# Scheduling in Multichannel Wireless Networks with Flow-Level Dynamics

Shihuan Liu
Electrical and Computer Engineering
Iowa State University
Ames, Iowa 50011
liush08@iastate.edu

Lei Ying
Electrical and Computer Engineering
Iowa State University
Ames, Iowa
leiying@iastate.edu

R. Srikant
Electrical and Computer Engineering
University of Illinois at Urbana-Champaign
Urbana, IL 61801
rsrikant@illinois.edu

## ABSTRACT

This paper studies scheduling in multichannel wireless networks with flow-level dynamics. We consider a downlink network with a single base station, $M$ channels (frequency bands), and multiple mobile users (flows). We also assume mobiles dynamically join the network to receive finite-size files and leave after downloading the complete files. A recent study [16] has shown that the MaxWeight algorithm fails to be throughput-optimal under this flow-level dynamics. The main contribution of this paper is the development of joint channel-assignment and workload-based scheduling algorithms for multichannel downlink networks with dynamic flow arrivals/departures. We prove that these algorithms are throughput-optimal. Our simulations further demonstrate that a hybrid channel-assignment and workload-based scheduling algorithm significantly improves the network performance (in terms of both file-transfer delay and blocking probability) compared to the existing algorithms.

## Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Wireless Communication*

## General Terms

Theory, Algorithms, Performance

## Keywords

Wireless scheduling, flow-level dynamics, multichannel downlink network

## 1. INTRODUCTION

Designing multi-user scheduling algorithms is a very challenging problem in wireless networks because of multi-scale dynamics: channel-level dynamics (channel fading), packet-level dynamics (stochastic packet arrivals) and flow-level dynamics (dynamic flow arrivals/departures). A seminal result in this area is the celebrated MaxWeight scheduling [15], which deals with both channel-level and packet-level dynamics by selecting users based on the product of channel state and queue length. Under the assumption that the set of users/nodes is fixed and all traffic flows are persistent, the MaxWeight scheduling is throughput optimal for general channel and traffic models [1, 7, 13].

While the MaxWeight scheduling achieves the maximum throughput under both channel-level and packet-level dynamics, a recent study [16] shows that the algorithm fails to retain the throughput optimality property in the presence of flow arrivals/departures, and can, in some instances, support only a small fraction of the throughput region. This observation has motivated recent interest in developing new scheduling algorithms for wireless networks with flow-level dynamics, and throughput-optimal scheduling algorithms for *single-channel* networks with flow-level dynamics have been developed in [11, 14, 16].

Motivated by current and next generation cellular systems (e.g., WiMax and LTE) implementing the Orthogonal Frequency Division Multiplexing (OFDM), in this paper, we consider multichannel wireless networks. These systems have hundreds of sub-carriers, which are grouped into tens of channels for scheduling purposes. In a multichannel network, the base-station can transmit to multiple mobile users simultaneously over different channels. Specifically, we consider a downlink wireless network where a base-station is responsible for scheduling downlink transmissions. We assume mobile users dynamically join the network for receiving finite-size files and leave the network after downloading the files. For such multichannel wireless networks, an important question that should be answered is the following: *Are the algorithms designed for single-channel networks [11, 14, 16] still throughput optimal for multichannel networks in the presence of flow-level dynamics?* The answer to this question is *no*. A counter example will be presented in Section 3. In fact in multichannel wireless networks, the base-station not only needs to decide which flow to serve on each channel, but also how to split a flow across multiple channels. We call the second problem the channel-assignment problem. Designing the channel-assignment algorithm is a key contribution of this paper, which makes both the algorithm and


*SIGMETRICS'10,* June 14–18, 2010, New York, New York, USA.

the analysis fundamentally different from those for single-channel networks.

Further, one limitation of the solutions in [11,14,16] is that they assume that each file arrives at the base station in one chunk or is served after the complete file arrives at the base-station, so these solutions result in a large file-transfer delay when a file is injected slowly into the network. Therefore another important question is the following: *Can a throughput-optimal scheduling algorithm serve a flow immediately after its first packet arrives?* The answer to this question is *yes*. Unlike prior work, we allow each file to either arrive at the base station in one chunk or to arrive packet-by-packet, and a flow is considered for scheduling immediately after the flow joins the network.

The main contributions of this paper are summarized in the following:

- We derive the necessary conditions for the stability of multi-channel downlink networks in the presence of flow-level dynamics. The conditions are simple flow-conservation constraints, which state that if the network can be stabilized under a given traffic load, there must exist a channel assignment scheme, which allocates flows (complete or partial) to different channels, such that the average workload of each channel is less than one, where the workload is defined by the number of channel uses.

- We develop a throughput optimal algorithm, which we call *joint channel-assignment and workload-based scheduling (CA-WS),* in which the channel assignment algorithm is derived based on an optimization formulation and its Lagrangian dual. We prove that the CA-WS algorithm is throughput optimal in the presence of flow-level dynamics.

- The CA-WS algorithm starts to serve a flow only after the complete file is received at the base-station, so the performance of the CA-WS is worse than the MaxWeight in light or medium traffic regimes. We then propose a hybrid CA-WS algorithm, which schedules those flows who are still injecting packets to the base-station using the MaxWeight scheduling; and schedules fully arrived flows (those flows who have been completely transferred to the base-station) using an algorithm called workload-based scheduling. It has been observed in [11] that from the performance perspective, it is important to treat flows with large size files as long-lived flows and schedule them using the MaxWeight algorithm. The paper [11] however does not discuss what size flows should be considered as "long-lived" and what size flows should be considered as "short-lived". Also in practice, the base-station may not know the size of a file before receiving the complete file. The hybrid CA-WS algorithm proposed in this paper overcomes these issues by viewing all flows as "long-lived" flows before their last packets arrive at the base-station, and seamlessly combines the MaxWeight scheduling and workload-based scheduling. We prove that the hybrid CA-WS is also throughput optimal in multichannel downlink networks with flow-level dynamics.

- Finally, we use simulations to evaluate the performance of the proposed algorithms. Our simulations confirm that the CA-WS achieves a higher throughput compared to the MaxWeight scheduling. Our simulations also show that the hybrid CA-WS algorithm leads to small file-transfer delays and blocking probabilities when the network is not critically loaded (due to the MaxWeight scheduling), and guarantees the stability of the network in the heavy-traffic regime (due to the channel assignment scheme and workload-based scheduling).

We finally comment that the stability of wireless systems under flow-level dynamics has also been studied under a time scale separation assumption [3–6, 10], where file sizes are sufficiently large so that the flows "see" the time-average throughput region which is fixed or changes slowly. This paper (similar to [11, 14, 16]) studies the performance of wireless scheduling algorithms without assuming such a time scale separation, and a flow may be completely served within one or several time slots so a flow may not "see" the time-average throughput region, which makes the problem different from those studied in [3–6, 10]. A further difference between [3–6, 10] and this paper (and [11, 14, 16]) lies in the fact that the former ones consider *utility*-based schedulers whereas the latter ones study *workload*-based (or delay-based) schedulers.

## 2. BASIC MODEL

In this section, we define the network, channel and traffic models that will be used in this paper.

**Network model:** We consider a wireless downlink network with multiple channels (frequency bands). We let $\mathcal{M}$ denote the set of channels and let $M = |\mathcal{M}|$. The network consists of a single base station and multiple flows (mobile users). The flows join the network for the purpose of receiving files from some remote source which is not modeled in our framework, and leave the network after downloading the complete file. The remote source transmits the file to the base-station, and then the base-station transmits to the mobile user. The base station can communicate with a mobile user using any of the $M$ channels. We assume time is slotted, and that at each time slot, only one flow can be served over a given channel (frequency band) but a flow can be served by multiple channels simultaneously. A two-channel, three-mobile downlink network is demonstrated in Figure 1.

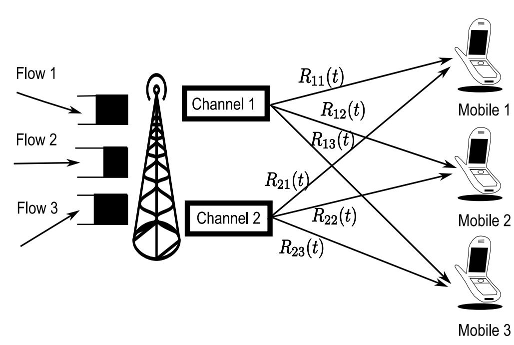


**Figure 1: A two-channel, three-mobile downlink network**

**Channel model:** We denote by $R_{if}(t)$ the state of channel $i$ seen by flow $f$ in time slot $t$, i.e., $R_{if}(t)$ denotes the number of packets that can be served by the channel at time instant $t$. We assume that $R_{if}(t)$ are a sequence of independent random variables (across time slots and across users), each distributed like some random variable $\mathfrak{R}_{if}$, where $\mathfrak{R}_{if}$ has a finite support. We denote by $R_{if}^{\max}$ the largest possible value of $\mathfrak{R}_{if}$ and $\mathbf{R}_f^{\max} = \left(R_{1f}^{\max}, \ldots, R_{Mf}^{\max}\right)$. We assume that there exists $p^{\max} > 0$ such that

$$\Pr(R_{if}(t) = R_{if}^{\max}) \geq p^{\max}$$

for all $i$, $f$ and $t$.

**Traffic model:** We denote by $\tilde{F}_f$ the size of the file associated with flow $f$ and assume $\tilde{F}_f$ are a sequence of independent random variables (across flows), each distributed like a random variable $\mathfrak{F}$. Thus, $\tilde{F}_f$ is the number of packets in flow $f$'s file. We classify flows into different classes according to the maximum-rate vector $\mathbf{R}_f^{\max}$ seen by them. So flows $f_1$ and $f_2$ belong to the same class if $R_{if_1}^{\max} = R_{if_2}^{\max}$ for all $i$. We let $\mathcal{K}$ denote the set of classes, and assume $K = |\mathcal{K}|$. We further denote by $k_f$ the class of flow $f$ and $\Lambda_{kF}(t)$ the number of class-$k$ flows that have a size of $F$ and join the network at time $t$. We assume $\Lambda_{kF}(t)$ are a sequence of independent random variables (across time slots), each distributed like $\Lambda_{kF}$, and $\lambda_{kF} = \mathbf{E}[\Lambda_{kF}]$. We further assume that the size of a file is upper bounded by $F^{\max}$ and

$$\sum_{k \in \mathcal{K}, F \leq F^{\max}} \Lambda_{kF}(t) \leq \lambda^{\max}$$

for any $t$. Finally, we denote by $F_f(t)$ the number of packets of flow $f$ queued at the base station at time $t$, and $\mathcal{F}(t)$ the set of flows in the network at time $t$.

A flow is called a *transient flow* if the last packet of the file has not arrived at the base station; and otherwise, we call the flow a *resident flow.* In this paper, we assume that the base station knows when a file is completely transferred to the base station (e.g., the base station can figure out if a flow is a resident flow by looking for a special end-of-file packet). We let $b_f$ denote the time flow $f$ joins the network, and $s_f$ the time flow $f$ becomes a resident flow. We further denote by $\mathcal{L}(t)$ the set of transient flows at time $t$, and $\mathcal{S}(t)$ the set of resident flows at time $t$.

# 3. JOINT CHANNEL ASSIGNMENT AND WORKLOAD BASED SCHEDULING

For single-channel networks in the presence of flow-level dynamics, throughput-optimal scheduling algorithms have been proposed in [11, 14, 16]. The key idea of these algorithms is to minimize the number of time-slots used to serve all traffic flows. Note that the minimum number of time slots required to fully transmit a file $f$ is $\left\lceil \tilde{F}_f / R_f^{\max} \right\rceil$, where $R_f^{\max}$ is the best channel state seen by flow $f$ and $\left\lceil \tilde{F}_f / R_f^{\max} \right\rceil$ is called the workload of flow $f$. So the idea is then to serve a flow $f$ only when $R_f(t) = R_f^{\max}$, in other words, serve a flow only if the workload of that flow can be reduced by one. Since the average workload injected into the network in one time slot should be less than one given the traffic load is within the throughput region, scheduling algorithms that reduce workload by one (with a high probability) during each time slot stabilize the network.

The reader may wonder whether we can directly use this workload-based approach to multichannel networks? *For example to be throughput-optimal, is it sufficient to serve on each channel $i$ a flow such that $R_{if}(t) = R_{if}^{\max}$?* The answer unfortunately is negative, as shown in the following example.

**Example:** Consider a network with two channels with constant service rates: $R_{1f} = B + 1$ and $R_{2f} = 2B$ for all $f$, and two types of flows in the network: the file size of a type 1 flow is $2B + 2$ and the file size of a type 2 flow is $4B$. We assume $B \geq 4$ and both types of flows arrive with a constant rate 1/2, i.e., one new arrival every two time slots.

Under this setting, consider a channel assignment that serves type 1 flows on channel 1 and type 2 flows on channel 2. Since each flow consumes two channel uses under this channel assignment, the network is stable.

However, we will now show that throughput optimality is not guaranteed by serving on each channel $i$ a flow with $R_{if}(t) = R_{if}^{\max}$. For this purpose, consider *a scheduling policy which gives priority to type 2 flows on channel 1 and priority to type 1 flows on channel 2.*

Note that each type 1 flow requires two channel uses, irrespective of the channels assigned to it, so channel 2 is fully occupied by type 1 flows with arrival rate 1/2. Each type 2 flow requires four channel uses on channel 1, so channel 1 alone cannot support type 2 flows with arrival rate 1/2. However, since channel 2 is fully occupied by type 1 flows, the number of type 2 flows will build up and the network is unstable. While this example considers deterministic arrivals for simplicity, it is not difficult to construct an example with stochastic arrivals to demonstrate the lack of throughput optimality of a policy which schedules a user with the best channel state on each channel. □

From this example, we can see the direct adoption of the workload-based algorithm for single channel networks may not be throughput-optimal for multichannel networks. This is because, in a multichannel network, a flow can be served by more than one channel, and the channels may have different best channel states. Therefore, to achieve the maximum throughput, we need to intelligently split a flow among the $M$ channels. In the previous example, the optimal solution is to assign all type-1 flows to channel 1 and all type-2 flows to channel 2. Now to develop efficient channel-assignment algorithms, our first step is to understand the throughput region of a multichannel network.

## 3.1 Necessary Conditions for Stability

To describe necessary conditions for supportability, we introduce the concept of a channel assignment vector $\mathbf{h}$. Associated with each flow is a channel assignment vector $\mathbf{h} = (h_1, h_2, ..., h_M)$, where $h_i$ denotes the number of time slots allocated to the flow on channel $i$. The parameter $h_i$ can be viewed as the workload imposed by the flow on channel $i$. For example, in a network with three channels, $\mathbf{h}_f(t) = (0, 1, 1)$ means that after time slot $t$, the base station is allowed to serve flow $f$ once (one time slot) over channel 2 and 3, but not allowed to serve the flow over channel 1. Next we define $Q_i(t) = \sum_{f \in \mathcal{S}(t)} h_{if}(t)$, where $h_{if}(t)$ is the $i^{th}$ element of vector $\mathbf{h}_f(t)$. Now given arrival rates $\{\lambda_{kF}\}$, we say that $\{\lambda_{kF}\}$ is *supportable* if there exists a scheduling algorithm, under which

$$\lim_{t \to \infty} \mathbf{E}\left[\sum_i Q_i(t)\right] < \infty$$

holds.

Further, if a flow has $F$ packets, then we only need to consider channel assignment vectors such that $\sum_{i\in\mathcal{M}} h_i \leq F$. If a traffic load is supportable, then the average rate at which workload arrives on each channel should be less than one, so we obtain the following necessary conditions for supportability.

LEMMA 1. *If arrival rates $\{\lambda_{kF}\}$ are supportable, then there exist $Z^*_{\mathbf{h}|kF} \geq 0$ such that*

$$\begin{aligned}
\sum_{k,F,\mathbf{h}} \lambda_{kF} Z^*_{\mathbf{h}|kF} h_i &\leq 1, i = 1, 2, ..., M && (1)\\
\sum_{\mathbf{h}} Z^*_{\mathbf{h}|kF} &= 1, \forall k, F && (2)\\
Z^*_{\mathbf{h}|kF} &= 0 \textit{ if } F > \sum_{i\in\mathcal{M}} h_i R^{\max}_{ik} && (3)
\end{aligned}$$

□

We comment that $Z^*_{\mathbf{h}|kF}$ can be viewed as the fraction of class $k$ flows of size $F$ that are assigned the channel assignment vector $\mathbf{h}$. Inequality (1) is the capacity constraint which says that the average workload assigned to a channel should be less than one. Inequality (2) states that every file should be associated with a channel assignment vector, and (3) states that considering a flow $f$, the channel assignment vector should guarantee sufficient service for transmitting the complete file. We do not provide a proof of Lemma 1. The proof follows along the lines of similar proofs in [15] or [7].

## 3.2 Joint Channel-Assignment and Workload-based Scheduling: An Optimization-based Design

Now based on the necessary conditions (Lemma 1), we derive an *on-line* channel-assignment algorithm using an optimization based approach. Consider the following optimization problem (feasibility problem):

$$\begin{aligned}
&\min_{\mathbf{Z}} 0\\
\sum_{k,F,\mathbf{h}} \lambda_{kF} Z_{\mathbf{h}|kF} h_i &\leq 1, i = 1, 2, ..., M\\
\sum_{\mathbf{h}} Z_{\mathbf{h}|kF} &= 1, \forall k, F\\
Z_{\mathbf{h}|kF} &= 0 \text{ if } F > \sum_{i\in\mathcal{M}} h_i R^{\max}_{ik}\\
Z_{\mathbf{h}|kF} &\geq 0, \ \forall \mathbf{h}, k, F
\end{aligned}$$

By appending some of the constraints to the objective using Lagrangian multipliers, we get:

$$\begin{aligned}
&\min_{\mathbf{Z}} \sum_i Q_i \left(\sum_{k,F,\mathbf{h}} \lambda_{kF} Z^*_{\mathbf{h}|kF} h_i - 1\right)\\
\text{subject to:} \quad &\sum_{\mathbf{h}} Z_{\mathbf{h}|kF} = 1, \forall k, F\\
&Z_{\mathbf{h}|kF} = 0 \text{ if } F > \sum_{i\in\mathcal{M}} h_i R^{\max}_{ik}\\
&Z_{\mathbf{h}|kF} \geq 0, \forall \mathbf{h}, k, F,
\end{aligned}$$

where $Q_i$ is the Lagrangian multiplier associated with constraint $\sum_{k,F,\mathbf{h}} \lambda_{kF} Z_{\mathbf{h}|kF} h_i \leq 1$.

The partially augmented problem can be decomposed into subproblems associated with each pair of $k$ and $F$ :

$$\begin{aligned}
&\min_{\mathbf{Z}} \sum_{\mathbf{h}} \sum_i Q_i Z_{\mathbf{h}|kF} h_i\\
\text{subject to:} \quad &\sum_{\mathbf{h}} Z_{\mathbf{h}|kF} = 1,\\
&Z_{\mathbf{h}|kF} = 0 \text{ if } F > \sum_{i\in\mathcal{M}} h_i R^{\max}_{ik}\\
&Z_{\mathbf{h}|kF} \geq 0, \ \forall \mathbf{h}, k, F.
\end{aligned}$$

Since the objective function is linear, the subproblem (for fixed $k$ and $F$) can be further written as:

$$\begin{aligned}
&\min_{\mathbf{h}} \sum_i Q_i h_i\\
\text{subject to:} \quad &F \leq \sum_{i\in\mathcal{M}} h_i R^{\max}_{ik}.
\end{aligned}$$

Therefore for each flow, the channel-assignment problem can be written as:

$$\begin{aligned}
&\min_{\mathbf{h}} \sum_i Q_i h_i\\
\text{subject to:} \quad &\tilde{F}_f \leq \sum_{i\in\mathcal{M}} h_i R^{\max}_{if}.
\end{aligned}$$

Recall that Lagrangian multipliers can be viewed as the price for using a given resource. Thus, if the Lagrangian multipliers are given, the channel assignment problem becomes a load balancing problem in which channel assignment is performed to minimize a weighted sum of channel prices. To compute the channel prices, we use the well-known intuition that the Lagrange multipliers are proportional to queue lengths. Note that the Lagrangian multiplier $Q_i$ is associated with the constraint $\sum_{k,F,\mathbf{h}} \lambda_{kF} Z_{\mathbf{h}|kF} h_i \leq 1$, where $\sum_{k,F,\mathbf{h}} \lambda_{kF} Z_{\mathbf{h}|kF} h_i$ is the average incoming workload during each time slot. Thus, the natural queue to consider here is the overall workload assigned to channel $i$ that has not yet been served by the network. We describe it more precisely next.

For each flow $f$, we define a channel assignment vector at time slot $t$

$$\mathbf{h}_f(t) = (h_{1f}(t), h_{2f}(t), ..., h_{Mf}(t)),$$

where $h_{if}(t)$ is the number of remaining time slots assigned to serve flow $f$ over channel $i$ at time slot $t$.

We then use $Q_i(t)$ as an estimate of the Lagrangian multiplier $Q_i$ and propose the following algorithm.

**Joint Channel-Assignment and Workload-based Scheduling (CA-WS):**

(i) **Channel-assignment:** When the last packet of flow $f$ is received at the base station (at time slot $s_f$),[1] the base station computes $\mathbf{h}_f(s_f)$ by solving the following optimization problem:

$$\begin{aligned}
\text{OPT}_f = \quad &\min \sum_{i\in\mathcal{M}} Q_i(b_f) h_{if}\\
\text{subject to:} \quad &\tilde{F}_f \leq \sum_{i\in\mathcal{M}} h_{if} R^{\max}_{if},
\end{aligned}$$

where $h_{if}$ are non-negative integers. Clearly $\mathbf{h}_f(t) = 0$ for $t < s_f$.

(ii) **Workload-based scheduling:** At time slot $t$, the base station selects a file $f$ for channel $i$ such that

$$R_{if}(t) = R^{\max}_{if} \text{ and } h_{if}(t) > 0, \quad (4)$$

[1]Here for simplicity we only consider the case where a flow can be served only after its last packet arrives at the system. Later we will consider a more general case where a flow may be served before the arrival of its last packet.

and transmits $R_{if}(t)$ packets to mobile user $f$. Then, the base station reduces $h_{if}(t)$ by one. If no flow satisfies (4), the base station randomly selects a flow, say flow $f$, and transmits $R_{if}(t)$ packets to mobile $f$ (in this case, $h_{if}(t)$ is not updated). Ties are broken arbitrarily. When the file $f$ has been completely transmitted to mobile $f$, the base station sets $h_{if}(t) = 0$ for all $i$.

□

THEOREM 2. *Assume that $s_f - b_f \leq T_{in}$ for all $f$. Given arrival rates $\{\lambda_{kF}\}$ such that $\{\frac{1}{1-\epsilon}\lambda_{kF}\}$ are supportable, the CS-WS algorithm guarantees*

$$\lim_{t\to\infty} \mathbf{E}\left[\sum_i Q_i(t)\right] < \infty.$$

PROOF. The proof of this theorem follows from the proof of Theorem 4 to be presented in the next section. Therefore, we omit the details here. □

**Remark:** The theorem assumes $s_f - b_f \leq T_{in}$, which means that the injection period of a flow (the time duration from the first packet arrives at the base station to the last packet arrives at the base station) is bounded by $T_{in}$. For example, if the flow is a constant-bit-rate flow with rate $r$, then the injection period is upper bounded by $F^{\max}/r$; and if the flow is an elastic flow whose rate is controlled by congestion control algorithm, then the injection period is also bounded when the injection rate is lower bounded as in the TCP congestion control algorithm (e.g., at least one packet over a fixed number of time slots).

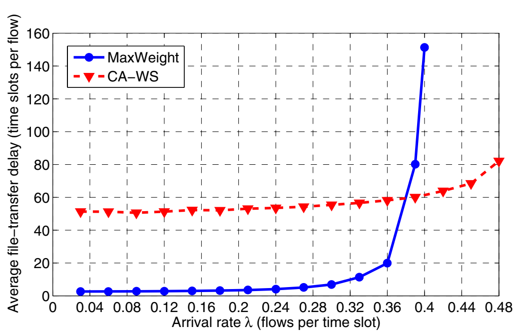


**Figure 2: Average File-transfer Delay of the CA-WS and MaxWeight algorithms**

Theorem 2 shows that the CA-WS algorithm is throughput-optimal for multichannel downlink networks, but the algorithm has two weaknesses:

- The performance of the algorithm can be poor in light to moderate traffic regimes. This is because *(i)* the base station serves a file only after the complete file is received at the base station, which results in large waiting times for large files, and *(ii)* the scheduling algorithm is independent of queue-sizes even in a light traffic regime, which again may result in large file-transfer delays for large files. Figure 2 shows a simulation result where we compare the MaxWeight algorithm and the CA-WS algorithm with uniform tie-breaking rule (CA-WSU) (the simulation setting will be described in Section 5). We can see while the CA-WSU has a smaller file-transfer delay than the MaxWeight in heavy traffic regime, in light and medium traffic regimes, the performance of the CA-WSU algorithm is much worse than the MaxWeight algorithm.

  In fact, it has been observed in [11] that from the performance perspective, we may need to serve the files with large sizes using the MaxWeight algorithm. The authors in [11] suggest that flows be classified as long-lived flows and short-lived flows, and use different scheduling algorithms for different types of flows. However, they do not provide any criterion for the classification. Further, in practice, the base station may not even know the size of a file before the file fully arrives at the base station.

- The algorithm assumes that $\mathbf{R}_f^{\max}$ is known a priori, which is unrealistic in practice.

To overcome these two weaknesses, we introduce a hybrid CA-WS algorithm in the next section.

# 4. A THROUGHPUT-OPTIMAL HYBRID CA-WS ALGORITHM

The key idea behind our hybrid algorithm is as follows: any flow whose last packet has not arrived at the base station (recall that these are called transient flows in the terminology of Section II) is treated as a persistent flow as in the traditional MaxWeight algorithm. The MaxWeight algorithm is then used to decide schedules among these flows. Flows that have fully arrived at the base station (called resident flows in Section II) are scheduled using the CA-WS algorithm. However, we have to further decide whether to schedule transient flows or resident flows over each channel. This is one of the key elements of the hybrid algorithm to be described later.

To tackle the issue of $\mathbf{R}_f^{\max}$, we adopt the learning idea introduced in [11]. We define a $\tilde{R}_{if}^{\max}(t)$ to be the best state of channel $i$ seen by flow $f$ from $b_f$ to $\min\{t, b_f + D\}$, and use $\tilde{R}_{if}^{\max}(t)$ to approximate $R_{if}^{\max}$. The parameter $D$ is called the learning period.

Before we present the hybrid CS-WS algorithm, we first define the sequence of events that take place within a slot. We assume the new flows (mobile users) arrive at the beginning of the time slot $t$ (denoted by $t_b$) and the channel state of time $t$ is also measured at $t_b$. Then we assume that any computation or recomputation of $\mathbf{h}_f(t)$ occurs at time $t_m$. Finally, the packets are served at the end of each time slot (denoted by $t_e$). The sequence of these events is demonstrated in Figure 3.

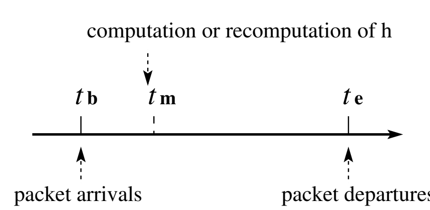


**Figure 3: The sequence of flow/packet arrivals, computation or recomputation of $\mathbf{h}(t)$ and packet departures within a time slot**

**Hybrid Channel Assignment and Workload-based Scheduling (Hybrid CA-WS):**

At time slot $t$, the flows are served as follows:

(i) When a new flow (say flow $f$) joins the network, it records $Q_i(b_f)$ for all channels $i$.

(ii) **Channel learning:** The base station measures $R_{if}(t)$ for all $i$ and $f$. Consider a flow $f$. If $t \leq b_f + D$ and $R_{if}(t) > \tilde{R}_{if}^{\max}(t-1)$ for some $i$, then flow $f$ updates the $\tilde{\mathbf{R}}^{\max}$ based on the new channel state, i.e.,

$$\tilde{R}_{if}^{\max}(t) = \max\left\{\tilde{R}_{if}^{\max}(t-1), R_{if}(t)\right\}.$$

(iii) **Channel-assignment:** Consider a resident flow $f$. If $t \leq b_f + D$ and $\tilde{R}_{if}^{\max}(t) \neq \tilde{R}_{if}^{\max}(t-1)$ for some $i$, the base station recomputes $\mathbf{h}_f(t)$ by solving the following optimization problem:

$$\begin{aligned} \mathrm{OPT}_f(t) = \quad & \min \textstyle\sum_{i\in\mathcal{M}} Q_i(b_f) h_{if}(t) \\ \text{subject to:} \quad & F_f(t) \leq \textstyle\sum_{i\in\mathcal{M}} h_{if}(t)\tilde{R}_{if}^{\max}(t), \end{aligned}$$

where $h_{if}(t)$ are non-negative integers. Note that the channel assignment for flow $f$ is recomputed every time we have a better estimate of $R_{if}^{\max}$ for any channel $i$ up to time $b_f + D$. This is necessary because the channel assignment algorithm is derived assuming $\mathbf{R}_f^{\max}$ is known.

(iv) Recall $\mathcal{L}(t)$ denotes the set of transient flows at time $t$ and $\mathcal{S}(t)$ denotes the set of resident flows at time $t$. The base station first checks:

$$\sum_{f\in\mathcal{L}(t)} F_f(t) \leq \sum_{f\in\mathcal{S}(t)} F_f(t). \tag{5}$$

- **Workload-based scheduling:** If inequality (5) holds, the base station selects a resident file $f$ for channel $i$ such that

$$\tilde{R}_{if}^{\max}(t) \leq R_{if}(t) \text{ and } h_{if}(t) \neq 0, \tag{6}$$

and transmits $R_{if}(t)$ packets to mobile user $f$. Then the base station reduces $h_{if}(t)$ by one. If no resident flow satisfies (6), the base station randomly selects a flow, say flow $f$, and transmits $R_{if}(t)$ packets to mobile $f$. Ties are broken uniformly or according to the arrival time $b_f$ (giving priority to flows with small $b_f$ may improve delay performance in practice although it has no effect on stability). When the file of flow $f$ is completely transferred to the mobile user, the base station sets $h_{if}(t) = 0$ for all $i$.

- **MaxWeight scheduling:** If inequality (5) does not hold, then the base station selects a transient file $f^*$ for channel $i$ such that

$$f^* \in \arg\max_{f\in\mathcal{L}(t)} F_f(t)R_{if}(t), \tag{7}$$

and transmits $\min\{F_f(t), R_{if^*}(t)\}$ packets to mobile user $f^*$ over channel $i$.

□

**Remark 1:** While each resident flow is associated with a channel assignment vector $\mathbf{h}_f(t)$, the packets of a flow are stored in the same queue and served in a First-In, First-Out (FIFO) fashion.

**Remark 2:** The advantages of using the MaxWeight algorithm for large-size flows are two-fold: *(i)* the file with a large size could experience smaller delay because it can be served at any $R_{if}(t)$ not just when the channel reaches the best state, and *(ii)* when only a few large-size flows are in the network, the MaxWeight algorithm can lead to a fair resource allocation. These advantages will be observed in the simulations.

In the next subsection, we will prove that the hybrid CA-WS algorithm is also throughput optimal. We would like to emphasize that because of the channel-assignment algorithm, which is not required for single-channel networks, the analysis is completely different from those in [11, 14, 16].

## 4.1 Throughput Optimality of the Hybrid CA-WS Algorithm

Without loss of generality we assume that $T_{in} = F^{\max}$, i.e., we assume that the injecting rate of any flow is at least one. All of our results apply more generally, but this assumption simplifies a lot of the notation. We first show that the number of transient flows is always bounded.

LEMMA 3. *Assume that $s_f - b_f \leq T_{in}$ for all $f$, then no more than $\lambda^{\max}F^{\max}$ transient flows are in the network during any time slot.*

PROOF. Recall that we assume that file sizes are upper bounded by $F^{\max}$, and the injecting rate is at least one. Therefore, the injection period, the time taken for a transient flow to become a resident flow, is upper bounded by $F^{\max}$. Furthermore, the number of new files joining the network at each time slot is bounded by $\lambda^{\max}$, so the number of transient files in the network is upper bounded by $\lambda^{\max}F^{\max}$. □

Since the number of transient flows is upper bounded at any time slot, to prove the stability of the network, we only need to consider the number of resident flows.

To study the performance of the hybrid CA-WS, we first define a sampled version of the network, sampled once every $T$ time slots, as follows:

$$\begin{aligned} \mathbf{M}(n) = \{&Y_f(nT), F_f(nT), \tilde{\mathbf{R}}_f^{\max}(nT), \min\{D, nT - b_f\}, \\ &\mathbf{Q}(b_f), \mathbf{h}_f(nT)\}_{f\in\mathcal{L}(t)\cup\mathcal{S}(t)}, \end{aligned}$$

where $Y_f(nT)$ is the number of packets of flow $f$ that have not been transmitted to the base station. It is easy to see that $\mathbf{M}(n)$ is a Markov chain. We also assume that the arrival process is such that the Markov chain is irreducible and aperiodic. The sampling interval $T$ in the definition of $\mathbf{M}(n)$ above will be chosen later. The reason we need this $T$ is that our proof uses the standard drift argument in Foster's criterion (see [2]), but the drift of $\mathbf{M}(n)$ may not be negative over successive time instants. The drift will be negative only after most flows in the network get reasonably accurate estimates of their channel assignment vectors, which may take several recomputations due to updates in the estimate of $\mathbf{R}^{\max}$. The parameter $T$ tries to capture the time interval that it takes for most flows to get sufficiently accurate estimates of their channel assignment vectors.

THEOREM 4. *Assume that $s_f - b_f \leq T_{in}$ for all $f$. Given arrival rates $\{\lambda_{ik}\}$ such that $\{\frac{1}{1-\epsilon}\lambda_{ik}\}$ are supportable, there* *exists a $D_\epsilon$ such that the Markov chain $\mathbf{M}(n)$ is positive-recurrent under the hybrid CA-WS algorithm with $D \geq D_\epsilon$, which implies that $\lim_{t\to\infty} \mathbf{E}\left[\sum_i Q_i(t)\right] < \infty$.*

PROOF. We consider the Lyapunov function

$$V(n) = \sum_{i\in\mathcal{M}} Q_i^2(nT),$$

and introduce the following notations:

- $\mathcal{C}(t)$ : We define $\mathcal{C}(t)$ to be the set of flows who become *resident flows* at the beginning of time slot $t$.
- $A_i(t)$ : We define $A_i(t) = \sum_{f\in\mathcal{C}(t)} h_{if}(t_m)$, which is the increase in workload for channel $i$ due to new resident flows, i.e., the flows in $\mathcal{C}(t)$.
- $\mu_i(t)$ : We define $\mu_i(t) = \sum_{f\in\mathcal{S}(t)} \left(h_{if}(t_m) - h_{if}(t_e)\right)$, which is the decrease in workload for channel $i$ when a resident flow is served over channel $i$.
- $A_i^r(t)$ : We define $A_i^r(t) = \sum_{f\in\mathcal{S}(t)\setminus\mathcal{C}(t)} \left(h_{if}(t_m) - h_{if}(t_b)\right)^+$, which is the increase in workload for channel $i$ due to the adjustment of the channel assignment vectors of existing resident flows (in other words, due to the re-computation of $\mathbf{h}_f$).
- $\mu_i^r(t)$ : We define $\mu_i^r(t) = \sum_{f\in\mathcal{S}(t)\setminus\mathcal{C}(t)} \left(h_{if}(t_b) - h_{if}(t_m)\right)^+$, which is the decrease in workload for channel $i$ due to the adjustment of channel assignments of existing resident flows.

Without causing confusion, we let $h_{if}(t) = h_{if}(t_m)$, i.e., $h_{if}(t)$ is the value after recomputation at time $t$. Now based on the notations above, the dynamics of $Q_i(t)$ can be written as

$$Q_i(t+1) = Q_i(t) + A_i(t) - \mu_i(t) + A_i^r(t) - \mu_i^r(t).$$

Note that the number of flows joining the network at each time slot is bounded by $\lambda^{\max}$, and the file size is also bounded by $F^{\max}$. Further each flow recomputes $\mathbf{h}_f$ for at most $D$ time slots. Therefore $A_i(t)$, $\mu_i(t)$, $A_i^r(t)$, and $\mu_i^r(t)$ are all bounded:

$$\begin{aligned} A_i(t) &\leq \lambda^{\max} F^{\max} \\ \mu_i(t) &\leq F^{\max} \\ A_i^r(t) &\leq \lambda^{\max} F^{\max} D \\ \mu_i^r(t) &\leq \lambda^{\max} F^{\max} D. \end{aligned}$$

Note that in general $\mu_i(t) \leq 1$ since only one channel use is allowed in one time slot. The case $\mu_i(t) > 1$ occurs when the flow is completely transmitted to the mobile user and we set $h_{if}(t) = 0$. Note that when there is no flow having $R_{if}(t) = \tilde{R}_{if}^{\max}(t)$, the base station serves a flow $f$ at rate $R_{if}(t)$ but does not reduce $h_{if}(t)$. So it is possible that even after almost all packets of a flow have been transmitted, we still have $h_{if}(t) > 1$.

Now based on the definitions and notations above, we have

$$|Q_i(nT) - Q_i(s)| \leq T\left(F^{\max} + \lambda^{\max} F^{\max}(2D+1)\right)$$

for all $s \in [nT, (n+1)T-1]$, and

$$\mathbf{E}[V(n+1) - V(n)|\mathbf{M}(n)] \leq \Phi_1$$

$$+\, 2\sum_i Q_i(nT)\mathbf{E}\left[\left.\sum_{t=nT}^{(n+1)T-1} A_i(t) + A_i^r(t) - \mu_i^r(t)\right| \mathbf{M}(n)\right] \quad (8)$$

$$-\, 2\sum_i Q_i(nT)\mathbf{E}\left[\left.\sum_{t=nT}^{(n+1)T-1} \mu_i(t)\right| \mathbf{M}(n)\right], \quad (9)$$

where $\Phi_1 = M\left(T\left(F^{\max} + \lambda^{\max} F^{\max}(2D+1)\right)\right)^2$.

In the following analysis, we will show that there exists a finite set $\mathcal{W}$ such that when $\mathbf{M}(n) \notin \mathcal{W}$, we have

$$\mathbf{E}[V(n+1) - V(n)|\mathbf{M}(n)] \leq -\frac{\epsilon}{2M}\sum_i Q_i(nT). \quad (10)$$

The theorem then follows from the Foster's criterion [2]. To prove (10), we will first analyze (8) and (9) separately, and then show that

$$\Phi_1 + (8) + (9) \leq -\frac{\epsilon}{2M}\sum_i Q_i(nT)$$

when $\mathbf{M}(n) \notin \mathcal{W}$.

### *Analysis of (8)*

Denote by $\mathcal{G}(n)$ the set of resident flows that are in the network at least in one of the time slots belonging to $[nT, (n+1)T-1]$. We further divide $\mathcal{G}(n)$ into five subsets (see Figure 4):

- $\mathcal{G}_A(n)$ : The set of resident flows that *(i)* become *resident* during $[nT, (n+1)T-D-1]$, *(ii)* are not served during $[nT, (n+1)T-1]$, and *(iii)* have learned $\mathbf{R}^{\max}$ by time $(n+1)T-1$.
- $\mathcal{G}_B(n)$ : The set of resident flows that *(i)* become *resident* during $[nT, (n+1)T-D-1]$, *(ii)* are not served during $[nT, (n+1)T-1]$, and *(iii)* have not learned $\mathbf{R}^{\max}$ by time $(n+1)T-1$.
- $\mathcal{G}_C(n)$ : The set of resident flows that become *resident* during $[nT, (n+1)T-D-1]$ and are served at least once during $[nT, (n+1)T-1]$.
- $\mathcal{G}_D(n)$ : The set of resident flows that become *resident* during $[(n+1)T-D, (n+1)T-1]$.
- $\mathcal{G}_E(n)$ : The set of resident flows that are in the system at $nT$.

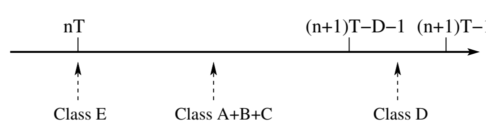


**Figure 4: Five subsets of $\mathcal{G}(n)$**

It is obvious to see that

$$\mathcal{G}(n) = \mathcal{G}_A(n) \cup \mathcal{G}_B(n) \cup \mathcal{G}_C(n) \cup \mathcal{G}_D(n) \cup \mathcal{G}_E(n).$$

Recall that $s_f$ denotes the time flow $f$ becomes a resident flow, so (8) can be rewritten as

$$\sum_{t=nT}^{(n+1)T-1} (A_i(t) + A_i^r(t) - \mu_i^r(t)) \tag{11}$$

$$= \sum_{f\in\mathcal{G}(n)} \sum_{t=\max\{s_f, nT\}}^{(n+1)T-1} (A_{if}(t) + A_{if}^r(t) - \mu_{if}^r(t)), \tag{12}$$

where $A_{if}(t)$, $A_{if}^r(t)$ and $\mu_{if}^r(t)$ are workload adjustments related to flow $f$. Next we analyze (12) case by case. To simplify our notations, we assume $D \geq \max\{F^{\max}, \frac{\lambda^{\max}F^{\max}}{\min_{k,F}\lambda_{kF}}\}$.

In the following analysis, we will show that *the subset of flows that determines the value of (8) is* $\mathcal{G}_A(n)$. Since the flows in $\mathcal{G}_A(n)$ learn the correct $\mathbf{R}_f^{\max}$ by $(n+1)T-1$ and are not served during $[nT, (n+1)T-D-1]$, we can compare the channel assignment vector under the hybrid CA-WS with that defined in the necessary conditions, which will lead to (10) by combining the analysis of (9).

**Case 1:** We first consider a flow in $\mathcal{G}_A(n)$, and have

$$\sum_{t=s_f}^{(n+1)T-1} A_{if}(t) + A_{if}^r(t) - \mu_{if}^r(t) \tag{13}$$

$$= h_{if}(s_f) + \sum_{t=s_f+1}^{(n+1)T-1} (A_{if}^r(t) - \mu_{if}^r(t)). \tag{14}$$

Since $f$ is not served before $(n+1)T$, according to the definitions of $A_{if}$, $A_{if}^r$, and $\mu_{if}^r$, we have

$$h_{if}(t+1) - h_{if}(t) = A_{if}^r(t+1) - \mu_{if}^r(t+1)$$

for any $s_f \leq t \leq (n+1)T-2$, which implies that

$$h_{if}(s_f) + \sum_{t=s_f+1}^{(n+1)T-1} (A_{if}^r(t) - \mu_{if}^r(t)) = h_{if}((n+1)T-1),$$

and

$$\mathbf{E}\left[\sum_{f\in\mathcal{G}_A(n)} \sum_{t=b_f}^{(n+1)T-1} (A_{if}(t) + A_{if}^r(t) - \mu_{if}^r(t)) \middle| \mathbf{M}(n)\right]$$
$$= \mathbf{E}\left[\textstyle\sum_{f\in\mathcal{G}_A(n)} h_{if}((n+1)T-1) \middle| \mathbf{M}(n)\right].$$

**Case 2:** Following the analysis of Case 1, for any $f \in \mathcal{G}_B(n)$, we obtain

$$\sum_{t=nT}^{(n+1)T-1} (A_{if}(t) + A_{if}^r(t) - \mu_{if}^r(t)) = h_{if}((n+1)T-1).$$

Since $h_{if}((n+1)T-1) \leq F^{\max}$ for any $f$ and any channel $i$,

$$\mathbf{E}\left[\sum_{f\in\mathcal{G}_B(n)} \sum_{t=s_f}^{(n+1)T-1} (A_{if}(t) + A_{if}^r(t) - \mu_{if}^r(t)) \middle| \mathbf{M}(n)\right]$$
$$= \mathbf{E}\left[\sum_{f\in\mathcal{G}_B(n)} h_{if}((n+1)T-1) \,|\mathbf{M}(n)\right]$$
$$\leq F^{\max}\mathbf{E}\left[\sum_{f\in\mathcal{G}_B(n)} 1\right].$$

Now we study the size of $\mathcal{G}_B(n)$. According to Lemma 3, the network has at most $\lambda^{\max}F^{\max}$ transient flows at time slot $nT-1$, which may become resident flows during $[nT, (n+1)T-1]$. Also at each time slot, at most $\lambda^{\max}$ flows join the network. For a resident flow with $s_f \leq (n+1)T-D$, the probability that the flow has not learned the $\mathbf{R}^{\max}$ by time $(n+1)T-1$ is less than $M(1-p^{\max})^D$. Therefore, we have

$$\mathbf{E}\left[\sum_{f\in\mathcal{G}_B(n)} 1\right]$$
$$\leq (\lambda^{\max}F^{\max} + (T-D)\lambda^{\max})\, M(1-p^{\max})^D,$$

and

$$\mathbf{E}\left[\sum_{f\in\mathcal{G}_B(n)} \sum_{t=s_f}^{(n+1)T-1} (A_{if}(t) + A_{if}^r(t) - \mu_{if}^r(t)) \middle| \mathbf{M}(n)\right]$$
$$\leq F^{\max}(\lambda^{\max}F^{\max} + (T-D)\lambda^{\max})\, M(1-p^{\max})^D$$
$$\leq F^{\max}\lambda^{\max}TM(1-p^{\max})^D,$$

where the last inequality holds under the assumption that $D \geq F^{\max}$.

**Case 3:** We now study the flows in $\mathcal{G}_C(n)$. Since flows $f$ are served before $(n+1)T$, according to the definition of the notations, we have

$$h_{if}(t+1) - h_{if}(t) = A_{if}^r(t+1) - \mu_{if}^r(t+1) - \mu_{if}(t)$$

for any $s_f \leq t \leq (n+1)T-2$, which implies that

$$\sum_{t=s_f}^{(n+1)T-1} A_{if}(t) + A_{if}^r(t) - \mu_{if}^r(t)$$
$$= h_{if}(s_f) + \sum_{t=s_f+1}^{(n+1)T-1} (A_{if}^r(t) - \mu_{if}^r(t))$$
$$= h_{if}((n+1)T-1) + \sum_{t=s_f+1}^{(n+1)T-1} \mu_{if}(t)$$
$$\leq F^{\max},$$

and

$$\mathbf{E}\left[\sum_{f\in\mathcal{G}_C(n)} \sum_{t=b_f}^{(n+1)T-1} (A_{if}(t) + A_{if}^r(t) - \mu_{if}^r(t)) \middle| \mathbf{M}(n)\right]$$
$$= F^{\max}\mathbf{E}\left[\textstyle\sum_{f\in\mathcal{G}_C(n)} 1 \middle| \mathbf{M}(n)\right].$$

Note that the number of flows that become *resident* during $[nT, (n+1)T-D-1]$ is no more than

$$\lambda^{\max}F^{\max} + \lambda^{\max}(T-D) \leq \lambda^{\max}T$$

since we have at most $\lambda^{\max}F^{\max}$ transient flows at time $nT-1$ and at most $\lambda^{\max}$ new flows join the network at each time slot $t$.

Now according to Lemma 5, which is presented in the appendix, that given any $\delta$, there exists $Q_\delta$ such that if $Q_i(nT) \geq Q_\delta$, then the probability that a flow with $s_f \geq nT$ is served any given time slot in $[nT, (n+1)T-1]$ is less than $\delta$. Therefore, if $Q_i(nT) \geq Q_\delta$, we have that

$$\mathbf{E}\left[\sum_{f\in\mathcal{G}_C(n)} 1 \middle| \mathbf{M}(n)\right] \leq \lambda^{\max}T^2\delta;$$

and otherwise

$$Q_i(nT)\mathbf{E}\left[\sum_{f\in\mathcal{G}_C(n)}\sum_{t=s_f}^{(n+1)T-1} A_{if}(t)+A^r_{if}(t)-\mu^r_{if}(t)\middle|\mathbf{M}(n)\right]$$
$$\leq Q_\delta\lambda^{\max}F^{\max}T.$$

We then conclude that

$$\sum_i Q_i(nT)\mathbf{E}\left[\sum_{f\in\mathcal{G}_C(n)}\sum_{t=s_f}^{(n+1)T-1} A_{if}(t)+A^r_{if}(t)-\mu^r_{if}(t)\middle|\mathbf{M}(n)\right]$$
$$\leq \sum_i \left(Q_i(nT)\lambda^{\max}F^{\max}T^2\delta+Q_\delta\lambda^{\max}F^{\max}T\right).$$

**Case 4:** We now study the flows in $\mathcal{G}_D(n)$. Following the analysis of Case 3, the size of set $\mathcal{G}_D(n)$ is upper bounded by

$$\lambda^{\max}F^{\max}+\lambda^{\max}D.$$

Therefore,

$$\begin{aligned}
&\mathbf{E}\left[\sum_{f\in\mathcal{G}_D(n)}\sum_{t=s_f}^{(n+1)T-1}\left(A_{if}(t)+A^r_{if}(t)-\mu^r_{if}(t)\right)\middle|\mathbf{M}(n)\right]\\
=\ &\mathbf{E}\left[\sum_{f\in\mathcal{G}_D(n)} h_{if}((n+1)T-1)+\sum_{t=s_f+1}^{(n+1)T-1}\mu_{if}(t)\middle|\mathbf{M}(n)\right]\\
\leq\ & F^{\max}\left(\lambda^{\max}F^{\max}+\lambda^{\max}D\right).
\end{aligned}$$

**Case 5:** We now analyze the last case: the set $\mathcal{G}_E(n)$. For a flow $f\in\mathcal{G}_E(n)$, we have the following facts:

- $A_{if}(t)=0,$
- $|A^r_{if}(t)-\mu^r_{if}(t)|\leq F^{\max}$ for any $nT\leq t<nT+D$, and
- $A^r_{if}(t)=\mu^r_{if}(t)=0$ for $t\geq nT+D$.

The last equality holds because a resident flow adjusts its $\mathbf{h}_f(t)$ for at most $D$ time slots after joining the network.

Now note that at most $\lambda^{\max}D$ flows join the network during $[nT-D,nT-1]$, which are the only flows in set $\mathcal{G}_E(n)$ that recompute $\mathbf{h}_f$ during $[nT,(n+1)T-1]$. Therefore, we obtain

$$\begin{aligned}
&\mathbf{E}\left[\sum_{f\in\mathcal{G}_E(n)}\sum_{t=nT}^{(n+1)T-1}\left(A_{if}(t)+A^r_{if}(t)-\mu^r_{if}(t)\right)\middle|\mathbf{M}(n)\right]\\
\leq\ &\lambda^{\max}D^2F^{\max}.
\end{aligned}$$

□

Summarizing the five cases above, we obtain

$$\begin{aligned}
&(8)\\
\leq&2\sum_i Q_i(nT)\mathbf{E}\left[\sum_{f\in\mathcal{G}_A(n)} h_{if}((n+1)T-1)\middle|\mathbf{M}(n)\right]\\
&+2\sum_i Q_i(nT)\Big(F^{\max}\lambda^{\max}TM\left(1-p^{\max}\right)^D+\\
&\qquad\lambda^{\max}F^{\max}T^2\delta+2\lambda^{\max}F^{\max}D+\lambda^{\max}F^{\max}D^2\Big)\\
&+MQ_\delta\lambda^{\max}F^{\max}T.
\end{aligned}\tag{15}$$

### Analysis of (9)

Next we consider (9) under the assumption that

$$\sum_i Q_i(nT)>\left(F^{\max}\right)^2\lambda^{\max}+TMF^{\max}.\tag{16}$$

It can be easily verified that under assumption (16), the base station always serves resident flows during $[nT,(n+1)T-1]$ because we have at most $\lambda^{\max}F^{\max}$ transient flows in the network at any given time.

Since $h_{if}(t)\leq F^{\max}$ for any $i$ and $f$, there are at least $Q_i(t)/F^{\max}$ flows having $h_{if}(t)>0$ at time $t$. Therefore, we obtain that

$$\Pr(\mu_i(t)=1)\geq 1-\left(1-p^{\max}\right)^{\frac{Q_i(t)}{F^{\max}}},$$

and

$$\mathbf{E}\left[\mu_i(t)|\mathbf{M}(n)\right]\geq 1-\left(1-p^{\max}\right)^{\frac{Q_i(nT)-\sqrt{\Phi_1}}{F^{\max}}}.\tag{17}$$

### Analysis of (8)+(9)

Recall that the theorem assumes that there exists $Z^*_{\mathbf{h}|kF}$ such that

$$\sum_{k,F,\mathbf{h}}\lambda_{kF}Z^*_{\mathbf{h}|kF}h_i\ \leq\ 1-\epsilon, i=1,2,...,M\tag{18}$$

$$\sum_{\mathbf{h}}Z^*_{\mathbf{h}|kF}\ =\ 1,\forall k,F\tag{19}$$

$$Z_{\mathbf{h}|kF}\ =\ 0\text{ if }F>\sum_{i\in\mathcal{M}}h_iR^{\max}_{ik},.\tag{20}$$

Next we define

$$\mathcal{H}_{kF}(n)=\{f:f\in\mathcal{G}_A(n),k_f=k,\tilde{F}_f=F\},$$

i.e., $\mathcal{H}_{kF}(n)$ is the set of class-$k$ flows that belong to set $\mathcal{G}_A(t)$ and with file length $F$. For any $f\in\mathcal{H}_{kF}(n)$, since $\mathbf{R}^{\max}_f$ has been correctly learned at time $(n+1)T-1$, we have

$$\sum_i Q_i(b_f)h_{if}((n+1)T-1)\leq\left(\sum_i Q_i(b_f)h_i\right)$$

for any $\mathbf{h}$ such that $\sum_i h_iR^{\max}_{if}\geq F$. Based on (19), we further obtain

$$\begin{aligned}
\sum_i Q_i(b_f)h_{i,f}((n+1)T-1)\leq&\ \sum_{\mathbf{h}}Z^*_{\mathbf{h}|kF}\left(\sum_i Q_i(b_f)h_i\right)\\
=&\ \sum_i Q_i(b_f)\sum_{\mathbf{h}}Z^*_{\mathbf{h}|kF}h_i.
\end{aligned}\tag{21}$$

Now based on inequality (21) and assume $T>F^{\max}$, we obtain

$$\sum_{k,F}\sum_i Q_i(nT)\mathbf{E}\left[\sum_{f\in\mathcal{H}_{kF}(n)} h_{i,f}((n+1)T-1)\middle|\mathbf{M}(n)\right]$$

$$\leq_{(a)}\Phi_1+\sum_{k,F}\mathbf{E}\left[\sum_{f\in\mathcal{H}_{kF}(n)}\sum_i Q_i(b_f)h_{i,f}((n+1)T-1)\middle|\mathbf{M}(n)\right]$$

$$\leq\Phi_1+\sum_{k,F}\mathbf{E}\left[\sum_{f\in\mathcal{H}_{kF}(n)}\sum_i Q_i(b_f)\sum_{\mathbf{h}} Z^*_{\mathbf{h}|kF}h_i\middle|\mathbf{M}(n)\right]$$

$$\leq_{(a)}2\Phi_1+\sum_{k,F}\sum_i\left(Q_i(nT)\left(\sum_{\mathbf{h}} Z^*_{\mathbf{h}|kF}h_i\right)\times \mathbf{E}\left[\sum_{f\in\mathcal{H}_{kF}(n)}1\middle|\mathbf{M}(n)\right]\right)$$

$$\leq_{(b)}2\Phi_1+\sum_{k,F}\sum_i\left(Q_i(nT)\left(\sum_{\mathbf{h}} Z^*_{\mathbf{h}|kF}h_i\right)\times \mathbf{E}\left[|\mathcal{L}(nT)|+\sum_{f:(n+1)T-D-1\geq b_f\geq nT}1\middle|\mathbf{M}(n)\right]\right)$$

$$\leq 2\Phi_1+\sum_{k,F}\sum_i\left(Q_i(nT)\left(\sum_{\mathbf{h}} Z^*_{\mathbf{h}|kF}h_i\right)\times (\lambda^{\max}F^{\max}+\lambda_{kF}(T-D))\right)$$

$$\leq 2\Phi_1+\sum_{k,F}\sum_i Q_i(nT)T\lambda_{kF}\sum_{\mathbf{h}} Z^*_{\mathbf{h}|kF}h_i, \tag{22}$$

where inequality (a) holds because $(n+1)T-1\geq b_f\geq nT-F^{\max}$ for any $f\in\mathcal{H}_{kF}(n)$ and $|Q_i(nT)-Q_i(b_f)|\leq T\left(F^{\max}+\lambda^{\max}F^{\max}(2D+1)\right)$, and inequality (b) holds because the flows in $\mathcal{H}_{kF}(n)$ must arrive during $[nT,(n+1)T-1]$ or are transient flows at time $nT$.

Now by combining inequalities (15) and (22), we get that

$$\begin{aligned}
&(8)-2\textstyle\sum_i Q_i(nT)\sum_{t=nT}^{(n+1)T-1}\sum_{\mathbf{h},k,F}\lambda_{k,F}Z^*_{\mathbf{h}|k,F}h_i\\
\leq\;& 4\Phi_1+MQ_\delta\lambda^{\max}F^{\max}T\\
&+2\textstyle\sum_i Q_i(nT)\Big(F^{\max}\lambda^{\max}TM\left(1-p^{\max}\right)^D+\\
&\lambda^{\max}F^{\max}T^2\delta+2\lambda^{\max}F^{\max}D+\lambda^{\max}F^{\max}D^2\Big)
\end{aligned}\tag{23}$$

Further, based on inequality (17), we have

$$\begin{aligned}
&2\sum_i Q_i(nT)\sum_{t=nT}^{(n+1)T-1}\sum_{\mathbf{h},k,F}\lambda_{k,F}Z^*_{\mathbf{h}|k,F}h_i+(9)\\
&=2\sum_i Q_i(nT)\times\\
&\mathbf{E}\left[\sum_{t=nT}^{(n+1)T-1}\left(\sum_{\mathbf{h},k,F}\lambda_{k,F}Z^*_{\mathbf{h}|k,F}h_i-\mu_i(t)\right)\middle|\mathbf{Q}(nT)\right]\\
&\leq 2\sum_i Q_i(nT)\left(-\epsilon T+T(1-p^{\max})^{\frac{Q_i(nT)-\sqrt{\Phi_1}}{F^{\max}}}\right).
\end{aligned}\tag{24}$$

Combining inequalities (23) and (24), we have

$$\begin{aligned}
&\mathbf{E}[V(n+1)-V(n)|\mathbf{M}(n)]\\
\leq\;&\Phi_1+(8)+(9)\\
\leq\;&5\Phi_1+MQ_\delta\lambda^{\max}F^{\max}T\\
&+2\textstyle\sum_i Q_i(nT)\Big(F^{\max}\lambda^{\max}TM\left(1-p^{\max}\right)^D+\\
&\lambda^{\max}F^{\max}T^2\delta+2\lambda^{\max}F^{\max}D+\lambda^{\max}F^{\max}D^2\Big)\\
&+2\textstyle\sum_i Q_i(nT)\left(-\epsilon T+T(1-p^{\max})^{\frac{Q_i(nT)-\sqrt{\Phi_1}}{F^{\max}}}\right).
\end{aligned}$$

Now we define a set $\mathcal{W}$ such that if $\mathbf{M}(n)\in\mathcal{W}$, then

$$\begin{aligned}
\sum_i Q_i(nT)\leq\;&\tfrac{2M}{\epsilon T}\Big(5\Phi_1+MQ_\delta\lambda^{\max}F^{\max}T\\
&+2MT\sqrt{\Phi_1}+2MTF^{\max}\tfrac{\log(\epsilon/4)}{\log(1-p^{\max})}\Big),
\end{aligned}$$

where $\delta=\frac{\epsilon}{16T\lambda^{\max}F^{\max}}$ and $Q_\delta$ is the constant defined in Lemma 5 in the appendix.

We now choose $D$ and $T$ such that

$$\begin{aligned}
D&\geq\frac{\log\frac{\epsilon}{16M\lambda^{\max}F^{\max}}}{\log(1-p^{\max})}\\
T&\geq\frac{32\lambda^{\max}F^{\max}D^2}{\epsilon}.
\end{aligned}$$

We can see that $\mathcal{W}$ is a set with a finite number of elements, and can verify that if $\mathbf{M}(n)\notin\mathcal{W}$, then

$$\mathbf{E}[V(n+1)-V(n)|\mathbf{M}(n)]<-\frac{\epsilon}{2M}\sum_i Q_i(nT).$$

Now according to the Foster's criterion, the Markov chain is positive recurrent, and further, $\lim_{t\to\infty}\mathbf{E}[\sum_i Q_i(t)]<\infty$ [12]. □

# 5. SIMULATIONS

In this section, we use simulations to evaluate the hybrid CA-WS algorithm and compare its performance with the MaxWeight scheduling scheme and the CA-WS scheduling scheme. Both the CA-WS and hybrid CA-WS algorithms in the simulations use learning to estimate the maximum transmission rate in each channel.

We consider a network with a single base station and five channels. We further assume there are three classes of flows (mobile users) in network. Class 1 users represent those close to the base station. The channel conditions of class 1 users therefore are better than those of other classes. Class 3 users represent those who are at the edge of the cell. The channel conditions of class 3 users are the worst. Class 2 users are assumed to be located in the middle of the cell. We assume that users in the same class experience the same channel fading, i.e., have the same channel distributions. We further assume that each channel has two possible states (high and low), and each of them happens with probability 0.5. The channel rate distributions of the five channels for the three classes are shown in Table 1.

The flow arrival rates of the three classes follow the same Poisson distribution with rate $\lambda$. In the simulation, we vary $\lambda$ to compare the performances of different scheduling schemes under different traffic loads. The file size of a flow follows the Pareto distribution with minimum possible value $x_m=50$,

Table 1: The Distributions of Channel Rates

| Class | Channel | High rate | Low rate |
|---|---|---|---|
| Class 1 | Channel 1 | 50 | 25 |
| | Channel 2 | 48 | 24 |
| | Channel 3 | 46 | 23 |
| | Channel 4 | 44 | 22 |
| | Channel 5 | 42 | 21 |
| Class 2 | Channel 1 | 40 | 20 |
| | Channel 2 | 38 | 19 |
| | Channel 3 | 36 | 18 |
| | Channel 4 | 34 | 17 |
| | Channel 5 | 32 | 16 |
| Class 3 | Channel 1 | 30 | 15 |
| | Channel 2 | 28 | 14 |
| | Channel 3 | 26 | 13 |
| | Channel 4 | 24 | 12 |
| | Channel 5 | 22 | 11 |

and decay factor $\alpha = 2$.[2] A transient flow keeps injecting packets into the base station until the complete file is transferred to the base station. The packet arrival rate of file $f$ is controlled by the following congestion controller [8,9]:

$$X_f(t) = \min\left\{\left\lceil \frac{50}{F_f(t)} \right\rceil, 50\right\}.$$

In the simulations, the learning period $D$ is chosen to be 20. We name the CA-WS algorithm with the uniform tie-breaking rule as CA-WSU.

## Simulation I: Number of Flows and File-Transfer Delay

We first consider the case where the base station does not limit the number of flows in the network. From the base station's perspective, it wants to minimize the total number of flows to reduce the buffer occupancy and computation complexity. From a user's perspective, the user wants to have small file-transfer delay. Therefore, we use simulations to compare the average numbers of flows in the network and the average file-transfer delays under the three scheduling algorithms.

The results are shown in Figure 5 and 6. We can see that when traffic load is light (i.e., $\lambda$ is small), the hybrid CA-WSU algorithm and the MaxWeight have similar performance, while the CA-WSU algorithm has much higher delays. The reason is that the CA-WSU scheme starts to serve a flow only after the complete file is received at the base station, which significantly increases the file-transfer delay. When $\lambda$ is large, the file-transfer delay of the MaxWeight algorithm becomes very large. This is because the MaxWeight is not throughput optimal.

Interestingly, the hybrid CA-WSU algorithm also performs much better than the CA-WSU algorithm even when $\lambda$ is large. Specifically, the average number of flows and file-transfer delay of the hybrid CA-WSU algorithm with $\lambda = 0.48$ are smaller than those under the MaxWeight or the CA-WSU algorithms with $\lambda = 0.4$.

[2] In the simulation, in order to see the performance of our algorithm under a general setting, we do not set an upper bound for file size distribution and flow arrival rate distribution.

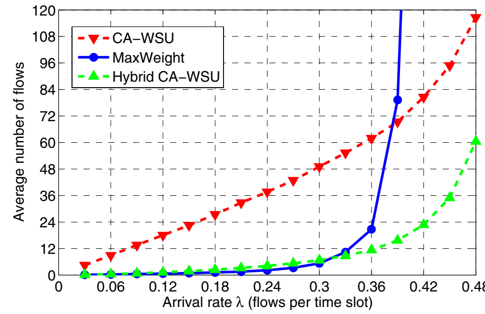


**Figure 5: The average numbers of flows under the CA-WSU, hybrid CA-WSU and MaxWeight algorithms**

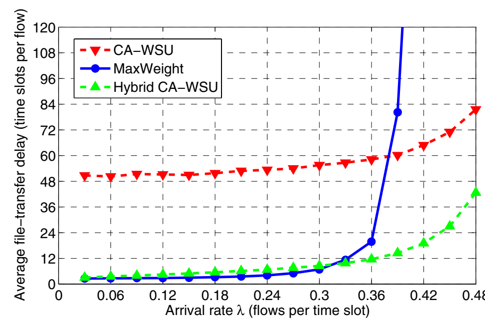


**Figure 6: The average file-transfer delays under the CA-WSU, hybrid CA-WSU and MaxWeight algorithms**

## Simulation II: Blocking probability of three algorithms

In practical systems, the base station can only support a finite number of mobiles at any given time slot. In this simulation, we assume the base station can accommodate at most 50 flows simultaneously. New flows are blocked if the number of flows in the network already reaches 50. We use the blocking probability as the performance metric to compare the three scheduling algorithms.

The result is shown in Figure 7. We can see under a small $\lambda$, all three algorithms have small blocking probabilities. However, when $\lambda = 0.5$, the blocking probability of the hybrid CA-WSU is only 6%, while the blocking probability of the MaxWeight algorithm is around 20% and the blocking probability of the CA-WSU algorithm is around 40%. Thus, our algorithm which was designed for throughput optimality assuming no limit on the number of simultaneous flows in the network also performs well in situations where the number of allowed flows is limited.

# 6. CONCLUSION

In this paper, we have developed a hybrid channel assignment and workload-based scheduling algorithm that is throughput optimal for multichannel downlink wireless networks in the presence of flow-level dynamics. The algorithm has been proved to be throughput optimal and the performance, including delay and blocking probability, has been shown to be much superior to other alternatives.

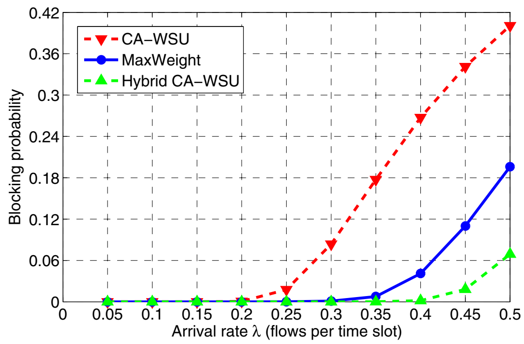


**Figure 7: The blocking probabilities of the CA-WSU, the hybrid CA-WSU and MaxWeight algorithms**

**Acknowledgments:** Research supported by NSF Grants 07-21286, 08-31756 and 09-53165, ARO MURI Subcontracts, and the DTRA grants HDTRA1-08-1-0016 and HDTRA1-09-1-0055.

## 7. REFERENCES

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

[2] Asmussen, S. *Applied Probability and Queues.* Springer-Verlag, New York, 2003.

[3] Bonald, T., Borst, S., and Proutiere, A. How mobility impacts the flow-level performance of wireless data systems. In *Proc. IEEE Infocom.* (2004), vol. 3, pp. 1872–1881.

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

[5] Borst, S. Flow-level performance and user mobility in wireless data networks. *Philosophical Transactions A 366*, 1872 (2008), 2047–2058.

[6] Borst, S., and Jonckheere, M. Flow-level stability of channel-aware scheduling algorithms. In *Proc. Intl. Symposium on Modelling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)* (April 2006), vol. 6, pp. 272–277.

[7] Eryilmaz, A., Srikant, R., and Perkins, J. R. Stable scheduling policies for fading wireless channels. *IEEE/ACM Trans. Network. 13*, 2 (2005), 411–424.

[8] Georgiadis, L., Neely, M. J., and Tassiulas, L. *Resource Allocation and Cross-Layer Control in Wireless Networks.* NOW, 2006. Foundations and Trends in Networking.

[9] Lin, X., Shroff, N., and Srikant, R. A tutorial on cross-layer optimization in wireless networks. *IEEE J. Sel. Areas Commun.* (2006).

[10] Liu, J., Proutiere, A., Yi, Y., Chiang, M., and Poor, V. Flow-level stability of data networks with non-convex and time-varying rate regions. In *Proc. Ann. ACM SIGMETRICS Conf.* (2007), p. 250.

[11] Liu, S., Ying, L., and Srikant, R. Throughput-optimal opportunistic scheduling in the presence of flow-level dynamics. In *Proc. IEEE Infocom.* (San Diego, CA, 2010).

[12] Meyn, S., and Tweedie, R. L. *Markov chains and stochastic stability.* Cambridge University Press, 2009.

[13] Neely, M., Modiano, E., and Rohrs, C. Dynamic power allocation and routing for time-varying wireless networks. *IEEE Journal on Selected Areas in Communications 23*, 1 (2005), 89–103.

[14] Sadiq, B., and de Veciana, G. Throughput optimality of delay-driven maxweight scheduler for a wireless system with flow dynamics. In *Proc. Ann. Allerton Conf. Communication, Control and Computing* (October 2009).

[15] Tassiulas, L., and Ephremides, A. Dynamic server allocation to parallel queues with randomly varying connectivity. *IEEE Trans. Inform. Theory 39* (March 1993), 466–478.

[16] van de Ven, P., Borst, S., and Shneer, S. Instability of MaxWeight scheduling algorithms. In *Proc. IEEE Infocom.* (Rio de Janeiro, Brazil, April 2009), pp. 1701 – 1709.

## Appendix

Lemma 5. *Consider the hybrid CA-WS with oldest-first or uniform tie-breaking rule and define $\mathcal{R}_i(t)$ to be the event that a resident flow $f$ with $s_f \geq nT$ is served over channel $i$ at time $t$. Given any $\delta > 0$, there exists $Q_\delta$ such that if $Q_i(nT) > Q_\delta$, then for any $nT \leq t \leq (n+1)T - 1$,*

$$\Pr(\mathcal{R}_i(t)) < \delta.$$

Proof. For any $t \in [nT, (n+1)T - 1]$, we denote by $\mathcal{O}_i(t)$ the set of resident flows that arrived before $nT - D$ ($D \geq F^{\max}$) and have $h_{if}(t) > 0$. It can be easily verified that

$$|\mathcal{O}_i(t)| \geq \frac{Q_i(nT)}{F^{\max}} - \lambda^{\max} D - T.$$

Consider the hybrid CA-WS with the oldest-first tie-breaking rule. Only if none of flows in $\mathcal{O}_i(t)$ have $R_{if}(t) = R_{if}^{\max}$, the base station will serve a flow which becomes a resident flow in $[nT, (n+1)T - 1]$ over channel $i$. Therefore, we have

$$\Pr(\mathcal{R}_i(t)) \leq (1 - p^{\max})^{\frac{Q_i(nT)}{F^{\max}} - \lambda^{\max} D - T},$$

and the lemma holds for the oldest-first tie-breaking rule.

Consider the hybrid CA-WS with the uniform tie-breaking rule. For any $t$ such that $nT \leq t \leq (n+1)T - 1$, the number of flows becoming resident flows after $nT$ is no more than $\lambda^{\max}(T + F^{\max})$. Furthermore, according to the Chernoff's bound, we have

$$\begin{aligned}&\Pr\left(\left|\{f : f \in \mathcal{O}_i(t) \text{ and } R_{if}(t) = R_{if}^{\max}\}\right| \geq (1-\delta)\Theta\right)\\ \geq\ & 1 - \exp\left(-\frac{\delta^2\Theta}{3}\right),\end{aligned}$$

where $\Theta = p^{\max}(\frac{Q_i(nT)}{F^{\max}} - \lambda^{\max} D - T)$.

Therefore, it can be easily shown that

$$\Pr(\mathcal{R}_i(t)) \leq \frac{\lambda^{\max}(T + F^{\max})}{\lambda^{\max}(T + F^{\max}) + (1-\delta)\Theta} + \exp\left(-\frac{\delta^2\Theta}{3}\right),$$

and the lemma holds for the uniform tie-breaking rule. $\square$