.ca).

X. Wang was with the University of Victoria, Victoria, BC V8W 2Y2, Canada. He is now with Fortinet Technologies (Canada) Inc., Burnaby, BC, V5C 6G9, Canada (e-mail: xuan_wang@ece.uvic.ca).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVT.2016.2625326

optimality is studied. The queueing behavior of HAD is analyzed in Section V. Performance evaluation is presented in Section VI, followed by the concluding remarks in Section VII.

II. Related Work

Utility-based scheduling is one type of various scheduling algorithm. One example is the proportional fairness (PF) scheduling algorithm, which has been adopted in the cellular systems such as the current LTD networks. PF guarantees the fairness performance for all the users in the system, while considering the history resource allocation and the multiuser diversity gain. However, PF is not throughput-optimal, as shown in [29]. It has been proven in [7] that the utility-based scheduling algorithms, including the PF scheduling, are not throughput-optimal, and thus, the corresponding stability region is less than the system capacity region in general.

The application of throughput-optimal scheduling with a fixed number of users in the system can be found in a wide range of areas such as smart grid, secure transmission, and so on [30]–[32]. How to implement distributed throughput-optimal scheduling algorithms regarding the heavy tailed traffic in practical scenarios was investigated in [33]–[36]. When considering flow-level dynamics, QMW is not applicable, and several new scheduling algorithms other than QMW have been proposed. In [22], Peter van de Ven *et al.* presented examples to explain that QMW is not throughput-optimal in the single-hop networks with dynamic flows and time-varying channels, and designed a backlog-grouping based algorithm to schedule dynamic flows for system stability. However, it is complicated to implement, since the prior knowledge of the channel and the traffic arrival of each flow is required, which limits its application. In the subsequent work in [37], the spatial inefficiency of QMW was also investigated, and a location-based grouping scheme as the solution was proposed. More examples and results in multi-hop wireless networks were given in [38] to further explain the instability and inefficiency of QMW.

A flow-delay-based MaxWeight (F-D-MW) scheduling algorithm was studied in [39], aiming to stabilize the system with dynamic flows. In the proof of the throughput-optimality, however, it is required that the channel rate distribution of each flow is independent and identically distributed (i.i.d.) in the system, or at least the maximum possible transmission rate of each flow stays identical, while the stability analysis in the heterogeneous networks was not complete. F-D-MW uses the product of the flow delay and the channel transmission rate as the weight, and thus, the old flows that enter the system earlier will be always assigned a higher priority. As a result, it may make the new users entering in the system later suffer a long start-up latency, which is not desired by users with stringent delay requirement. Its throughput-optimality in the heterogeneous networks with both long-lived and short-lived flows was proved in [40], but only the spatial inefficiency is considered, and the channel variance is not included in the theoretical analysis. A max-rate (MR) scheduling algorithm was proposed for flow-level dynamic systems in [28]. The MR scheduling algorithm opportunistically schedules the user, which reaches its largest transmission rate in each time slot. To be throughput-optimal, the MR scheduling algorithm is either an offline algorithm that requires the prior knowledge of the channel and traffic distributions or an online one that needs a sufficiently long learning period. The design of the learning period is an open issue.

Other than queue-length-based algorithms, delay-based scheduling algorithms have also been investigated in the literature. McKeown *et al.* [41] introduced the MaxWeighted scheduling algorithm, in which the delay is used as the link weight. This algorithm has been extended in [23], [42], and [43] to provide the throughput-optimality for wireless networks. The delay-based scheduling algorithm for network utility maximization was studied in [44]. Based on the above works, Ji *et al.* [45] developed the delay-based back-pressure scheduling algorithm for the throughput-optimality in multihop wireless networks. Regarding head-of-line (HOL) delay-based scheduling, Andrews *et al.* [12] studied the HOL packet delay-based scheduling for systems only with persistent flows and discussed a framework for stable scheduling algorithms. The throughput-optimality of this scheduling algorithm was further investigated in [42]. However, none of these works considers to adopt the HAD-based scheduling in flow-level dynamic systems. What we also need to clarify here is that the HOL packet delay, based on which the existing works were developed, is calculated from the moment when the HOL packet arrives in the system, which is a different concept of the HAD in our work. The definition of the HAD is given in Definition 1, and the advantage of our work over the other HOL delay-based scheduling will also be explained later.

In this paper, we investigate the scheduling algorithm based on the HAD. Different from F-D-MW in [39], we give the proof of the throughput optimality without the assumption of the i.i.d. channel transmission rate. We also show that the last packet problem of the QMW scheduling algorithm can be solved with the proposed HAD-based scheduler.

III. System Models

We consider a time-slotted heterogeneous wireless network with one base station (BS) and multiple mobile users (MUs). Each MU can be associated with one or multiple short-lived flows. A short-lived flow is a traffic burst with a finite number of bits, and the flow size is defined as the size of traffic burst upon its arrival. Since the objective of our scheduling algorithm is the data flows, we use the term "flow" instead of "user" thereafter. Flows can enter the system at any time slot and will leave the system after all the bits are transmitted.

The system has multiple classes of flows, which make the system a heterogeneous network. Within each class, the flows have i.i.d. traffic arrival characteristic and the same channel rate profile. Assume that all the flows can be assorted into K classes. The ith flow of class-k at time t is denoted by $Q_{\mathrm{ki}}(t)$. The amount of the remaining bits of $Q_{\mathrm{ki}}(t)$ waiting for transmission at the beginning of time slot t is denoted by $|Q_{\mathrm{ki}}(t)|$ and called the residual bits. The number of flows of class-k at the beginning of time slot t is $N_k(t)$, and the total number of flows at the beginning of time slot t is $N(t) = \sum_{k=1}^{K} N_k(t)$. Within each class, the flows are indexed by their arrival time.

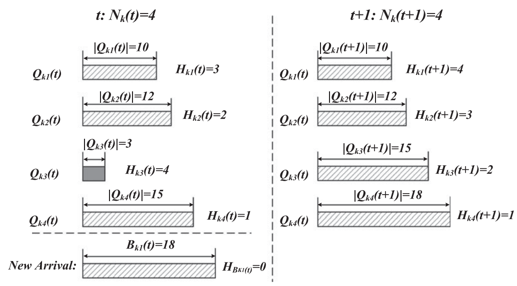


Fig. 1. System illustration of class-k flows from slot t to $t+1$.

A. Arrival Model

New flows can arrive at any time. Let $A_k(t) \in \{0\} \cup \mathbb{Z}_+$ denote the number of class-k flows arriving during time slot t, which is a random variable. $A_k(\cdot)$ is i.i.d. with the mean $\lambda_k = \mathbb{E}[A_k(1)]$. We suppose that the scheduling decision is made at the beginning of every time slot, and therefore, all the flows that arrive after the beginning of slot t can only be scheduled at the beginning of slot $t+1$. Let $B_{\mathrm{ki}}(t)$ denote the flow size of the ith class-k flow, which arrives during slot t. In class-k, we assume that $B_{\mathrm{ki}}(t)$ is the i.i.d. copy of some integer random variable B_k and has a finite mean $\beta_k = \mathbb{E}[B_k]$. The second moments of $A_k(\cdot)$ and $B_k(\cdot)$ are both finite. We define $|Q_k(t)| = \sum_{i=1}^{N_k(t)} |Q_{\mathrm{ki}}(t)|$ as the class-k backlog and $|Q(t)| = \sum_{k=1}^{K} |Q_k(t)|$ as the system backlog.

B. Channel Model

Let $r_{\mathrm{ki}}(t)$ denote the transmission rate of the wireless channel at time t between $Q_{\mathrm{ki}}(t)$ and the BS. The unit of the channel rate is bit/slot. The BS can transmit at most $r_{\mathrm{ki}}(t)$ bits at time t for $Q_{\mathrm{ki}}(t)$. $r_{\mathrm{ki}}(t)$ may vary over time as a result of fading. For class-k, we assume $r_{\mathrm{ki}}(\cdot)$ are i.i.d. copies of positive integer random variable R_k with finite supports, i.e., $R_k \in \{R_{k1}, R_{k2}, ..., R_{\mathrm{km}_k}\}$. Different classes may have heterogeneous channel condition distributions. The maximum possible transmission rate of the class-k flows is defined as $R_k^{\max} = \sup\{r : \mathbb{P}\{R_k = r\} > 0\}$, and the maximum possible transmission rate of the system is defined as $R^{\max} = \max_{1 \leqslant k \leqslant K} \{R_k^{\max}\}$.

An example of class-k dynamic flows of the network is illustrated in Fig. 1, which shows the evolution from time slot t to $t+1$. At the beginning of time slot t, there are four flows in the system. After that, there is one new flow, which has the HAD $H_{B_1(t)} = 0$. Suppose that $Q_{k3}(t)$ is scheduled at time slot t, and $r_{k3}(t) \geqslant |Q_{k3}(t)|$, $Q_{k3}(t)$ will finish all its transmission and leave the system, and we have four flows in class-k at the beginning of slot $t+1$.

C. System Capacity Region and Throughput-Optimality

Let γ_k represent the expected number of time slots that are required for the service of a class-k flow if served with $R_k^{\max}$, and we have $\gamma_k = \mathbb{E}\lceil \frac{B_k}{R_k^{\max}} \rceil$. Let $\rho_k = \lambda_k \gamma_k$ denote the traffic intensity of class-k flows, and $\rho = \sum_{k=1}^{K} \rho_k$ denote the system traffic intensity. The system capacity region is defined as $\mathcal{S} = \{(\lambda_1, \lambda_2, \ldots, \lambda_K), (\gamma_1, \gamma_2, \ldots, \gamma_K) : \rho < 1\}$. For any arrival process that lies in the capacity region, if the system is strongly stable, i.e., $\limsup_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}|Q(t)| < \infty$, then the corresponding scheduling algorithm is throughput-optimal. Intuitively, for a system working in slotted time, if the system is stable, the total amount of the queued data should be finite at any time slot, while if unstable, the total amount of the queued data will grow into infinity when $t \to \infty$ given infinite system buffer size. Considering that the physical meaning of the traffic intensity ρ is the average number of time slots that are required to transmit the arrival traffic in one time slot when the maximum possible transmission rate is always adopted, the sufficient condition for stability to be achievable is $\rho < 1$ [22]. In other words, if the average amount of arrival traffic in one time slot can be transmitted in less than one time slot by the maximum possible transmission rate, there exists at least one scheduling algorithm to achieve system stability. If $\rho > 1$, on average, more than one slot are required to transmit the amount of arrival data in one slot, and the residual data will accumulate into infinity over time, which results in instability. From this perspective, the system capacity region is defined as $\rho < 1$, and any arrival rate that is in the capacity region can be stably transmitted by the throughput-optimal algorithms.

If the system has no flow-level dynamics, i.e., the number of flows is fixed, we can use $|Q(t)|$ as the metric of the system stability. For the systems with flow-level dynamics, each flow has a finite amount of data to transmit upon arrival and leaves the system once all the data are transmitted. When the system stability is achieved, the number of flows in the system is finite, i.e., $N(t) < \infty$ at any time slot. If $N(t) \to \infty$ when $t \to \infty$, we have $|Q(t)| \to \infty$ as well, and thus, the system is unstable. As a result, we can also use $N(t)$ as the metric of the system stability when considering flow-level dynamics.

IV. Head-Of-Line Access Delay-Based Scheduling

Since we consider a delay-based scheduling algorithm, we give the definition of the HAD, which we will use in our scheduling.

Definition 1 (HAD $H_{\mathrm{ki}}(t)$): Let $I_{\mathrm{ki}}^H(t)$ denote the head bit in $Q_{\mathrm{ki}}(t)$, which will be the first bit to be transmitted. The HAD of $Q_{\mathrm{ki}}(t)$ is defined as $H_{\mathrm{ki}}(t) = t - t_0$, where t is the current time, and t_0 is the time at which $I_{\mathrm{ki}}^H(t)$ becomes the first bit in $Q_{\mathrm{ki}}(t)$.

HAD can be calculated according to the following equation:

$$H_{\mathrm{ki}}(t+1) = (H_{\mathrm{ki}}(t) + 1)\left(1 - \mathbb{1}_{\{Q_{\mathrm{ki}}(t)\}}(t)\right) \quad (1)$$

where $\mathbb{1}_{\{Q_{\mathrm{ki}}(t)\}}(t)$ is the indicator function such that $\mathbb{1}_{\{Q_{\mathrm{ki}}(t)\}}(t) = 1$ only when $Q_{\mathrm{ki}}(t)$ is scheduled at time slot t. With the system model and the definition of HAD in the above, we adopt the following HAD-based scheduling algorithm.

Algorithm 1: The HAD-based MaxWeight scheduling algorithm seeks the flow $\{k, i\}$ to transmit that satisfies the following condition at the beginning of time slot t:

$$\{k,i\}^*(H_{\mathrm{ki}}(t), r_{\mathrm{ki}}(t)) \in \underset{1 \le k \le K, 1 \le i \le N_k(t)}{\arg\max} H_{\mathrm{ki}}(t) \cdot r_{\mathrm{ki}}(t) \quad (2)$$

with uniform tie-breaking if there are a number of flows satisfying the condition. The scheduling decision is made in every time slot independently.

Next, we prove that, with flow-level dynamics, HAD scheduling is throughput-optimal, i.e., the system is stable with HAD scheduling, by three steps. In the first step, we explain that in an unstable system, there are countless flows that have infinite HADs, and we investigate a sufficient condition for stability. Second, we prove one property of the HAD scheduling algorithm when the system is unstable. Third, based on the above results, we further prove that a system with flow-level dynamics is stable when HAD is adopted, as long as the traffic intensity lies inside the system capacity region.

Theorem 1: Given an infinite buffer, for a flow-level dynamic multiuser wireless system, if the HAD scheduling algorithm cannot stabilize the system, there will be an infinite number of flows in the system that have infinite HADs when the system time goes to infinity.

Proof: Suppose that the system is unstable when $t \to \infty$ and then at least there is one flow with an infinite HAD. If there is only one flow that is unstable associated with the infinite HAD, and all the other flows have finite HADs, the only flow with the infinite HAD will be scheduled according to (2) at a certain time slot, for example t_1, and its HAD in the next time slot $t_1 + 1$ is 0. Because we have the assumption here that all the other flows have finite HAD, we can come to the conclusion that at time slot $t_1 + 1$, all the flows have finite HADs. This conclusion is contradicted with the instability condition that there is at least one flow with an infinite HAD.

Similarly, we can prove that if there are only a finite number of flows associated with infinite HADs, the system is also stable. Finally, we can conclude that if the system is unstable, an infinite number of flows in the system must have infinite HADs when $t \to \infty$. ■

Theorem 2 (Sufficient condition): Let $r(t)$ denote the real transmission rate of the network at time t. If a class-k flow $Q_{\rm ki}(t)$ is scheduled, i.e., $r(t) = r_{\rm ki}(t)$, the sufficient condition for the network with flow-level dynamics to be stable for any arrival rate that lies in the capacity region is

$$\lim_{t\to\infty} \mathbb{P}\{r(t) < R_k^{\max}\} = 0. \tag{3}$$

Proof: We use the following Lyapunov function $L(t) = (W(t))^2$ to prove Theorem 2, where $W(t)$ is defined as the workload of the system at time t, i.e., $W(t) = \sum_{k=1}^{K}\sum_{i=1}^{N_k(t)} \left\lceil \frac{|Q_{\rm ki}(t)|}{R_k^{\max}} \right\rceil$. The workload is apparently a direct reflection of the total queue length of the system. We define $W_A(t) = \sum_{k=1}^{K} \sum_{i=1}^{A_k(t)} \lceil \frac{B_{\rm ki}(t)}{R_k^{\max}} \rceil$ as the amount of the new workload injected in the network at time t, and $W_R(t) = \sum_{k=1}^{K}\sum_{i=1}^{N_k(t)} \lceil \frac{r_{\rm ki}(t)}{R_k^{\max}} \rceil \cdot \mathbb{1}_{\{Q_{\rm ki}(t)\}}(t)$ as the decrease of the workload if $Q_{\rm ki}(t)$ is scheduled for transmission at time t, i.e., $r(t) = r_{\rm ki}(t)$, where $\mathbb{1}_{\{Q_{\rm ki}(t)\}}(t) = 1$ if $Q_{\rm ki}(t)$ is scheduled, and $\mathbb{1}_{\{Q_{\rm ki}(t)\}}(t) = 0$ otherwise. Based on the above notations, the evolution of the workload in the system can be described as $W(t+1) = [W(t) + W_A(T) - W_R(t)]^+$. Then, we calculate the square of this equation, and after some manipulation, we can obtain

$$\begin{aligned}(W(t+1))^2 &- (W(t))^2\\ &\leqslant (W_A(t))^2 + (W_R(t))^2 - 2W(t)W_R(t)\\ &\quad + 2W_A(t)\,(W(t) - W_R(t))\\ &\leqslant (W_A(t))^2 + (W_R(t))^2 - 2W(t)\,(W_R(t) - W_A(t)). \end{aligned} \tag{4}$$

Since the arrival rates lie in the capacity region, and the second moments of the arrival rates are bounded, we can conclude that there exists a $U = \mathbb{E}[(W_A(t))^2] + \mathbb{E}[(W_R(t))^2] < \infty$. By taking the expectation of (4), we can calculate the Lyapunov drift of the Lyapunov function as follows:

$$\begin{aligned}\mathbb{E}[(W(t+1))^2] &- \mathbb{E}[(W(t))^2]\\ &\leqslant U - 2\mathbb{E}[W(t)\,(W_R(t) - W_A(t))].\end{aligned}$$

The above holds for all $t \in \{0, 1, 2, ...\}$. Summing over $t \in \{0, 1, \ldots, T-1\}$ for some integer $T > 0$ yields

$$\begin{aligned}\mathbb{E}[L(W(T))] &- \mathbb{E}[L(W(0))]\\ &\leqslant U \cdot T - \sum_{t=0}^{T} \mathbb{E}W(t)\mathbb{E}\,(W_R(t) - W_A(t)).\end{aligned}$$

Note that $\mathbb{E}[L(W(T))] \geqslant 0$, taking a lim sup yields

$$\begin{aligned}&\limsup_{T\to\infty} \frac{1}{T\varepsilon} \sum_{t=0}^{T} \left(\sum_{k=1}^{K} \sum_{i=1}^{N_k(t)} \mathbb{E}[W(t)] \right) \cdot\\ &\times \left(\sum_{k=1}^{K} \sum_{i=1}^{N_k(t)} \left\lceil \frac{r_{\rm ki}(t)}{R_k^{\max}} \right\rceil \cdot \mathbb{1}_{\{Q_{\rm ki}(t)\}}(t) - \sum_{k=1}^{K}\sum_{i=1}^{A_k(t)} \frac{B_{\rm ki}(t)}{R_k^{\max}} \right)\\ &\leqslant \limsup_{T\to\infty} \frac{E[L\,(W\,(0))]}{T\varepsilon} + \frac{U}{\varepsilon}.\end{aligned}$$

From the definition of the capacity region in the previous section, we have

$$\lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T}\sum_{k=1}^{K}\sum_{i=1}^{N_k(t)} \mathbb{E}\,[B_{\rm ki}(t)/R_k^{\max}] = 1 - \varepsilon.$$

Noting that

$$\lim_{t\to\infty} \mathbb{E}\lceil r_{\rm ki}(t)/R_k^{\max}\rceil = 1$$

as well as that $\mathbb{E}[L\,(W\,(0))]$ is bounded, we have

$$\limsup_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[W(t)] < \infty$$

which indicates that the total queue length in the system is bounded, and hence, the system is stable. ■

The intuitive explanation of the above theorem is as follows. If the scheduling algorithm always tries to schedule a flow when it has its possible maximum transmission rate, the system is stable, thanks to the efficient utilization of resource. From the definition of the capacity region (in Section III), we can say that if a flow is scheduled when it is not in its maximum transmission channel rate, it probably needs more time slots for transmission and, hence, leads to waste of resource. However, the above is not a necessary condition for a system to be stable. For example, if there is a large gap between the arrival rate vector and the capacity region, i.e., the traffic intensity of the system is quite low, it is possible that the system is able to deliver all the arrival bits, though some transmission is associated with a low transmission rate. For a network with a very high traffic intensity, i.e., there is a very small gap between the arrival vector and the system capacity, however, the condition in (3) is almost a necessary condition.

Lemma 1: Given infinite buffer, for a single-class ($K = 1$) flow-level dynamic multiuser wireless system with the HAD scheduling algorithm, as in (2), if the system is unstable, we have

$$\lim_{t\to\infty} \mathbb{P}\{r(t) < R_1^{\max}\} = 0. \quad (5)$$

Proof: In a homogeneous system, without loss of generality, we suppose $Q_1(t)$ has the maximum HAD in the system at time t, i.e., $H_1(t) = H^{\max}(t)$. For the simplicity of presentation, suppose that $r_i(t)$ are copies of a positive integer random variable $R \in \{R_1, R_2\}$ and $R_1 < R_2$. One can extend the proof to the multirate case with the same approach.

Let $\mathscr{U}(t)$ denote the set of flows, in which all the flows have the HAD larger than $(H_1(t) \cdot R_1/R_2)$ at time t, and let $\tilde{N}(t)$ denote the number of flows in $\mathscr{U}(t)$. From (2), we know that the probability $\mathbb{P}\{r(t) < R^{\max}\}$ is equivalent to the probability of the event that $\forall Q_i(t) \in \mathscr{U}(t)$ are associated with R_1 at time t. Denoting $p = \mathbb{P}\{r_i(t) = R_1\}$, we have

$$\mathbb{P}\{r(t) < R^{\max}\} = p^{\tilde{N}(t)}. \quad (6)$$

Next, we prove that $\lim_{t\to\infty} \tilde{N}(t) \to \infty$.

Suppose that we can find a positive integer $\mathscr{M}$ such that $\tilde{N}(t) < \mathscr{M}$ for all the time t. If $r_1(t) = R_2$, $Q_1(t)$ will be scheduled, so the probability for $\{Q_i(t) : i = \arg\max_{1\leqslant i\leqslant N(t)} H_i(t)\}$ to be scheduled is larger than $1 - p$ for any time slot. Since $\mathscr{M}$ and p are not time coupled, we can find a positive integer $\mathscr{T}$, which is also irrelevant with time, and for any arbitrarily small positive ε, the probability that $\forall Q_i \in \mathscr{U}(t)$ can be scheduled within $\mathscr{T}$ slots is larger than $1 - \varepsilon$. In other words, the maximum HAD at time $t + \mathscr{T}$, denoted by $H_1(t + \mathscr{T})$, is smaller than $(H_1(t)R_1/R_2) + \mathscr{T}$ with a probability larger than $1 - \varepsilon$. Since we can find a positive value $\mathscr{H}$ such that we have $\mathbb{P}\{\frac{H_1(t)R_1}{R_2} + \mathscr{T} < H_1(t)\} > 1 - \varepsilon$ when $H_1(t) > \mathscr{H}$, so we have $\mathbb{P}\{H_1(t + \mathscr{T}) < H_1(t)\} > (1 - \varepsilon)^2$. This indicates that when the maximum HAD is larger than $\mathscr{H}$, and after $\mathscr{T}$ slots, the maximum HAD is not likely to increase beyond and bounded by $\mathscr{H}$. This conclusion contradicts with the system instability that we discuss here. Therefore, the assumption that we can find $\mathscr{M}$ such that $\tilde{N}(t) < \mathscr{M}$ for all the time t is not true, which leads to $\lim_{t\to\infty} \tilde{N}(t) \to \infty$. Consequently, from (6), we have $\lim_{t\to\infty} \mathbb{P}\{r(t) < R_k^{\max}\} = 0$. ■

Theorem 3: Given infinite buffer, for a flow-level dynamic multiuser wireless system with the HAD scheduling algorithm as in (2), if the arrival rates lie in the capacity region, the system is stable when $K \geqslant 1$.

Proof: First, we consider the case that $K = 1$. Based on Lemma 1, we have $\lim_{t\to\infty} \mathbb{P}\{r(t) < R_k^{\max}\} = 0$. Since the arrival rates lie in the capacity region, based on Theorem 2, we have the conclusion that the system is stable.

Now, consider a heterogeneous system where $K > 1$. Suppose this system is unstable with the HAD scheduling. Without loss of generality, we assume $R_{k1} < R_{k2} < \cdots < R_{k(m_k-1)} < R_{\mathrm{km}_k}$ if the class-k flows have m_k rates.

Suppose that class-1 is the unstable class, i.e., $N_1(t) \to \infty$ when $t \to \infty$. Similar to the proof of Theorem 1, we can have the conclusion that if $\exists i \in \{1, 2, \ldots, K\}$, such that $N_i(t) \to \infty$, then $\forall k \in \{1, 2, \ldots, K\}$, we have $N_k(t) \to \infty$, i.e., if one class is unstable, then all the classes are unstable. If a flow in class-1 is scheduled, from the proof of Lemma 1, we can directly come to $\lim_{t\to\infty} \mathbb{P}\{r_{1i}(t) < R_{1m_k}\} = 0$ if $r(t) = r_{1i}(t)$. This conclusion can be extended to a more general one that for any class-k in the system, if it is unstable, $\mathbb{P}\{r_{\mathrm{ki}}(t) < R_{1m_k}\} = 0$ if $r(t) = r_{\mathrm{ki}}(t)$. However, from Theorem 2, the system is stable. This is a contradiction to the assumption of instability, and hence, this assumption is not true. Combined with the case where $K = 1$, we have Theorem 3 proved. ■

With the above analysis, we can draw the conclusion that HAD can stabilize the systems when the number of flows is not fixed as long as the arrival rate lies in the system capacity region, so it is throughput-optimal for the systems with flow-level dynamics. Note that essentially the traffic arrival characteristics do not influence the algorithm's throughput-optimality through the analysis above. In the system model of our work, however, because the definition of the system capacity region is related to both of the traffic and the channel profile, we put the flows that have the same traffic arrival characteristic and the same channel profile into one class just for the convenience of the presentation.

HAD scheduling has several advantages compared with the existing scheduling algorithms. First, in the MR scheduling, either the prerequisite of channel condition distribution is required or a learning period is necessary to learn the possible maximum channel rate, which has an unknown influence on the system performance, while HAD scheduling is an online scheduling and its decision-making process is simple. It is more practical in situations where the channel distribution may not be available in advance. Second, in F-D-MW, the necessary condition in the proof of stability [39] is that all the flows have i.i.d. channel condition, or at least the maximum channel rates among all the flows are identical, which narrows the utilization, while in HAD, heterogeneous channel condition distributions of different classes are supportable. Even when the $R_K^{\max}$ is different among the K classes, HAD is still able to stabilize the system according to Theorem 3, and in F-D-MW, the flows that come into the system earlier will always have a higher priority to win the chance for transmission than the flows that enter the system later, so that the new flows may suffer long start-up latency, while in HAD scheduling, the new flows can have more opportunities to be served. Last but not least, in the F-D-MW scheduling algorithm, each flow has to record the delay for every packet, while HAD only needs a simple counter for the HAD; thus, the overhead is reduced.

The implementation of our proposed HAD scheduler is similar to that of the classic QMW scheduler, and HAD does not bring more signaling overhead than the widely adopted

PF and the other online throughput-optimal scheduling algorithms such as QMW and F-D-MW. In the existing works, the MaxWeight type of scheduler has been implemented and tested, e.g., by the work from the Bell Laboratories, Alcatel-Lucent, in 2011 [46]. Moreover, according to [16], different types of the MaxWeight scheduling components have already been adopted and implemented in practice, e.g., data center bridging by Cisco [47] and Qualcomm's Flashlinq peer-to-peer wireless networks [48].

V. Queueing Behavior Analysis

In the above section, we proved that HAD is able to stabilize the system with flow-level dynamics. We know that the number of flows $N(t)$ and the total backlog $|Q(t)|$ in the system are bounded with HAD, and it is important to investigate how much $N(t)$ and $\sum B(t)$ will be when they converge, as well as the delay performance of HAD. In this section, we focus on the theoretical analysis of the system performance of HAD scheduling, including the expectation of $N(t)$ and the delay performance.

As shown in [49], HAD is able to achieve a certain level of fairness between flows. Generally speaking, other throughput-optimal scheduling algorithms tend to let one (or a few) flows(s) exclusively occupy the channel resource for a long time, while by adopting HAD, flows in the system are able to fairly share the transmission opportunities. Although TCP flows are considered in [49], the fairness can be achieved when scheduling other non-TCP controlled flows. We here focus on the performance study of HAD by the approach of queueing theory.

For simplicity, we investigate a homogeneous network with one class of flows and assume that the arrival rate satisfies $A(t) \leqslant 1$. Let $H_i^{\text{sch}}(t)$ denote the HAD when $Q_i(t)$ is scheduled. If we focus on homogeneous networks, it is shown that if all the flows have i.i.d. channel rate distribution, i.e., $\mathbb{E}[r_i(t)] = \mathbb{E}[r_j(t)]$ and $r_i^{\max} = r_j^{\max}$, we have $\mathbb{E}[H_i^{sch}(t)] = \mathbb{E}[H_j^{sch}(t)]$ [49]. Considering that $P\{r(t) = R^{\max}\} = 1$ if HAD is adopted when ρ is large enough, we have

$$\mathbb{E}[H_i^{sch}] = \mathbb{E}[N(t)]. \tag{7}$$

Assume that the length of the time slot is δ, and hence, the maximum transmission rate in one time slot is $\delta R^{\max}$. Let $\mathbb{E}[B_i(t)] = \bar{B}$ denote the mean value of the initial queue length of the flows if $B_i(t)$ is a random variable with a finite second-order moment. Let $T_i^{tx}(t)$ denote the number of time slots to finish the transmission for $Q_i(t)$, which has a finite amount of data to transmit, and the expectation of $T_i^{tx}(t)$ is

$$\mathbb{E}[T_i^{tx}(t)] = \mathbb{E}[H_i^{\max}] \cdot \left\lceil \frac{\bar{B}}{\delta R^{\max}} \right\rceil = \mathbb{E}[N(t)] \cdot \left\lceil \frac{\bar{B}}{\delta R^{\max}} \right\rceil. \tag{8}$$

The average throughput of $Q_i(t)$, defined as $\mathbb{E}\{W_i(t)\}$, can be calculated by

$$\mathbb{E}[W_i(t)] = \frac{R^{\max}}{\mathbb{E}[N(t)]}. \tag{9}$$

From the above analysis, we can see that the average time that one dynamic flow stays in the system and the throughput of each flow are related to the parameters including the initial queue length, the maximum channel rate, the time slot duration, and the average number of flows in the system. Next, we will figure out how to calculate the average number of flows in the system in order to analyze the delay and throughput performance. In the following analysis, we assume that $\delta = 1$ time unit, and thus, δ can be omitted in the equations.

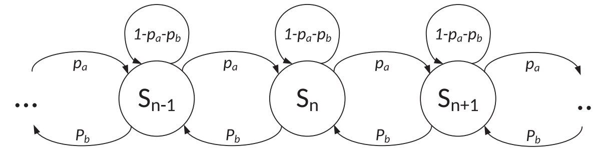


Fig. 2. Markov chain of $N(t)$.

A. State-Dependent Markov Model

In a homogeneous wireless network working in slotted time with flow-level dynamics, if HAD scheduling algorithm is adopted, the number of flows in the system in one time slot can be described by a discrete-time Markov chain as a queueing system, which is shown in Fig. 2. The state S_i represents that the number of flows in the system in a time slot is i, and $S_i = S_{i-1} + 1$. Suppose that in time slot t, we have $N(t) = S_i$. From t to $t+1$, $N(t)$ is possible to change from S_i to S_{i-1} or S_{i+1}, or stay in the state of S_i, depending on if there is a new flow's arrival or an old flow's departure. We define the transit probability as $p_a = P\{N(t+1) = N(t) + 1\}$, $p_b = P\{N(t+1) = N(t) - 1\}$, and $p_c = P\{N(t+1) = N(t)\} = 1 - p_a - p_b$.

The scheduler behaves differently, depending on whether the traffic intensity ρ is small or big. When ρ is small, the scheduler is able to stabilize the system without holding the condition in (3) to be always true. When ρ is close to 1, the condition in (3) needs to be satisfied to achieve system stability. Thus, the average scheduled transmission rate and the transit probability of the Markov process are dependent on the state of the system. We use $R^{\text{avg}}(n)$ to denote the average scheduled transmission rate when the system is in state S_n; the state transition probability can be calculated as follows:

$$\begin{cases} p_1 = P\{\text{one arrival}\} \\ p_{\text{ai}} = p_1 \left(1 - \dfrac{1}{\bar{w}(i)}\right) \\ p_{\text{bi}} = (1 - p_1) \dfrac{1}{\bar{w}(i)} \\ p_{ci} = (1 - p_{\text{ai}} - p_{\text{bi}}). \end{cases} \tag{10}$$

In (10), $\bar{w}(i)$ is the state-dependent average work load of each flow in terms of the necessary time slots for the transmission of the whole flow and can be calculated as $\bar{w}(i) = \lceil \bar{B}/R^{\text{avg}}(i) \rceil$. We consider how to obtain R^{avg} of a network, in which all the nodes have a homogeneous channel profile. As mentioned, for each flow, the transmission rate can be chosen from the channel rate set $\mathscr{R}$, which has finite supports, i.e., $\mathscr{R} = \{R_1, R_2, \ldots, R_m\}$, where m is a positive integer. The maximum channel rate in $\mathscr{R}$ is denoted as $R^{\max}$. For simplicity, we consider $m = 2$ and $R_1 < R_2$ and define $R^{slow} = R_1$. When $m > 2$, we use the following approximation in the analysis:

$$R^{\text{slow}} = \left\{ \frac{1}{m-1} \sum_i R_i | R_i \in \mathscr{R}, R_i \neq R^{\max} \right\}.$$

Let $\mathscr{U}(t)$ denote the set of flows, in which all the flows have the HAD larger than that of the rest of the other flows in the system at time t. We use $M(i)$ to denote the number of flows in $\mathscr{U}(t)$, i.e., $|\mathscr{U}(t)| = M(i)$ when the system is in state S_i. Since the flows with larger HAD have the priority to be scheduled in HAD, the scheduler will choose a flow with R^{slow} at time slot t when all the M flows in $\mathscr{U}(t)$ are in the channel state R^{slow}, where $M(i)$ satisfies the following inequality considering (7):

$$R^{\text{slow}} \cdot i \geqslant R^{\max} \cdot (i - M(i)).$$

Furthermore, we have $M(i) \geqslant \lceil i(1 - \frac{R^{\text{slow}}}{R^{\max}}) \rceil$ in the time-slotted system, and thus, we use

$$M(i) \approx i \left(1 - \frac{R^{\text{slow}}}{R^{\max}}\right) \tag{11}$$

in our analysis.

The number of flows in the system is changing along with the variance of the traffic intensity ρ. With the calculation of $M(i)$, the average channel rate of the system in state S_i can be calculated as follows:

$$\begin{aligned} R^{\text{avg}} &= (1 - P\{r_i(t) = R^{\max}\})^{M(i)} \cdot R^{\text{slow}} \\ &\quad + [1 - (1 - P\{r_i(t) = R^{\max}\})^{M(i)}] \cdot R^{\max}. \end{aligned} \tag{12}$$

The difference equations of (10) are as follows:

$$\begin{cases} P_i(p_{\text{ai}} + p_{\text{bi}}) = P_{i+1} \cdot p_{b(i+1)} + P_{i-1} \cdot p_{\text{ai}} \\ P_0 p_a = P_1 p_{b1}. \end{cases} \tag{13}$$

The solution of (13) is

$$P_0 = \frac{1}{1 + \sum_{j \geqslant 1} \frac{\Pi_{i=0}^{j-1} p_{\text{ai}}}{\Pi_{i=0}^{j} p_{\text{bi}}}} \tag{14}$$

and

$$\begin{aligned} P_n &= \frac{\Pi_{i=1}^{n-1} p_{\text{ai}}}{\Pi_{i=1}^{n} p_{\text{bi}}} P_0 = \frac{p_1^{n-1} \Pi_{i=1}^{n-1} 1 - \frac{1}{\bar{w}(i)}}{(1-p_1)^n \Pi_{i=1}^{n} \frac{1}{\bar{w}(i)}} P_0 \\ &= \frac{p_1^{n-1} \Pi_{i=1}^{n-1} \frac{\bar{B} - R^{\text{avg}}(i)}{\bar{B}}}{(1-p_1)^n \Pi_{i=1}^{n} \frac{R^{\text{avg}}(i)}{\bar{B}}} P_0. \end{aligned} \tag{15}$$

The number of flows in the system can be calculated by $\mathbb{E}[N(t)] = \sum_i i \cdot P_i$ with the definition of $\bar{w}(i)$, (14) and (15). However, this method is very complicated. We further developed an approximation method to analyze the system.

B. Approximation

First, we analyze the scenario when ρ is small such that (3) is not necessary for system stability. In this case, the system can also be described, as shown in Fig. 2, of which the transit probability and the balance equations can be found as

$$\begin{cases} \bar{w} = \lceil \bar{B}/R^{\text{avg}} \rceil \\ p_1 = P\{\text{one arrival}\} \\ p_a = p_1 \left(1 - \frac{1}{\bar{w}}\right) \\ p_b = (1 - p_1) \frac{1}{\bar{w}} \\ p_c = (1 - p_a - p_b) \\ P_n(p_a + p_b) = P_{n-1} p_a + P_{n+1} p_b \\ P_0 p_a = P_1 p_b. \end{cases} \tag{16}$$

In (16), $\bar{w}$ and R^{avg} are defined the same as that in (10). Different from the analysis with the state-dependent Markov process, here, we let M denote the average number of flows in $\mathscr{U}(t)$, i.e., $\mathbb{E}\{|\mathscr{U}(t)|\} = M$, and thus, (11) becomes

$$M \approx \left\lceil \mathbb{E}\{N(t)\} \left(1 - \frac{R^{\text{slow}}}{R^{\max}}\right) \right\rceil \tag{17}$$

and (12) becomes

$$\begin{aligned} R^{\text{avg}} &= (1 - P\{r_i(t) = R^{\max}\})^M \cdot R^{\text{slow}} \\ &\quad + [1 - (1 - P\{r_i(t) = R^{\max}\})^M] \cdot R^{\max}. \end{aligned} \tag{18}$$

Using the average channel rate to build the Markov chain as described in (16), the solution can be found as

$$\begin{cases} P_0 = 1 - \frac{p_a}{p_b} \\ P_n = \left(1 - \frac{p_a}{p_b}\right) \left(\frac{p_a}{p_b}\right)^n \\ E[N] = \sum_{n=0}^{\infty} n P_n = \frac{p_a}{p_b - p_a} = p_1 \frac{1 - \frac{1}{\bar{w}}}{\frac{1}{\bar{w}} - p_1} \\ E[H^{\max}] = E[N] \\ p_{\text{ki}} = \frac{R_k^{\max}}{\sum_k N_k R_k^{\max}} \\ S_{\text{ki}} = \frac{(R_k^{\max})^2}{\sum_k N_k R_k^{\max}} \\ E[T_i^{tx}] = E[N] \bar{w}. \end{cases} \tag{19}$$

We refer to this analytical model as the M/M/1 approximation. Since (17) is involved in (19) to find $\mathbb{E}\{N(t)\}$, which brings extra computational complexity, we consider to further simplify the analysis by proposing the M/M/1-M approximation with two iterations of the M/M/1 approximation. In M/M/1-M, we solve the M/M/1 model for two times. The result of the first time is used in the second time. We first solve (16) by setting $\bar{w} = \lceil \bar{B}/R^{\max} \rceil$ to obtain an $\mathbb{E}[N(t)]$ in the first iteration and, then, plug it in (17) to obtain the M/M/1-M approximation results by solving (16) again. Although we have two iterations in the M/M/1-M approximation, the computational complexity is reduced, since we can avoid the exponential (or logarithm) computation in the M/M/1 approximation. However, the accuracy of the M/M/1-M approximation may be compromised, which is explained as follows. Since $N(t)$ is conservatively approximated with $\bar{w} = \lceil \bar{B}/R^{\max} \rceil$ in the first iteration, the average transmission rate is likely to be reduced in the second iteration, and as a result, $N(t)$ of the M/M/1-M approximation may be slightly larger than the actual situation.

Next, we consider the scenario when ρ is close to 1. In this case, since (3) becomes necessary for stability, we can use an M/D/1 Markov model to calculate, where

$$\bar{w} = \lceil \bar{B}/R^{\max} \rceil.$$

Hence, the average number of flows in the system can be obtained as follows:

$$E[N] = \frac{\left(\frac{2p_a}{p_b} - \left(\frac{p_a}{p_b} \right)^2 \right)}{2 \left(1 - \frac{p_a}{p_b} \right)}. \tag{20}$$

In this case, we refer the analytical model as M/D/1 approximation. By substituting the results in (19) and (20) into (8) and (9), we are able to obtain the desired performance results.

The difference between M/M/1-M approximation and M/D/1 approximation is that in the M/M/1-M approximation, the probability that each queue is served in the maximum channel rate is not 1. As a result, the service time of each flow is random, while in the M/D/1 approximation, the serving time of each flow is deterministic. This implies that we can treat the result of the M/M/1-M approximation for a relatively small ρ, while taking the result of M/D/1 approximation as the lower bound for the performance of HAD. Our analysis results can also be treated as a reference of the other throughput-optimal scheduling algorithms, considering that HAD performs similarly to the other throughput-optimal algorithms.

VI. Simulation Results

In this section, we evaluate the performance of HAD scheduling along with the other scheduling algorithms, including QMW [9], F-D-MW [39], MR [28], and PF [29].

A. Throughput-Optimality

In the simulation, we have two classes of short-lived flows. The traffic burst size of the class-1 and class-2 flows is 30 (units) and 60 (units), respectively. We adopted Good–Bad channel model, i.e., each class has two transmission rates. The channel rate for class-1 flows is $R_1 = \{9, 10\}$ (units/slot), and $\mathbb{P}\{R_1 = 9\} = 0.1$, $\mathbb{P}\{R_1 = 10\} = 0.9$, while $R_2 = \{16, 20\}$ (units/slot), with $\mathbb{P}\{R_2 = 16\} = 0.2$, $\mathbb{P}\{R_2 = 20\} = 0.8$. The arrival probability is calculated according to the traffic intensity ρ. The history rate window size for PF is 1000 time slots [29]. The simulation tool is MATLAB. With traffic intensity $\rho = 0.999$, the throughput-optimality of HAD scheduling is compared with the other algorithms, as illustrated in Figs. 3–5.

The results shown in Fig. 3 are the evolution of the number of flows $N(t)$ in the system with y-axis in the logarithmic form. We can observe that $N(t)$ of QMW and PF increases with time, and from the increasing trend, we can tell that the system cannot be stabilized with either the QMW or the PF scheduling. While with the other three scheduling algorithms, $N(t)$ is bounded and the system can be stabilized. We can observe that $N(t)$ with HAD is slightly larger than that with MR, and F-D-MW has the smallest $N(t)$, which is the result of using flow delay as the scheduling weight so that the old flows in the system have more chances to transmit. By allowing more flows to coexist in the system, HAD scheduling may allocate more resources to the newer flows and, hence, can achieve a lower start-up latency and a better fairness between the old and new flows.

Fig. 4 shows the evolution of the system backlog $|Q(t)|$ with y-axis in the logarithmic form. Similar to Fig. 3, we can observe that $|Q(t)|$ with the QMW and PF scheduling algorithm keeps increasing with time, and $|Q(t)|$ with the other three algorithms is bounded and is almost identical. The same conclusion can be drawn that the system can be stabilized by all the scheduling algorithms except QMW and PF.

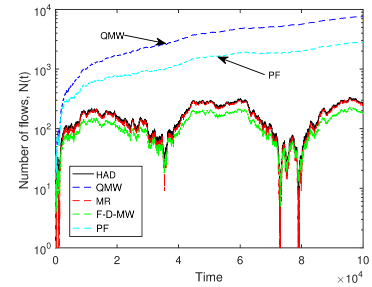


Fig. 3. Number of flows $N(t)$ with $\rho = 0.999$.

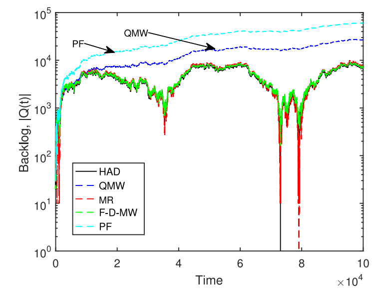


Fig. 4. System backlog $|Q(t)|$ with $\rho = 0.999$.

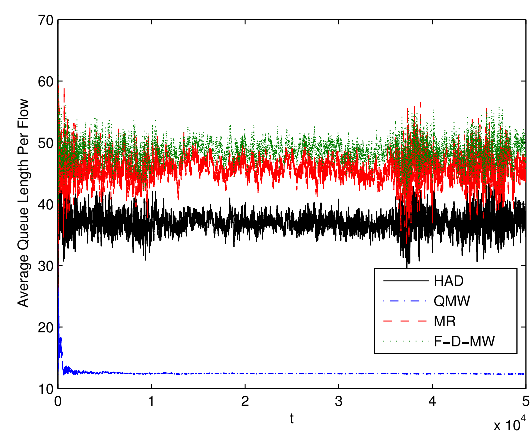


Fig. 5. Average queue length with $\rho = 0.999$.

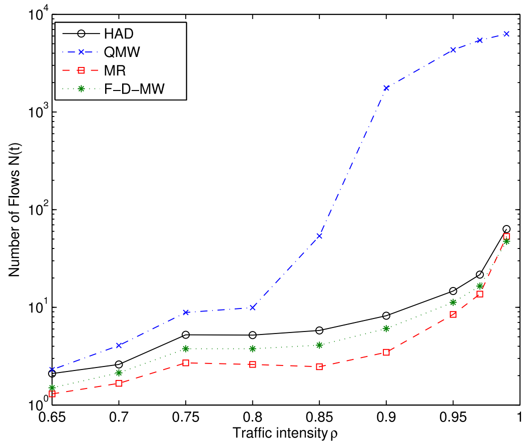


Fig. 6. Number of flows $N(t)$ with different ρ at $t = 50\,000$.

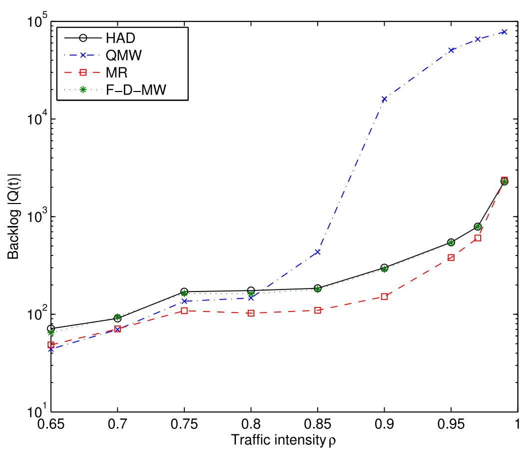


Fig. 7. Backlog $|Q|$ with different ρ at $t = 50\,000$.

The evolution of the average queue length per flow, which is defined as $\bar{Q}(t) = |Q(t)|/N(t)$, is illustrated in Fig. 5. The QMW scheduling has the smallest $\bar{Q}(t)$, while the other three algorithms all have a larger $\bar{Q}(t)$. The reason is that the QMW scheduling computes the weight of each flow proportional to the individual queue length. Thus, once a flow has only a small amount of data left for transmission, it will have a small weight during the scheduling process, which results in two consequences: 1) There will be an increasing number of flows accumulated in the system, as shown in Fig. 3; and 2) these large number of flows with the small number of tail bits will hardly get a chance for transmission, which makes the average queue length maintained at a low level, while for the other three throughput optimal scheduling algorithms, $N(t)$ is kept to be very low, and the average queue length (mainly associated with the new arriving flows) is relatively high. It can also be observed that the $\bar{Q}(t)$ of F-D-MW is the largest, and the $\bar{Q}(t)$ of HAD is the smallest. As the system backlog is almost identical for all these three algorithms, $\bar{Q}(t)$ has an inverse relationship with $N(t)$.

In Figs. 6 and 7, we compare the performance of HAD scheduling and the other three scheduling algorithms with the traffic intensity varying from 0.65 to 0.999. The x-axis is the traffic intensity ρ, which is defined in the system model. Figs. 3 and 4 show the evolution of the system $N(t)$ and $|Q(t)|$ when $\rho = 0.999$, while here, we take the snapshots of the evolution of $N(t)$ and $|Q(t)|$ with ρ increasing from 0.65 to 0.999 at the moment of $t = 50\,000$. The results are averaged over ten simulations. When $\rho \leqslant 0.8$, all of the four algorithms are stable because of the low traffic intensity. Because the weight is proportional to the queue length, QMW has a good performance on system backlog $|Q(t)|$, although it has more flows in the system, as we can observe in Fig. 6. It is noticeable in Fig. 6 that, when $\rho \geqslant 0.8$, the number of flows of the QMW scheduling algorithm in the system grows fast, and the same trend on the system backlog $|Q(t)|$ can also be observed in Fig. 7, while the $N(t)$ and $|Q(t)|$ of HAD and the other two algorithms only experience a slow increase. When ρ is small, the MR scheduling has the best performance, thanks to the full knowledge of the system channel information, while the performance of the three throughput optimal algorithms tends to converge when ρ approaches 1.

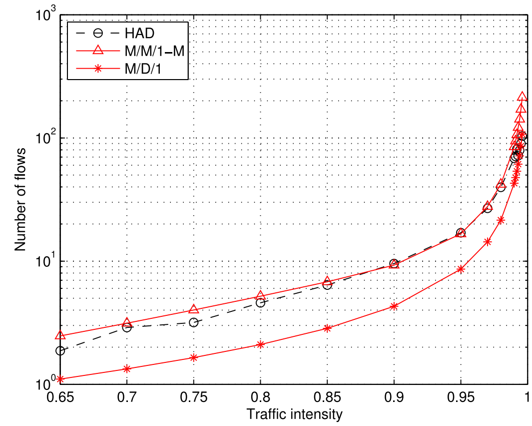


Fig. 8. Number of flows in the system with varying ρ.

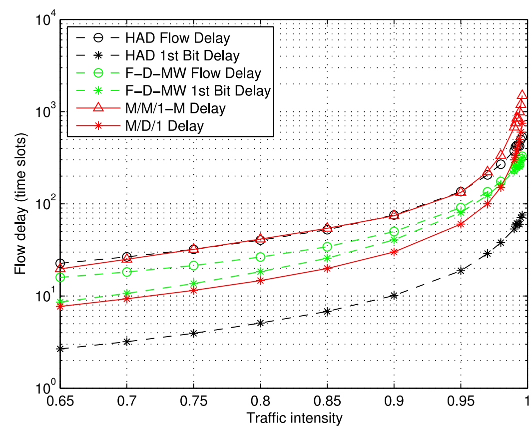


Fig. 9. Average queue delay with varying ρ.

B. Number of Flows and Flow Delay

To verify the analysis of HAD in terms of $N(t)$ and $T_i^{tx}(t)$ in Section V, we compared the analytical and simulation results, which are shown in Figs. 8 and 9. In our work, we only adopt M/M/1-M and M/D/1 approximations. In this simulation, we have one class of flows in the system and further increased the traffic burst size to 70 units.

In Fig. 8, the M/M/1-M and M/D/1 approximation result for $N(t)$ is shown as the red curve marked with upward-pointing triangles and stars, respectively. When ρ is small, the M/M/1-M approximation is very close to $N(t)$ of HAD, while the M/D/1 approximation is shown as the lower bound. When ρ is close to 1, the M/D/1 approximation can better describe the system behavior.

The delay performance is shown in Fig. 9. Before $\rho \leqslant 0.97$, the M/M/1-M approximation result for $T_i^{tx}(t)$ can accurately describe the flow sojourn time in the system, which is getting increasingly closer to the M/D/1 approximation when ρ increases from 0.97 to 1. This result verifies our previous analysis. We also included the simulation results of the flow delay of F-D-MW, which lies in between the M/M/1-M and M/D/1 approximation results for most of the situations except when ρ is extremely large. In this figure, the first-bit delay is the waiting time between the moments of the entrance a flow in the system and the first schedule of this flow, which is referred as the start-up latency. From the simulation, we can also observe that HAD has a much shorter start-up latency, which is the key to support real-time data, while the flow delay of HAD is only marginally larger than that of F-D-MW.

From all the above simulation results, we can observe that the HAD scheduling algorithm is not only able to maintain the stability and throughput-optimality with flow-level dynamics in a heterogeneous system, but provides better fairness among flows as well, which is a desirable feature. With HAD, a new flow in the system does not have to wait for a long time before the first transmission, while with the F-D-MW scheduling, the first few packets in a new flow have to wait for a long time to be transmitted, which may result in a large start-up latency. The fact that HAD requires no prior knowledge of the arrival process, and the channel rate distribution makes it easier to implement.

VII. Conclusion

We have proposed and studied the HAD scheduling algorithm in multiuser wireless networks with flow-level dynamics. A sufficient condition of the system queueing stability for short-lived flows has been provided, based on which we have further investigated the throughput-optimality of the proposed HAD algorithm. HAD is an online algorithm that requires no prior knowledge of the statistics of the data arrival nor the channel state information, and hence, it is practical and simpler to implement compared with other throughput-optimal scheduling algorithms for flow-level dynamic systems, such as MR and F-D-MW. We have also built a Markov analytic model to study the proposed algorithm in terms of the number of flows in the system and the user delay. The performance evaluation has demonstrated that HAD outperforms the QMW scheduling and is able to stabilize the networks with dynamic flows. Simulation results match the analytical conclusion very well and show that HAD can better support real-time data regarding the delay performance.

We plan to further investigate the following topics as our future work. First is a thorough investigation on the threshold of the traffic intensity when the performance of HAD converges to the M/D/1 model. Second is the design of a distributed scheduling algorithm for multihop wireless networks. Since a central scheduler may be too costly in some multihop wireless networks in practice, it is worth to study the distributed scheduling algorithm for multihop networks, which is throughput-optimal and easy to implement. Third, because we noticed that the system cannot always be stabilized by HAD if the number of users is fixed, we plan to generalize the proposed scheduling algorithm for the systems with both persistent and dynamic flows.

References

[1] Y. Chen, X. Wang, and L. Cai, “HOL delay based scheduling in wireless networks with flow-level dynamics,” in *Proc. IEEE Global Commun. Conf.*, 2014, pp. 4898–4903.

[2] J. Liu, J. Gao, X. Jiang, H. Nishiyama, and N. Kato, “Capacity and delay of probing-based two-hop relay in MANETs,” *IEEE Trans. Wireless Commun.*, vol. 11, no. 11, pp. 4172–4183, Nov. 2012.

[3] J. Liu, X. Jiang, H. Nishiyama, and N. Kato, “Generalized two-hop relay for flexible delay control in MANETs,” *IEEE/ACM Trans. Netw.*, vol. 20, no. 6, pp. 1950–1963, Dec. 2012.

[4] M. J. Neely and E. Modiano, “Capacity and delay tradeoffs for ad hoc mobile networks,” *IEEE Trans. Inf. Theory*, vol. 51, no. 6, pp. 1917–1937, Jun. 2005.

[5] M. Dong, K. Ota, A. Liu, and M. Guo, “Joint optimization of lifetime and transport delay under reliability constraint wireless sensor networks,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 27, no. 1, pp. 225–236, Jan. 1, 2016.

[6] M. J. Neely, “Stochastic network optimization with application to communication and queueing systems,” *Synthesis Lectures Commun. Netw.*, vol. 3, no. 1, pp. 1–211, 2010.

[7] X. Wang and L. Cai, “Stability region of opportunistic scheduling in wireless networks,” *IEEE Trans. Veh. Technol.*, vol. 63, no. 8, pp. 4017–4027, Oct. 2014.

[8] J. Liu, X. Jiang, H. Nishiyama, and N. Kato, “Throughput capacity of MANETs with power control and packet redundancy,” *IEEE Trans. Wireless Commun.*, vol. 12, no. 6, pp. 3035–3047, Jun. 2013.

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

[10] L. Tassiulas, “Scheduling and performance limits of networks with constantly changing topology,” *IEEE Trans. Inf. Theory*, vol. 43, no. 3, pp. 1067–1073, May 1997.

[11] L. Tassiulas and A. Ephremides, “Dynamic server allocation to parallel queues with randomly varying connectivity,” *IEEE Trans. Inf. Theory*, vol. 39, no. 2, pp. 466–478, Mar. 1993.

[12] M. Andrews, K. Kumaran, K. Ramanan, A. Stolyar, R. Vijayakumar, and P. Whiting, “Scheduling in a queuing system with asynchronously varying service rates,” *Probab. Eng. Inf. Sci.*, vol. 18, no. 2, pp. 191–217, 2004.

[13] B. Ji, C. Joo, and N. B. Shroff, “Throughput-optimal scheduling in multihop wireless networks without per-flow information,” *IEEE/ACM Trans. Netw.*, vol. 21, no. 2, pp. 634–647, Apr. 2013.

[14] K. Kar, X. Luo, and S. Sarkar, “Throughput-optimal scheduling in multichannel access point networks under infrequent channel measurements,” *IEEE Trans. Wireless Commun.*, vol. 7, no. 7, pp. 2619–2629, Jul. 2008.

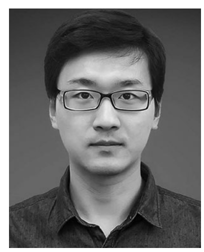
[15] K. Kar, S. Sarkar, A. Ghavami, and X. Luo, “Delay guarantees for throughput-optimal wireless link scheduling,” *IEEE Trans. Autom. Control*, vol. 57, no. 11, pp. 2906–2911, Nov. 2012.

[16] J. Liu, A. Eryilmaz, N. B. Shroff, and E. S. Bentley, “Heavy-ball: A new approach to tame delay and convergence in wireless network optimization,” in *Proc. 35th Annu. IEEE Int. Conf. Comput. Commun.*, 2016, pp. 1–9.

[17] A. Eryilmaz and R. Srikant, “Fair resource allocation in wireless networks using queue-length-based scheduling and congestion control,” in *Proc. 24th Annu. Joint Conf. IEEE Comput. Commun. Soc.*, 2005, vol. 3, pp. 1794–1803.

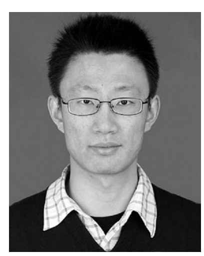
[18] M. J. Neely, E. Modiano, and C.-P. Li, “Fairness and optimal stochastic control for heterogeneous networks,” *IEEE/ACM Trans. Netw.*, vol. 16, no. 2, pp. 396–409, Apr. 2008.

[19] A. L. Stolyar, “Maximizing queueing network utility subject to stability: Greedy primal-dual algorithm,” *Queueing Syst.*, vol. 50, no. 4, pp. 401–457, 2005.
[20] M. Chiang, “Balancing transport and physical layers in wireless multihop networks: Jointly optimal congestion control and power control,” *IEEE J. Sel. Areas Commun.*, vol. 23, no. 1, pp. 104–116, Jan. 2005.
[21] M. J. Neely, E. Modiano, and C. E. Rohrs, “Dynamic power allocation and routing for time-varying wireless networks,” *IEEE J. Sel. Areas Commun.*, vol. 23, no. 1, pp. 89–103, Jan. 2005.
[22] P. van de Ven, S. Borst, and S. Shneer, “Instability of maxweight scheduling algorithms,” in *Proc. IEEE INFOCOM*, 2009, pp. 1701–1709.
[23] S. Shakkottai and A. L. Stolyar, “Scheduling for multiple flows sharing a time-varying channel: The exponential rule,” *Transl. Amer. Math. Soc. Ser. 2*, vol. 207, pp. 185–202, 2002.
[24] B. Sadiq and G. De Veciana, “Optimality and large deviations of queues under the pseudo-log rule opportunistic scheduling,” in *Proc. 46th Annu. Allerton Conf.Commun., Control, Comput.*, 2008, pp. 776–783.
[25] B. Sadiq, S. J. Baek, and G. De Veciana, “Delay-optimal opportunistic scheduling and approximations: The log rule,” *IEEE/ACM Trans. Netw.*, vol. 19, no. 2, pp. 405–418, Apr. 2011.
[26] K. Jagannathan, M. Markakis, E. Modiano, and J. N. Tsitsiklis, “Throughput optimal scheduling in the presence of heavy-tailed traffic,” in *Proc. 48th Annu. Allerton Conf. Commun., Control, Comput.*, 2010, pp. 953–960.
[27] C. Joo, “On the performance of back-pressure scheduling schemes with logarithmic weight,” *IEEE Trans. Wireless Commun.*, vol. 10, no. 11, pp. 3632–3637, Nov. 2011.
[28] S. Liu, L. Ying, and R. Srikant, “Throughput-optimal opportunistic scheduling in the presence of flow-level dynamics,” *IEEE/ACM Trans. Netw.*, vol. 19, no. 4, pp. 1057–1070, Aug. 2011.
[29] M. Andrews, “Instability of the proportional fair scheduling algorithm for HDR,” *IEEE Trans. Wireless Commun.*, vol. 3, no. 5, pp. 1422–1426, Sep. 2004.
[30] L. Zheng and L. Cai, “A distributed demand response control strategy using Lyapunov optimization,” *IEEE Trans. Smart Grid*, vol. 5, no. 4, pp. 2075–2083, Jul. 2014.
[31] X. Wang, Y. Chen, L. Cai, and J. Pan, “Scheduling in a secure wireless network,” in *Proc. IEEE Conf. Comput. Commun.*, 2014, pp. 2184–2192.
[32] J. Chen, W. Xu, S. He, Y. Sun, P. Thulasiraman, and X. Shen, “Utility-based asynchronous flow control algorithm for wireless sensor networks,” *IEEE J. Sel. Areas Commun.*, vol. 28, no. 7, pp. 1116–1126, Sep. 2010.
[33] B. Li and A. Eryilmaz, “Optimal distributed scheduling under time-varying conditions: A fast-CSMA algorithm with applications,” *IEEE Trans. Wireless Commun.*, vol. 12, no. 7, pp. 3278–3288, Jul. 2013.
[34] Q. Li and R. Negi, “Distributed throughput-optimal scheduling in ad hoc wireless networks,” in *Proc. IEEE Int. Conf. Commun.*, 2011, pp. 1–5.
[35] S. Xia and P. Wang, “Distributed throughput optimal scheduling in the presence of heavy-tailed traffic,” in *Proc. IEEE Int. Conf. Commun.*, 2015, pp. 3490–3496.
[36] K. Jagannathan, M. G. Markakis, E. Modiano, and J. N. Tsitsiklis, “Throughput optimal scheduling over time-varying channels in the presence of heavy-tailed traffic,” *IEEE Trans. Inf. Theory*, vol. 60, no. 5, pp. 2896–2909, May 2014.
[37] P. Van de Ven, S. Borst, and L. Ying, “Spatial inefficiency of maxweight scheduling,” in *Proc. Model. Optim. Mobile, Ad Hoc Wireless Netw. Int. Symp.*, 2011, pp. 62–69.
[38] B. Ji, C. Joo, and N. B. Shroff, “Exploring the inefficiency and instability of back-pressure algorithms,” in *Proc. IEEE INFOCOM*, 2013, pp. 1528–1536.
[39] B. Sadiq and G. De Veciana, “Throughput optimality of delay-driven maxweight scheduler for a wireless system with flow dynamics,” in *Proc. 47th Annu. Allerton Conf. Commun., Control, Comput.*, 2009, pp. 1097–1102.
[40] B. Li, A. Eryilmaz, and R. Srikant, “On the universality of age-based scheduling in wireless networks,” in *Proc. IEEE Conf. Comput. Commun.*, 2015, pp. 1302–1310.
[41] N. McKeown, A. Mekkittikul, V. Anantharam, and J. Walrand, “Achieving 100% throughput in an input-queued switch,” *IEEE Trans. Commun.*, vol. 47, no. 8, pp. 1260–1267, Aug. 1999.
[42] A. Eryilmaz, R. Srikant, and J. R. Perkins, “Stable scheduling policies for fading wireless channels,” *IEEE/ACM Trans. Netw.*, vol. 13, no. 2, pp. 411–424, Apr. 2005.
[43] B. Ji, G. R. Gupta, X. Lin, and N. B. Shroff, “Performance of low-complexity greedy scheduling policies in multi-channel wireless networks: Optimal throughput and near-optimal delay,” in *Proc. IEEE INFOCOM*, 2013, pp. 2589–2597.
[44] M. J. Neely, “Delay-based network utility maximization,” *IEEE/ACM Trans. Netw.*, vol. 21, no. 1, pp. 41–54, Feb. 2013.
[45] B. Ji, C. Joo, and N. B. Shroff, “Delay-based back-pressure scheduling in multihop wireless networks,” *IEEE/ACM Trans. Netw.*, vol. 21, no. 5, pp. 1539–1552, Oct. 2013.
[46] R. Laufer, T. Salonidis, H. Lundgren, and P. Le Guyadec, “Xpress: A cross-layer backpressure architecture for wireless multi-hop networks,” in *Proc. 17th Annu. Int. Conf. Mobile Comput. Netw.*, 2011, pp. 49–60.
[47] Cisco, “Data center bridging,” Tech. Rep. 2010. [Online]. Available: http://www.cisco.com/c/dam/en/us/solutions/collateral/data-center-virtualization/ieee-802-1-data-center-bridging/at_a_glance_c45-460907.pdf
[48] QualComm, “Qualcomm aims at peer-to-peer with FlashLinq,” Tech. Rep., 2011. [Online]. Available: http://www.pcworld.com/article/219048/article.html
[49] Y. Chen, X. Wang, and L. Cai, “On achieving fair and throughput-optimal scheduling for TCP flows in wireless networks,” Dept. Elect. Comput. Eng., Univ. Victoria, Victoria, BC, Canada, Tech. Rep., 2015. [Online]. Available: http://web.uvic.ca/ chenyi/doc/techreport_fairtcp_jrnl.pdf



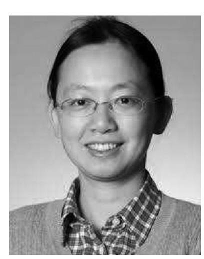
Yi Chen (S’14) received the B.Eng. and M.A.Sc. degrees in communications and information engineering from Northwestern Polytechnical University, Xi’an, China, in 2008 and 2011, respectively. He is currently working toward the Ph.D. degree with the Department of Electrical and Computer Engineering, University of Victoria, Victoria, BC, Canada.

His current research interests include scheduling design and resource allocation in wireless networks.



Xuan Wang received the B.Eng. degree in information security in 2007 and the M.S. degree in signal and information processing in 2010, both from Beijing University of Posts and Telecommunications, Beijing, China, and the Ph.D. degree in electrical engineering from the Department of Electrical and Computer Engineering, University of Victoria, Victoria, BC, under the supervision of Prof. L. Cai, in 2014.

His research interests include scheduling, resource allocation, and cross-layer design in wireless networks.



Lin Cai (S’00–M’06–SM’10) received the M.A.Sc. and Ph.D. degrees in electrical and computer engineering from the University of Waterloo, Waterloo, ON, Canada, in 2002 and 2005, respectively.

Since 2005, she has been with the Department of Electrical and Computer Engineering, University of Victoria, Victoria, BC, Canada, where she is currently a Professor. Her research interests span several areas in communications and networking, with a focus on network protocol and architecture design supporting emerging multimedia traffic over wireless, mobile, ad hoc, and sensor networks.

Dr. Cai received the National Science and Engineering Council of Canada (NSERC) Discovery Accelerator Supplement Grants in 2010 and 2015 and the Best Paper Awards from the 2008 IEEE International Conference on Communications and the 2011 IEEE Wireless Communications and Networking Conference. She served as a Technical Program Committee Symposium co-chair for IEEE Globecom’07, Globecom’10, and Globecom’13. She has served as an Associate Editor for the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, the *EURASIP Journal on Wireless Communications and Networking*, the *International Journal of Sensor Networks*, and *the Journal of Communications and Networks* and is Distinguished Lecturer of the IEEE Vehicular Technology Society.