On the Value of Coordination and Delayed Queue Information in Multicellular Scheduling

Aditya Gopalan, Constantine Caramanis, *Member, IEEE*, and Sanjay Shakkottai, *Senior Member, IEEE*

***Abstract*—We study limited-coordination scheduling in a wireless downlink network with multiple base stations, each serving a collection of users over shared channel resources. When neighboring base stations simultaneously schedule users on the same channel resource, collisions occur due to interference, leading to loss of throughput. Full coordination to avoid this problem requires each base station to have complete "instantaneous" channel-state information for all its own users, as well as the ability to communicate on the same timescale as channel fluctuations with neighboring base stations. As such a scheme is impractical, if not impossible, to implement, we consider the setting where each base station has only limited instantaneous channel-state information for its own users, and can communicate with other base stations with a significant lag from the channel state variations to coordinate scheduling decisions.**

In this setting, we first characterize the throughput capacity of the system. A key insight is that sharing delayed queue-length information enables coordination on a slow timescale among the base stations, and this permits each base station to use limited and local channel-state along with global delayed queue-state to stabilize its users' packet queues. Based on this, we develop a distributed, queue-aware scheduling (and information exchange) algorithm that is provably throughput-optimal. Finally, we study the effect of inter-base-station coordination delay on the system packet delay performance under the throughput-optimal algorithm.

***Index Terms*—Multicellular scheduling, orthogonal frequency division multiple access (OFDMA), resource allocation.**

I. Introduction

NEXT-generation cellular systems like 3 GPP-long term evolution (LTE) [1] are based on the technique of Orthogonal Frequency Division Multiple Access (OFDMA), and promise high-speed packet-based services for a variety of applications. In a typical downlink of such a cellular system, base stations enable multiple mobile users to share available channel resources by assigning them different frequency bands or "tones". These tone-based channels experience temporal fluctuations in quality due to fading, and these fluctuations dictate the instantaneous data rates that can be sustained within a time slot. Base stations obtain channel quality measurements from the mobile users attached to them, and perform resource allocation, on a timescale of about every 1–2 milliseconds. This helps the base stations opportunistically exploit local channel fluctuations to schedule data transmissions to the users.

Data transmission in a *multicellular* environment with *many base stations* is, however, impeded by the following factors:

1) **Co-channel interference**: A system of several base stations is prone to inter-cell interference, where transmissions to neighboring mobile users assigned the same frequencies in different cells collide, resulting in a loss of throughput. The issue of interference management is especially vital in modern LTE-based *femtocell networks* [2], which are comprised of a heterogeneous deployment of base stations, each with a small footprint. Due to unplanned or ad-hoc deployments, femtocell base stations suffer from radio interference from nearby femtocells and macrocells, and this impacts overall throughput. Mitigating inter-cell interference in femtocell networks has thus been a subject of much recent research [3], [4].
2) **High backhaul latency**: Avoiding interference entirely demands that the base stations coordinate their transmissions using instantaneous channel state information acquired from other base stations. Such coordination at the timescale of channel fluctuations, however, is rendered infeasible by the relatively high latencies (of the order of 10–100 s of milliseconds) of backhaul links that connect base stations. Femtocell base stations typically use third-party IP/Ethernet backhauls to coordinate interfering transmissions, and are constrained to communicate over much slower timescales [3]. Communication between base stations is thus limited to sharing information which is significantly delayed compared to instantaneous channel state variations.
3) **Partial local channel state information**: Base stations cannot acquire even the complete instantaneous channel states for all their own users. This is because OFDMA-based systems like LTE have many sub-channels, and getting channel state information for all users on each sub-channel in every time-slot may be prohibitive in terms of available feedback bandwidth.

In such a setting, the challenge is how to effectively use a combination of network state information available at the base stations—(a) partial "local" information, i.e., instantaneous channel quality estimates, gathered from users at the timescale of channel state variations, and (b) other "global" information (such as channel statistics, accumulated queue lengths, user interference patterns etc.) gathered from other base stations

Manuscript received September 20, 2011; revised July 12, 2012; accepted December 01, 2012. Date of publication January 14, 2013; date of current version May 20, 2013. This paper was presented in part at the Information Theory and Applications Workshop, San Diego, CA, February, 2011. This work was supported in part by NSF grants, CNS-0721532, CNS-0964391, CNS-1017549, EFRI-0735905, ECCS-1056028, by the Defence Threat Reduction Agency under grant HDTRA 1-08-0029, and by the DARPA ITMANET program. Recommended by Associate Editor S. Mascolo.

A. Gopalan is with the Department of Electrical Engineering, Technion-Israel Institute of Technology, Haifa, Israel (e-mail: aditya@ee.technion.ac.il).

C. Caramanis and S. Shakkottai are with the Department of Electrical and Computer Engineering, The University of Texas at Austin, Austin TX 78713 USA (e-mail: constantine@utexas.edu; shakkott@austin.utexas.edu).

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/TAC.2013.2240098

and significantly delayed from the instantaneous channel state variations—and schedule for maximum throughput.

This paper considers a collection of base stations, each serving an exclusive set of users, in a time-slotted system. To capture the fact that certain users may interfere (e.g., users in different cells on the same frequencies located close to each other) while others may not (e.g., users on orthogonal channels, or on common frequencies but far from each other), we model an *arbitrary* collection of subsets of interfering users in the user population. Inter-cell interference is modeled by assuming that transmissions to interfering users by their respective base stations collide if scheduled simultaneously. At each time slot, each base station can access instantaneous channel states for a subset of its users, exchange delayed information with other base stations, and finally schedule users from the chosen subset.

With this information structure at the base stations, we first characterize the network throughput region, i.e., the set of all long-term joint service rates achievable for all users. A key observation we exploit is that common state information provided by global delayed queues allows coupling of decisions across base stations. We demonstrate the optimal way of using this coupled state to coordinate scheduling across multiple base stations, and develop a provably throughput-optimal scheduling algorithm. In other words, when it is possible to share global delayed information among base stations, it is enough to share delayed queue lengths to achieve throughput-optimality. To the best of our knowledge, this is the first throughput-optimality result using the information structure of local limited instantaneous channel state and global delayed information (queue lengths). We also quantify, via analysis and simulations, how the packet delay performance of our throughput-optimal scheduling algorithm varies with the amount of delay in the shared queue length information.

A. Main Contributions

The main contributions of this paper are as follows:

1) We derive the throughput region of a multi-base-station system, with given arbitrary subsets of interfering users, in which the base stations schedule using the information structure of (a) local limited channel state information, and (b) globally shared information which is independent of the instantaneous channel states. Moreover, we show that any rate within the throughput region can be obtained by timesharing, using common randomness, across a simple class of *static scheduling policies*. In a static scheduling policy, each base station always picks a fixed subset of its users, and schedules each user in that subset depending solely on its instantaneous channel state and a fixed binary vector associated with that state.
2) We present a two-tier, distributed and provably throughput-optimal scheduling algorithm that relies on the base stations sharing their users' queue lengths every T (an integer parameter) time slots. Specifically, at every T-th time slot, all the base stations communicate their queue lengths to each other. For the next T time slots, each base station uses this (delayed) queue length information, along with knowledge about channel statistics and interfering users, to locally observe an appropriate subset of its channels' states and schedule the corresponding users. The parameter T in our algorithm is the maximum "staleness" of exchanged queue length information, and is also a measure of the inter-base-station coordination time. Moreover, the scheduling algorithm is throughput-optimal for *any* fixed value of T, meaning that the coordination time T can be easily adapted to suit the latency of the backhaul between base stations without sacrificing throughput.
3) We provide analytical bounds on the system packet delay performance as a function of T, and carry out simulations to illustrate the degradation in the packet delay with increasing T.

B. Related Work

Throughput-optimal scheduling for wireless networks dates back to the pioneering work of Tassiulas *et al.* [5], [6]. Since then, there has been much work on throughput-optimal wireless scheduling, both with a central scheduler having complete network-state information [7]–[9] and distributed implementations [10]–[13]. Further references can be found in [14], [15]. Scheduling with partial or limited channel state information has been addressed in [16], where infrequent channel state information used to schedule, and [17]–[20] where scheduling is studied with partial or inaccurate observability of the aggregate channel state. In [21]–[23], the authors develop throughput-optimal algorithms using delayed channel-state information with channel state and topology uncertainty in an ad hoc network setting, where channels are independent across users. Our results differ in two ways. First, the authors in [21]–[23] do not consider the setting as in this paper where only *limited* channel-state is available at base stations—in the ad hoc network setting where neighborhoods are small, complete local-channel state is available, which is not the case in 4G base stations. In addition to the challenge of the subset selection problem, the key conceptual difference and contribution of this paper is that this subset selection occurs through the base station coordination, as we further explain below. Second, our results in this paper allow channels to be arbitrarily correlated across users. This combination of limited and correlated channel state leads to different trade-offs and scheduling algorithms.

In the multiple base station setting, two-tiered interference mitigation through load balancing and base station coordination has been studied in [24], but under the assumption that a central scheduler has instantaneous queue states of all users, and each base station has complete channel states of its users. The authors use the central scheduler to determine (based on statistics and instantaneous queue state) which of the base stations are allowed to transmit (ON base stations) and which are OFF in order to minimize interference, following which each ON base station schedules users based on its channel state information. However, the authors do not investigate queue-stability or throughput-optimality. Further, as we see from our analysis, in a distributed setting where there is no central coordinator, the optimal scheduler in fact allows collisions between transmissions from multiple base stations. The intuition for this is that due to channel randomness, it is better to be "optimistic" under some situations and attempt transmission at a base station with the hope that a contending base station's channels will be poor, and hence the contending base station will not attempt to transmit.

In [25] the authors propose a gradient power-control algorithm to mitigate inter-cell interference and dynamically reuse frequencies, while [26] considers scheduling algorithms to effectively allocate subcarriers or frequencies to users in a multicellular environment to maximize the sum throughput of the system. In [27] the authors assume coarse-grained communication among base stations along with a dynamic user model in which users randomly enter and exit the network, and present simulation results for scheduling strategies with the main metric being file transfer delay.

The authors in [4], [3] consider the problem of delayed coordination that results among LTE-Advanced femtocell base stations connected by an IP-based backhaul. They develop heuristic scheduling algorithms that account for coordination latencies, and carry out extensive numerical studies. None of the above works, though, examines the importance of using global information via delayed queue lengths and local instantaneous channel state information to stabilize queues and achieve throughput-optimality.

Finally, there is work from a physical layer perspective to maximize sum rate. However, it does not address either delayed/limited information or stability. The reader is referred to [24] for a comprehensive survey.

II. Motivating Example: How Throughput Depends on the Coordination Timescale

In this section, we present an example to illustrate how the extent of coordination in scheduling, i.e. sharing information across base stations, affects the throughput/capacity of multicellular wireless systems.

Let us consider a scenario involving two base stations and two wireless users: base station b_1 serving user u_1 and base station b_2 serving user u_2 in discrete time slots. Assume that the joint channel states of the two users are either (1, 2) or (2, 1), each with probability 0.45, or (2, 2) with probability 0.1, independently in each time slot. The channel state denotes how many packets can be transmitted to the user in the event of a successfully scheduled transmission. Corresponding to the situation in an LTE system where the two users are assigned the same frequencies and are located close to each other at the cell boundary, we assume that transmissions to these two users collide if scheduled together. At every time slot, each base station decides whether to schedule its respective user or not depending on the structure of network state information it possesses. We consider three possible structures of network state information:

1) First, assume that at every time slot, each base station knows only its own user's current channel state (i.e., the base stations have *local channel state information with no coordination*). In this case we can show that the throughput region is enclosed by the solid curved lines connecting the points (0, 1.55), (0.9, 0.9) and (1.55, 0) in Fig. 1. Essentially, this is equivalent to saying that each base station decides independently to schedule its own user with some fixed probability. The first (respectively third) point represents the case when user u_1 (respectively u_2) is always scheduled and the other user is always not scheduled. The second point represents the case when each user is scheduled if and only if its observed channel state is 2.

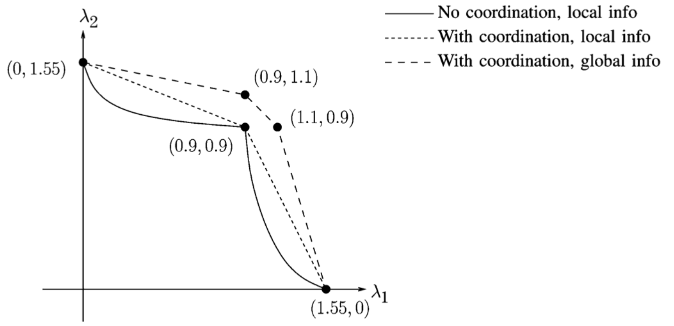


Fig. 1. Throughput region for a 2-user system under scheduling with different information structures.

2) Next, assume again that each base station knows only its own user's current channel state, but that the base stations can exchange delayed or slowly varying information—more specifically, any information independent of their users' current channel states. This models the fact that backhaul capabilities between the base stations do not permit exchange of instantaneous channel state information occurring on a fast timescale. For instance, the base stations can rely on a source of common randomness to make their scheduling decisions. This is the situation in which the base stations have *local channel state information with "slow" global coordination*. We see here that the throughput region expands to the convex hull of the earlier three points (Fig. 1); intuitively, *collaboration allows timesharing*.
3) Finally, we assume the base stations can obtain instantaneous global channel state information, i.e., acquire both the users' channel states before making scheduling decisions. This models the fact where the base stations can hypothetically exchange information as fast as the instantaneous channel states vary, and we find that the throughput region expands further to the convex hull of the points (0, 1.55), (0.9, 1.1), (1.1, 0.9) and (1.55, 0) (Fig. 1). This is because the second (respectively third) point can be achieved by scheduling *only* user u_1 (respectively u_2) when its channel state is 2—the advantage in knowing both users' instantaneous channel states comes from the fact that a base station can suitably "back off" when both channels have state 2.

The example shows that there can be a significant difference in network throughput depending on the extent of global coordination (none/slow/fast) between base stations. As mentioned earlier, Case 3—when the base stations access complete instantaneous channel state information—is practically infeasible due to the high latency of inter-base station backhaul links relative to the timescale of channel quality variations. However, backhauls in present-day LTE-based systems effectively allow delayed information on a slower timescale to be shared across base stations. This prompts us to treat Case 2 (where coordination among base stations is possible only on a timescale slower than that of the channel state) in detail in this paper, and to develop throughput-optimal scheduling algorithms for this setting.

III. System Model

This section describes the notation and definitions necessary to develop a formal model for coordinated wireless scheduling,

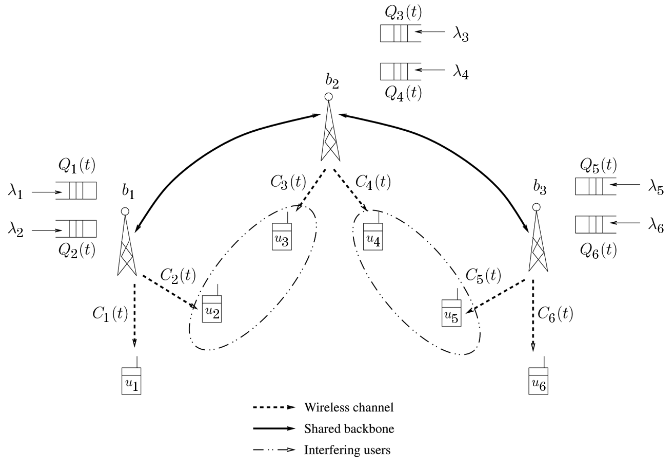


Fig. 2. Coordinated scheduling with local information.

incorporating the effect of coordination latencies between scheduling base stations.

A. Network Model

Consider N base stations $b_1, \ldots, b_N$ wishing to send packet data to M users $u_1, \ldots, u_M$ on the wireless downlink. Each user is associated with a unique base station from which it can receive data; we use $\mathcal{U}(b_i)$ to denote the set of users associated to base station b_i, and $\mathcal{B}(u_j)$ to denote the base station to which user u_j is associated. We denote the set of base stations and the set of users by $\mathcal{N}$ and $\mathcal{M}$, respectively.

B. Arrival and Channel Model

Time is slotted into discrete units. Data packets destined for user $u_j \in \mathcal{U}(b_i)$ arrive at base station b_i as a stationary nonnegative integer-valued random process $A_j(t)$, $t = 1, 2, \ldots$. For simplicity we will assume that $A_j(t)$ is *independent and identically distributed (iid)* over time slots t with $\mathbb{E}[A_j(t)] = \lambda_j$, and $A_j(t) \leq A_{\max}$. Let $A(t) \triangleq (A_1(t), \ldots, A_M(t))$. Packets get queued if they are not immediately transmitted. The channel between user u_j and its associated base station is time-varying, and we assume that its state stays constant for the duration of a time slot. We denote the channel state random process by $C_j(t)$, $t = 1, 2, \ldots$, where for any $j, C_j(t)$ takes values in a finite set $\mathcal{C}$. We explicitly assume that $\mathcal{C}$ consists of the integers $0 \leq c_1 < c_2 < \cdots < c_K = C_{\max}$. The aggregate channel state process $C(t) \triangleq (C_j(t) : j = 1, \ldots, M)$ is assumed to be independent and identically distributed (*iid*) over time slots, but the *channel states can be correlated across users*. For a subset $W \subseteq \{u_1, \ldots, u_M\}$ of users, we overload notation and denote by $C_W(t)$ the channel state of just the subset W. Let $\pi(\cdot)$ denote the probability mass function of the aggregate channel state $(C_1(t), \ldots, C_M(t))$, i.e., $\pi(r_1, \ldots, r_M) \triangleq \mathbb{P}[C_1(t) = r_1, \ldots, C_M(t) = r_M]$. Here, for each channel i, r_i takes values in the set of possible channel states $\mathcal{C} = \{c_1, c_2, \ldots, c_K\}$. Such a canonical wireless system is shown in Fig. 2.

C. Queueing Model

Each base station b_i maintains one packet queue for every user u_j associated with it, into which data packets destined to u_j get buffered if they are not immediately transmitted. We denote the fact that user $u_j \in \mathcal{U}(b_i)$ is *successfully scheduled* for data reception at time slot t by setting a binary random variable $D_j(t) = 1$. When this happens, up to $C_j(t)$ packets can be drained from its packet queue. Thus if $Q_j(t)$ denotes the queue-length process for the packet queue of user u_j, then the evolution of Q_j can be described as

$$Q_j(t+1) = \max\{Q_j(t) - D_j(t)C_j(t), 0\} + A_j(t). \tag{1}$$

Another form of (1) which we use later is

$$Q_j(t+1) = Q_j(t) + A_j(t) - E_j(t) \tag{2}$$

where $E_j(t) \triangleq \min\{D_j(t)C_j(t), Q(t)\}$. Let $Q(t)$ represent the vector of queue lengths $(Q_1(t), \ldots, Q_M(t))$ at time slot t.

D. Multiple Base Station Scheduling Model

In the multiple base station network, at every time slot t, each base station b_i schedules transmissions for a set of its users. Following the motivating example, we model the information structure that each base station uses to schedule users as having two important properties:

1) **Limited, local channel state information at base stations**: Each base station accesses instantaneous channel state information for only a *subset* of its users prior to scheduling. We let each base station b_i choose a subset $O_{b_i}(t)$ of its users at every time slot t, from a fixed arbitrary collection $\mathcal{O}(b_i)$ of subsets of $\mathcal{U}(b_i)$. Following this, the instantaneous state $C_{O_{b_i}(t)}(t)$ of the chosen users' channels is available to b_i, and it can use this knowledge to schedule users in $O_{b_i}(t)$. We use a binary random variable $B_i(t)$ to represent whether a user $i \in O_{b_i}(t)$ is scheduled at time t ($B_i(t) = 1$ denotes that i is scheduled). This framework formally captures the fact that channel state feedback capabilities between base stations and their users are potentially limited.
2) **Delayed, slower-timescale coordination between base stations**: Base stations can share information to help coordinate scheduling, but instantaneous channel state information at each time slot cannot be shared. This captures the fact that the backhaul links that allow base stations to communicate suffer from a high latency relative to the timescale of channel state variation.
To model this formally, we first fix a sufficiently large integer $R > 0$ which we will call the system *history parameter*. At time slot t, we assume that each base station can access the history of the entire network—queue lengths, channel states, arrivals—for all the previous R time slots up to and *not including* t, denoted by the (random) vector $H_R(t) \triangleq (Q(t-R), \ldots, Q(t-1), C(t-R), \ldots, C(t-1), A(t-R), \ldots, A(t-1))$. This says that the backhaul coordination links are capable of letting the base stations share their past observations with each other; however, any instantaneous (at time t) channel state information cannot be propagated within the same time slot. We remark that the restriction of available system history to the previous R time slots is made for technical convenience—our results hold when, for instance, the entire system history (past

channel states, scheduling decisions and arrivals) is available at the base stations.[1]

At each time slot t, in addition to the system history $H_R(t)$, we assume that all the base stations can utilize *common randomness*, represented by a random variable $G(t)$ that is *independent from the instantaneous channel state* $C(t)$. The independence of $G(t)$ from the instantaneous channel state is in keeping with the constraint that base stations are incapable of sharing instantaneous channel state information among themselves. $G(t)$ represents any common, auxiliary information which can be used by all the base stations. For instance, $G(t)$ can denote measured channel/arrival/queue statistics, or just a current time index that drives time-dependent scheduling, or the outcome of a common "coin toss" sequence to timeshare across base stations etc. It models common information that can be propagated across the inter-base station backhaul in order to coordinate scheduling (we discuss this in detail in Section IV in the context of time-sharing policies).

Thus, the total information available to the base stations at each time slot t, as a result of slow coordination over the backhaul, is $X_R(t) \triangleq (H_R(t), G(t))$, which we call the *system state*.

In summary, every base station first picks a subset of its users to observe their instantaneous channel states—this can depend on queue lengths, channel states, arrivals and auxiliary information in the last R slots. After having observed the instantaneous channel states for that subset, the base station schedules users in the subset depending on their instantaneous channel states and the global information it already possesses. We term the collective set of rules applied at each time slot by every base station to schedule users as the *scheduling policy or algorithm* used by the base stations.

E. Interference Model

As introduced earlier, data transmissions to users in different cells that are close to each other and use the same frequencies are prone to interference. At the same time, transmissions to users on suitably orthogonal channels (e.g. different frequencies or tones in an OFDMA-based system) can occur simultaneously without any interference. For each user $u_j \in \mathcal{M}$, let $\mathcal{I}(u_j) \subset \mathcal{M}$ denote the set of users that interfere with u_j, meaning that user u_j cannot receive any data packets in a time slot at which a user in $\mathcal{I}(u_j)$ is scheduled. We assume that $u_j \notin \mathcal{I}(u_j)$ for all j. For instance, $\mathcal{I}(u_j) = \emptyset$ means that user u_j does not experience interference from any other user. Thus, we have, for all users u_j

$$D_j(t) = B_j(t) \prod_{u_k \in \mathcal{I}(u_j)} (1 - B_k(t)) \,. \tag{3}$$

With this, the maximum number of packets that can be drained from the queue for user u_j at time t becomes

$$F_j(t) \triangleq C_j(t) D_j(t). \tag{4}$$

Such a collision interference model together with the "GO/NO-GO" type scheduling model described earlier models a rudimentary "binary" power-control scheme for users in the network.[2]

F. Objective/Performance Metric

For the setup described above, note that under any scheduling policy, the system state $X_T(t)$ is a discrete time Markov chain. Let us assume that this Markov chain is irreducible and aperiodic.[3] Following standard terminology, we say that a vector of arrival rates $\lambda = (\lambda_1, \ldots, \lambda_M)$ with $\lambda_i \geq 0, i = 1, \ldots, M$ is *supported* by a scheduling policy if the Markov chain $X_T(t)$ is positive recurrent under the policy when the packet arrival rates at the user queues are $\mathbb{E}[A_j(t)] = \lambda_j, j = 1 \ldots, M$ (this corresponds to the intuitive notion that the queues in the network are drained as fast as they fill up, i.e., they are *stable*). The goal is then to characterize the *stability region*, which we define to be the set of all vectors of arrival rates $(\lambda_j \colon j = 1, \ldots, M)$ supported by at least one scheduling policy. In addition, we wish to investigate whether there exists a *single scheduling policy* which can support *any* arrival rate vector in the stability region—a property we call *throughput optimality* of a scheduling policy.

IV. Stability Region With Slow Global Coordination

In this section, we explicitly characterize the stability region of a system of base stations that schedule users with coordination and limited local channel state information. We first introduce a class of *static* scheduling policies, called *Static Service Split (SSS)* policies, which use only instantaneous local channel state information at each base station to make scheduling decisions. SSS policies can achieve a finite set of rates in the stability region, and by using additional common randomness (e.g., a common sequence of coin toss outcomes), the base stations can suitably time-share across SSS policies. We show in Theorem 1 that the stability region is, in fact, the convex hull that results from all the possible time-sharing combinations of SSS policies. In other words, any given scheduling policy is similar, in the sense of long-term service rates, to a *Static Time-sharing (STS)* policy, or a time-shared combination of SSS policies.

A. SSS Policies

Let us consider a class of "simple" scheduling policies which, inspired by [7], we will term *Static Service Split* (SSS) policies. However, unlike the standard SSS policies used in literature for scheduling with complete channel state information [7], [28],

[1] In addition, we note that if queue length information is shared with a latency of $\eta > 1$ time slots instead of 1, the scheduling algorithm proposed in this work can be used with a slight modification to yield the same throughput and queue length performance, so knowing the state of the network at time $t-1$ is not a restrictive modeling assumption. This is because because using delayed queue length information in scheduling does not affect stability properties as long as the delay is bounded.

[2] We remark that this zero-rate interference model is assumed only for ease of exposition. With minor changes, our results and algorithm can be extended to a more complex model in which interfering users experience reduced rates. More precisely, reduced channel rates under collisions can be captured by modifying the definition of $F_j(t)$ in (3), (4) to be a specification of rates for each user corresponding to which users interfere. This would replace the hard interference constraint expressed in (4). Thus, the effect of adaptive modulation/coding schemes under interference can also be modeled, but at the expense of significant notational complications.

[3] As in [18], this can be ensured by imposing appropriate conditions on the arrival and channel process (e.g. their marginal distributions are supported with positive probability on the components in $\{0, 1, \ldots, L\}$). Weaker conditions for different stability definitions are possible, see Section II, [18] and Section III, [9] for additional discussion.

our SSS policies are essentially "two-tiered," and are specifications of both (*i*) fixed subsets that base stations must always pick and (*ii*) fixed "binary vectors" for every observed subset channel state, that indicate exactly which users must be scheduled when that channel state is observed.

Formally, an SSS policy $\mathcal{P}$ is defined by a tuple $\mathcal{P} = (W_1, \ldots, W_N, z_1, \ldots, z_N)$, where for each i, W_i is a permissible subset of users for base station b_i, and z_i is a collection of binary length-$|W_i|$ vectors, one for each possible set of observed channel states of users in W_i. Equivalently, we can think of z_i as a map that takes the channel states $C_{W_i}(t)$ observed for subset W_i into a binary vector $z_i(C_{W_i}(t)) \in \{0,1\}^{|W_i|}$. Scheduling using the SSS policy $\mathcal{P}$ is carried out as follows. At each time slot t,

1) Each base station b_i picks a *fixed* subset $O_{b_i}(t) = W_i$ of its users in order to observe their instantaneous channel states.
2) All the users for b_i not in W_i are not scheduled, i.e., $B_j(t) = 0$ for such users. A user u_j in W_i is scheduled if and only if $(z_i(C_{W_i}(t)))_j = 1$, i.e., $B_j(t) = (z_i(C_{W_i}(t)))_j$.

As an example, suppose that base station b_1 picks the subset $W_1 = \{u_1, u_2\}$ of two of its users, where each user's channel state can be either 0 or 1. Then, it can use 4 binary vectors $z_i(C_{W_1})$—one for every 2-tuple C_{W_1} of observed channel states (there are 4 in all). If, say, $z_1([1,1]) = [0,1]$, this means that if both channel states are observed to be 1, only the second user must be scheduled to transmit, and so on.

Thus, an SSS policy only specifies "local" scheduling rules per base station: each base station uses channel-state information from a predefined subset of users and schedules users in the subset accordingly. Note that scheduling decisions among different base stations are functions purely of the instantaneous channel states of their respective users, and are not coupled by any other common/shared information. In what follows, we introduce a more general class of scheduling policies called *Static Time-sharing policies*, which involve the base stations time-sharing between SSS policies using a common randomness sequence $\{G(t)\}_t$.

B. Static Time-Sharing Policies

A *Static Time-sharing (STS)* policy results when base stations use common randomness to time-share between SSS policies. It is parameterized by a finite set of SSS policies $(\mathcal{P}_i)_{i=1}^K$, together with a corresponding set of nonnegative weights $(\phi_i)_{i=1}^K$ that sum to 1. At each time slot t, independent of previous time slots, all the base stations together decide to schedule according to the SSS policy $\mathcal{P}_i$ with probability ϕ_i. This can be achieved, if, for instance, the base stations use a common random sequence $\{G(t)\}_t$ where each $G(t) \in \{1, 2, \ldots, K\}$ is the outcome of an independent K-sided coin-toss with probability distribution $(\phi_i)_{i=1}^K$, and indexes the SSS policy $\phi_{G(t)}$ for time slot t. Thus, in an STS policy, scheduling decisions among different base stations are coupled not only through their users' instantaneous channel states, but also via the common randomness that they use. In the next section, we will see that STS policies can achieve all rates in the convex hull of the rates of SSS policies (i.e., the "corner point rates"), and moreover, that no scheduling policy limited by a high-latency backhaul can stabilize rates outside this convex hull.

C. Characterization of the Stability Region

Towards an explicit characterization of the stability region with slow/high-latency backhauls, let us define the *rate vector* $\mu^{\mathcal{P}}$ associated with an SSS policy $\mathcal{P}$. For each user u_j, as in (4), let

$$F_j^{\mathcal{P}}(t) \triangleq C_j(t) D_j^{\mathcal{P}}(t)$$

where $D_j^{\mathcal{P}}(t)$ is simply $D_j(t)$ from (3) but with the superscript $\mathcal{P}$ indicating explicit dependence on the scheduling policy $\mathcal{P}$. Next, let $\mu^{\mathcal{P}} \triangleq (\mu_1^{\mathcal{P}}, \ldots, \mu_M^{\mathcal{P}})$, where

$$\begin{aligned}\mu_j^{\mathcal{P}} &\triangleq \mathbb{E}\left[F_j^{\mathcal{P}}(t)\right] \\ &= \sum_{r \equiv (r_1, \ldots, r_M)} \pi(r_1, \ldots, r_M) r_j \left(z_i(r|_{W_i})\right)_j \\ &\quad \times \prod_{\substack{u_k \in \mathcal{I}(u_j) \\ b_l = \mathcal{B}(u_k)}} \left(1 - (z_l(r|_{W_l}))_k\right). \end{aligned} \tag{5}$$

Observe that $\mu^{\mathcal{P}}$ represents the vector of long-term, ergodic service rates that the SSS policy $\mathcal{P}$ delivers to the flows to all the users in the system.[4] In a similar manner, if $\mathcal{P}$ is an STS policy, i.e., $\mathcal{P}$ is a combination of SSS policies $(\mathcal{P}_1, \ldots, \mathcal{P}_K)$ with weights $\phi_1, \ldots, \phi_K$, then we define the rate vector $\mu^{\mathcal{P}} = (\mu_1^{\mathcal{P}}, \ldots, \mu_M^{\mathcal{P}})$ associated with $\mathcal{P}$ by

$$\mu_j^{\mathcal{P}} \triangleq \sum_{i=1}^K \phi_i \mu_j^{\mathcal{P}_i}. \tag{6}$$

Essentially, the rate vector for an STS policy is defined to be the convex combination of the rate vectors of its component SSS policies.

For an SSS policy $\mathcal{P}$ with rate vector $\mu^{\mathcal{P}}$, all arrival rate vectors $\lambda = (\lambda_1, \ldots, \lambda_M)$, with $\lambda_i \geq 0$, $i = 1, \ldots, M$, that are dominated by $\mu^{\mathcal{P}}$ lie in the system stability region, since the scheduling policy $\mathcal{P}$ stabilizes them. Furthermore, any arrival rate vector that is the convex combination of SSS policy rate vectors also belongs to the stability region, since an appropriate STS policy corresponding to the convex combination stabilizes it. Let

$$\begin{aligned}\mathcal{R} &\triangleq \text{int Co}\left(\{\mu^{\mathcal{P}} : \mathcal{P} \text{ an SSS policy}\}\right) \\ &= \text{int}\{\mu^{\mathcal{P}} : \mathcal{P} \text{ an STS policy}\}\end{aligned}$$

where Int Co (Δ) refers to the interior of the convex hull of the set Δ in standard Euclidean space. Then, the preceding argument indicates that $\mathcal{R}$ is definitely an *inner* bound to the stability region (recall that the stability region consists of all those arrival rate vectors which can be supported by *some* scheduling policy). Theorem 1 below states that in fact, the stability region is no more than $\mathcal{R}$:

[4] Equation (5) defines the expected offered service rate $\mu_j^{\mathcal{P}}$ per time slot that user u_j sees under the scheduling policy $\mathcal{P}$. For this purpose, the product on the right hand side in (5) is taken over all users that interfere with user u_j (these users could be associated to different BSs). Thus, a general term in the product is for a user u_k that interferes with u_j and the (unique) BS $b_l = \mathcal{B}(u_k)$ to which u_k is associated. In this regard, the subscript l in the sum represents the BS of an interfering user.

TABLE I
CHANNEL STATE DISTRIBUTION FOR THREE-USER, TWO-BASE-STATION SYSTEM EXAMPLE

Channel \ State	s_1	s_2	s_3	s_4
$C_1(t)$	0	0	1	1
$C_2(t)$	1	0	0	1
$C_3(t)$	0	1	0	1
State probability	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{8}$

Theorem 1: The stability region of the system is $\mathcal{R}$, i.e., a vector of arrival rates $\lambda = (\lambda_1, \ldots, \lambda_M)$ with $\lambda_i \geq 0$, $i = 1, \ldots, M$ is supported by a scheduling policy if and only if $\lambda \in \mathcal{R}$.

This result says that any scheduling policy which stabilizes the system for a certain choice of arrival rates effectively behaves like an STS policy, i.e., a suitable time-shared combination of SSS scheduling policies, in the sense of the long-term service rates it delivers. This result is useful later, in Section V, towards showing that a particular scheduling algorithm we develop is throughput-optimal. The proof of this theorem is similar in spirit to the results in [7], [21], [22] used to characterize the stability region. It uses the fact that a system stable/ergodic under a policy must have consistent long-term fractions, which are in turn used to construct STS policies yielding the same service rates. Refer to Appendix A for the proof.

D. Example: Stability Region for a Three-User, Two-Base-Station System

To illustrate the concepts and result of the previous section, let us derive the stability region for a simple case of two base stations b_1 and b_2 serving a total of three users $\{u_1, u_2, u_3\}$. u_1 is associated to b_1 whereas u_2 and u_3 are associated to b_2. Channel states for all the three users are either 0 or 1 (*ON/OFF* channels). Consider the case when all the users are in geographic proximity to each other and have been assigned the same frequency bands by their respective base stations, so that simultaneously scheduled transmissions to any two users collide. In other words, $\mathcal{I}(u_1) = \{u_2, u_3\}$, $\mathcal{I}(u_2) = \{u_1, u_3\}$ and $\mathcal{I}(u_3) = \{u_1, u_2\}$.

Let us assume that base station b_2 can pick at most one of its two users at any time slot to sample, i.e., $\mathcal{O}(b_2) = \{\{b_2\}, \{b_3\}\}$, while $\mathcal{O}(b_1) = \{\{u_1\}\}$ trivially. For simplicity, we let the joint channel state distribution of the aggregate channel $(C_1(t), C_2(t), C_3(t))$ take one of four states $s_1, \ldots, s_4$ as shown in Table I.

Let us compute the throughput region of the system with the given channel state statistics, according to Theorem 1. First, consider the case when base station b_2 always picks u_2 to sample in the first scheduling step. The set of achievable long-term throughput rates with just users u_1 and u_2 is the shaded region shown in Fig. 4(a). In this figure, the extreme points (1/4, 0) and (0, 5/8) are the service rates when users u_1 and u_2 are always scheduled for service respectively, with the other user in each case never scheduled. The extreme point (1/8, 1/2) represents the service rates when users u_1 and u_2 are scheduled if and only if their respective channel state is 1 (*ON*). In this case there is a loss of throughput due to collision when both channel states are 1.

Remark: The dotted line in Fig. 4(a) represents the additional throughput obtained when both base stations can see the channel states of *both* u_1 and u_2 before scheduling. This helps reduce collisions when both the channels have state 1 and hence increases throughput.

Similarly, we can compute the set of service rates when base station b_2 always picks u_3. The set of achievable long-term throughput rates with just users u_1 and u_3 is the shaded region shown in Fig. 4(b). Here again, we see three extreme points on the "northeast" boundary of the region, having similar interpretations as in the previous figure. Also, the dotted line represents throughput gained if both base stations know the joint channel states of u_1 and u_3 before scheduling.

Theorem 1 now tells us that the stability region of the system can be found by taking the convex hull of the two "sub"-rate regions we found earlier. This is depicted graphically as the shaded region in Fig. 4(c).

V. THROUGHPUT-OPTIMAL SCHEDULING WITH SLOW GLOBAL COORDINATION

The result of Theorem 1 characterizes the network stability region as the long-term service rates of static timesharing scheduling policies. However, observe that

- Any given static timesharing policy is not throughput-optimal, since the set of arrival rates stabilizable by the policy is merely the "cube" of points that are dominated in every coordinate by its long-term service rates, and not the whole stability region.
- For any arrival rate in the stability region, stabilizing the queueing system with an appropriate static timesharing scheduling policy requires knowledge of the arrival rate. In other words, the static timesharing policies that stabilize the system for a given arrival rate explicitly depend on the arrival rate. This motivates the need for a throughput-optimal scheduling algorithm that, for *any* arrival rate in the stability region, keeps all queues stable.

In this section, we focus on developing a *throughput-optimal* information exchange and scheduling algorithm in which base stations share delayed queue length information among themselves, and use this information to select local SSS policies.

A. A Throughput-Optimal Scheduling Algorithm With Slow Global Coordination

The scheduling algorithm $\mathcal{P}_T$ that we describe here uses an integer parameter $T > 0$, which represents the time interval (in slots) between successive communication exchanges between the base stations. Specifically, the algorithm operates in successive epochs of T time slots, and requires that all the queue lengths in the system be shared globally across the base stations at the beginning of each epoch. Thus, the timescale at which the base stations exchange queue length information for coordinating their scheduling actions is once every T time slots. The scheduling policy $\mathcal{P}_T$ is as described in Algorithm 1, and essentially operates as follows:

Algorithm 1 Scheduling algorithm $\mathcal{P}_T$

```
Parameter: Integer $T > 0$.
for $t = 1, 2, \ldots$
    1) if $t \equiv 1 \pmod{T}$
        • Let the global queue length vector at time slot $t-1$ be $Q(t-1) = q \equiv (q_1, \ldots, q_M)$. Each base station $b_i$ then solves the following (common) optimization problem with the decision variables being (i) one observable subset $S_{b_i}$ per base station $b_i$, and (ii) a specification of users—represented by binary vectors $z_i(r_{S_{b_i}})$—to be scheduled for each observable subset $S_{b_i}$ and every joint channel state $r_{S_{b_i}}$ of the subset:
```

$$\max_{\substack{S_{b_1},\ldots,S_{b_N}\\ z_1,\ldots,z_N}} \sum_{j=1}^{M} q_j \sum_{r_1,\ldots,r_M} \pi(r_1,\ldots,r_M) r_j \left(z_i(r_1,\ldots,r_M)\right)_j \times \prod_{u_k \in \mathcal{I}(u_j): u_k \in \mathcal{U}(b_h)} \left(1 - \left(z_h(r_1,\ldots,r_m)\right)_k\right)$$

$$\text{s.t.} \quad S_{b_i} \in \mathcal{O}(b_i),\ z_i : \mathcal{C}^M \to \{0,1\}^M,\ i = 1,\ldots,N, \quad (z_i(\cdot))_j \equiv 0\ \forall j \notin S_{b_i}, \quad z_i(r_1,\ldots,r_m) = z_i\left(r_1',\ldots,r_m'\right) \text{ if } r_l = r_l'\ \forall l \in S_{b_i}. \tag{7}$$

```
        • Let $(S_{b_1}^*, \ldots, S_{b_N}^*, z_1^*, \ldots, z_N^*)$ be a choice of arguments that solves (7).
    end if
    2) Each base station $b_i$ chooses the subset $S_{b_i}^*$ of its users, and schedules every user $u_j$ in that subset for which $(z_i^*(C(t)))_j = 1$.
end for
```

1) Each base station accesses the global vector of queue lengths every T slots.
2) The global vector of queue lengths time slot kT is used to choose a "temporally local" SSS policy for the next T time slots. The subsets and binary decision vectors for this local SSS policy are chosen in such a way as to *maximize the sum of "local" service rates delivered to each queue weighted by its corresponding delayed queue length*.

Remark: The constraints in the optimization problem (7) mainly deal with the structure of the binary decision vectors z_i for BS b_i while observing only a subset S_{b_i} of its channel states. The two nontrivial constraints that a vector z_i must satisfy, expressed by the final two constraints, are: (a) no user outside the subset S_{b_i} can be scheduled by b_i, i.e., any such user's binary decision is 0, and (b) for a user within S_{b_i}, its scheduling decision can depend only on channel states observed within the subset S_{b_i}, i.e., the user's binary decision cannot change if no channel state in S_{b_i} changes.

Note that when the history parameter R is at least T, the scheduling algorithm $\mathcal{P}_T$ formally makes the state process $\{X_T(t)\}_{t=0,1,\ldots}$ a time-inhomogeneous Markov chain, since within each scheduling epoch, the queue lengths used as weights in (7) depend on how far into the epoch the algorithm is operating. To keep the presentation clear, we avoid such technicalities and deal, instead, with the system state sampled at the start of every epoch $\{X_T(kT)\}_{k=1,2,\ldots}$, which is a homogeneous Markov chain under the algorithm $\mathcal{P}_T$.

Our next result—Theorem 2—establishes two key properties of the algorithm $\mathcal{P}_T$:

1) **Throughput-optimality**: The scheduling policy $\mathcal{P}_T$ is throughput-optimal *for any fixed value of* T, i.e., it can support any arrival rate λ in the stability region $\mathcal{R}$.
2) **Packet-delays under $\mathcal{P}_T$ are *linear* in T**: The average queue lengths in the system, under the scheduling algorithm $\mathcal{P}_T$, grow at most linearly with the information lag T. As a result, by Little's Law, the average packet delays are linear in the average queue lengths for fixed arrival rates, and hence also grow at most linearly in T. Indeed, the parameter T models the lag or delay incurred by the base stations in exchanging queue length information, and with an increasing information lag T, queueing delays seen by incoming arrivals grow. Theorem 2 result helps quantify this intuition precisely by giving an $O(T)$ average queue length bound.

Theorem 2: For a fixed $T > 0$, if the arrival rate $\lambda \in \mathcal{R}$, then under the scheduling algorithm $\mathcal{P}_T$,

1) $\{X_T(kT)\}_k$ is a positive-recurrent Markov chain,
2) There exists a constant $\alpha > 0$, not depending on T, such that under the stationary distribution of $\{X_T(kT)\}_k$

$$\mathbb{E}^{\pi}\left[\sum_{j=1}^{M} Q_j(lT)\right] \le \alpha T \quad \forall l. \tag{8}$$

($\mathbb{E}^{\pi}$ denotes the expectation under the stationary distribution of $\{X_T(kT)\}_k$.)

We have seen earlier—in Section IV and Theorem 1—that scheduling using SSS policies can only achieve finitely many extreme points of the stability region; additional common randomness is required to stabilize the entire throughput region via time-sharing across SSS policies. What Theorem 2 shows is that this crucial role of global, common randomness can be played by delayed/slow-timescale queue-length information shared among base stations during the operation of the scheduling algorithm $\mathcal{P}_T$. The delayed queue-length updates help to correctly couple the base stations' local SSS scheduling decisions, so as to achieve the right time-sharing combination and stabilize any valid arrival rate. This is reminiscent of the manner in which queue length information is used in a distributed fashion by Ying and Shakkottai [21] for transmission contention over interfering collision channels to achieve throughput-optimality. However, their work does not investigate a two-time-scale information structure (i.e., slower, delayed backhaul information and faster, instantaneous channel states restricted to base stations), whereas our model addresses both (a) delayed inter-base station coordination and (b) the issue of choosing subset-based partial channel state information at each base station.

The proof of Theorem 2 relies on the key observation that delayed queue length information in the optimization (7) helps to pick out the "right" set of contending base stations so that, even when collisions occur within each epoch of T time slots, the base stations' decisions are time-shared to ensure the T-slot negative drift of a quadratic Lyapunov function of the queue lengths. At the same time, since each base station can access only partial, subset-based channel state information from its users, we employ techniques from scheduling with subset-based CSI [18] to show that the "correct" observable subsets are picked and time-shared by every base station so as to locally achieve the right service rates. The results in the above work [18] solve the problem of choosing partial, subset-based channel state information at a lone base station for throughput-optimality; however, in this paper we face and overcome the novel challenge of simultaneously combining (a) partial channel state information at every base station with (b) global delayed queue-length updates across different base stations, for achieving throughput-optimality in the presence of inter-cell interference.

We show the negative drift of the quadratic Lyapunov function under $\mathcal{P}_T$ by proving that

1) The T-slot drift under $\mathcal{P}_T$ is of the same form as that under a static time-sharing scheduling policy, and
2) Because $\mathcal{P}_T$ solves the optimization (7) at every T-th time slot, its local T-slot drift is the most negative (and bounded away from zero) across all static time-sharing policies.

Finally, the queue-length bound in (8) is obtained from the negative T-slot Lyapunov drift by using a technique due to Neely [14]. The reader is referred to Appendix B for the complete proof of the theorem.

VI. Simulation Results: How Packet Delays Under $\mathcal{P}_T$ Vary With T

In this section we present simulation results that illustrate the impact of the coordination delay T and system load (i.e., how close the arrival rate vector is to the boundary of the throughput region) on the average delay experienced by arriving packets, under the scheduling algorithm $\mathcal{P}_T$. We consider two example setups involving multiple base stations and users, interfering channels and subset-constrained scheduling at base stations and explore the packet delay performance of the algorithm.

A. 3-User, 2-Base Station Example

The network model we first consider for numerical simulation is the one presented and discussed in Section IV-D, with 3 users and 2 base stations. With regard to the throughput region of the system, shown in Fig. 4(c), consider the rate vector $\hat{\lambda} = (1/16, 1/4, 3/16)$ which is the midpoint of the edge joining the corner points (1/8, 1/8, 0) and (0, 0, 3/8), and on the boundary of the throughput region. For a scaled version $\lambda_\epsilon = (1-\epsilon)\hat{\lambda}$, we say that λ_ϵ represents a "load" of $1-\epsilon$ to the system, analogous to the terminology used in describing load in an *M/M/1* queue. Arrivals are generated in an *iid* Bernoulli fashion and scheduling is performed using the T-slot throughput-optimal policies developed in Section V. We examine the average delay or waiting time experience by packets that enter the network, in the following two cases:

1) Effect of Coordination Delay: For five different loads to the system (0.55 to 0.95 in steps of 0.1), the impact of varying the coordination interval T from 1 to 100 on the packet delay is as shown in Fig. 5(a). We observe that the growth in average packet delay is linear with T which is in accordance with the result of part 2 of Theorem 2, since by Little's law the average delay in the network is proportional to the average queue lengths for a fixed net arrival rate.
2) Effect of Load: For five different values of coordination interval ($T = 1$, $T = 10$, $T = 50$, $T = 100$ and $T = 150$), we plot the average packet delay in the system versus load increasing from 0.5 towards 1. The increase in average packet delay is observed to be particularly severe as the load approaches 100%.

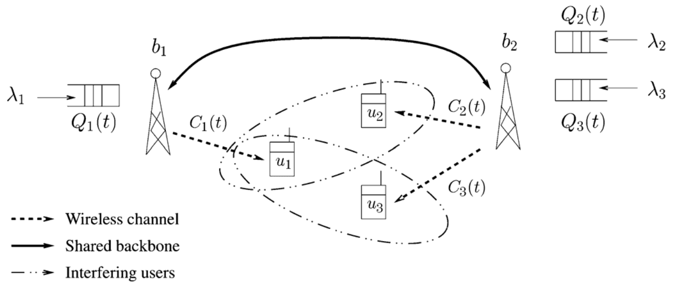


Fig. 3. Stability region example: Three-user, Two-base-station System with slow coordination between the base stations.

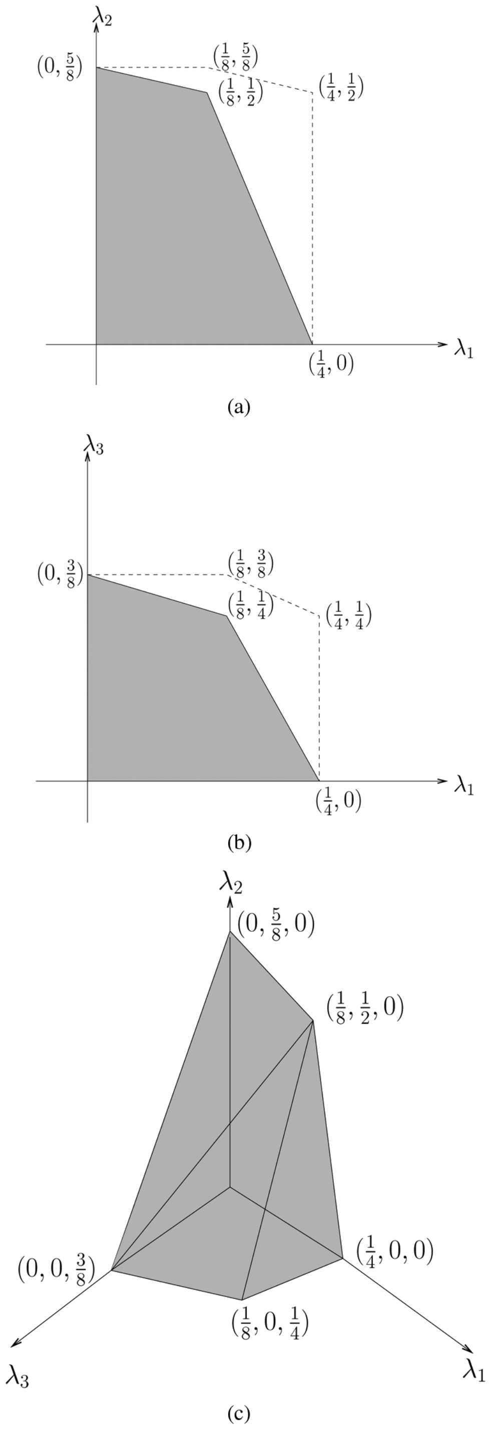
B. 10-User, 3-Base Station Example

For the second simulation study, we consider a wireless network comprised of 3 base stations b_1, $b2$ and b_3 that serve a total of 10 users $u_1, u_2, \ldots, u_{10}$. Users u_1, u_2 and u_3 are associated with b_1, u_4, u_5 and u_6 with b_2 and the remaining 4 users are served by base station b_3. Fig. 6 depicts the user locations within their respective cells and their interference pattern. Users $\{u_1, u_4\}$ form an interfering or colliding set of users, as do $\{u_5, u_8\}$ and $\{u_3, u_6, u_9\}$. This happens, for instance, when each user in the interfering set is close to its cell edge and is allotted the same transmission frequency by its base station (indicated in Fig. 3 by a colored ellipse containing the respective interfering users).

The channel rates for users 1, 5 and 9 are modeled as Bernoulli (0.7) independent random variables at each time slot, and the rates for all other users are assumed to be always ON (i.e., 1 packet per time slot at all times). We assume a symmetric arrival rate of 0.05 packets per time slot to all users. Each base station can observe instantaneous NSI for 3 of its users, which means that b_3 must pick a subset of 3 of its 4 users at each time to observe channel state. The packet delay performance of the scheduling algorithm $\mathcal{P}_T$ is plotted in Fig. 7, for each value of the coordination delay T. This is in agreement with our result that the average packet delay increases linearly with the coordination latency between base stations.

VII. Conclusion

In this work, we considered multi-base-station wireless downlink scheduling with slow, global coordination and limited, local channel state information. We characterized the network stability region under this information structure, and developed a throughput-optimal distributed scheduling algorithm in which it is sufficient for base stations to share delayed queue lengths on a slow timescale to pick appropriate subsets of users, and use the locally observed channel states of these users to make good scheduling decisions. In this way, coordination between the base stations on a slow timescale—in


λ_2
$(0,\frac{5}{8})$
$(\frac{1}{8},\frac{5}{8})$
$(\frac{1}{8},\frac{1}{2})$
$(\frac{1}{4},\frac{1}{2})$
λ_1
$(\frac{1}{4},0)$
λ_3
$(0,\frac{3}{8})$
$(\frac{1}{8},\frac{3}{8})$
$(\frac{1}{8},\frac{1}{4})$
$(\frac{1}{4},\frac{1}{4})$
λ_1
$(\frac{1}{4},0)$
λ_2
$(0,\frac{5}{8},0)$
$(\frac{1}{8},\frac{1}{2},0)$
$(0,0,\frac{3}{8})$
$(\frac{1}{4},0,0)$
$(\frac{1}{8},0,\frac{1}{4})$
λ_3
λ_1

(a)

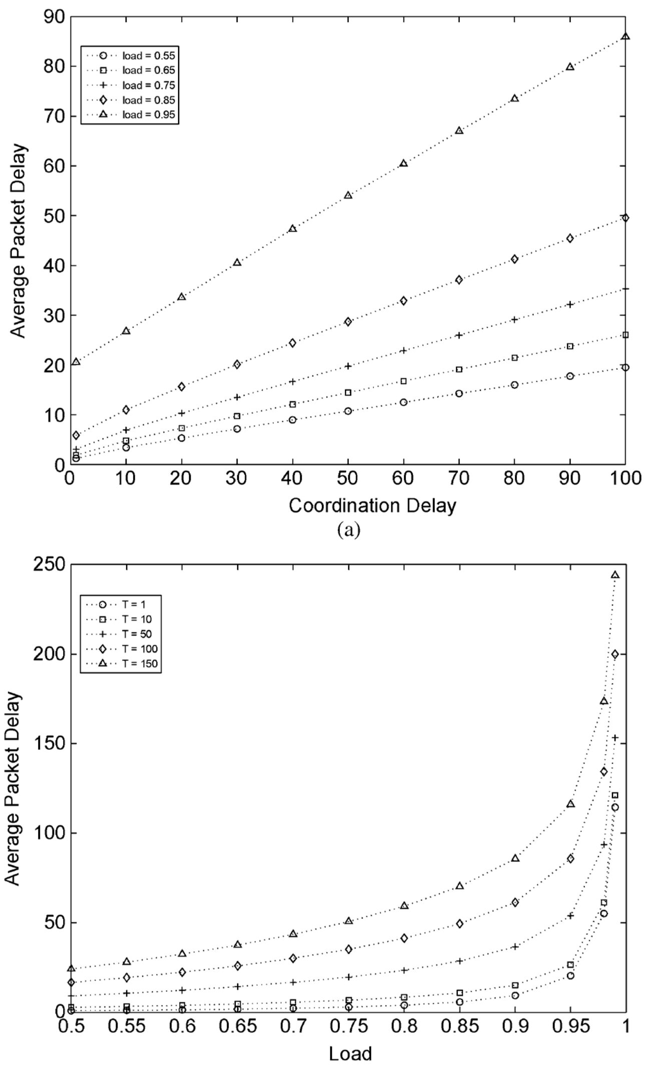
(b)

(c)

Fig. 4. (a) Stability region for users u_1 and u_2, (b) stability region for users u_1 and u_3, (c) stability region for all three users.

the form of delayed queue lengths—helps solve the subset-selection problem at each base station and, together with the right rules for scheduling users in those subsets, achieves throughput-optimality. We also investigated the impact of the delay in shared queue length information on the average packet delay performance of the system.

Future directions of research include: (*i*) evaluating the throughput performance of greedy, low-complexity scheduling strategies, and (*ii*) refining packet delay estimates using large-deviations/heavy-traffic analysis.



(a)

(b)

Fig. 5. (a) Average packet delay with lag T for various loads, (b) Average packet delay with load for various lags T.

Appendix A
Proof of Theorem 1

For showing necessity, assume that there exists a scheduling policy $\mathcal{P}$ which supports the arrival rate vector $\lambda = (\lambda_1,\ldots,\lambda_M)$. This means that under $\mathcal{P}$, the vector $X_R(t)$ is a positive recurrent discrete time Markov chain. Consider this Markov chain in its stationary regime (arbitrarily close approximations to the stationary regime will also suffice).

We will need the following additional notation for the proof:

1) Let $O(t) \triangleq (O_{b_1}(t),\ldots,O_{b_N}(t))$ be a representation of the collection of user subsets that each base station picks to observe at time slot t.
2) For user subsets $W_1,\ldots,W_N$, and x in the support of $X_R(t)$, let $\phi_{W_1,\ldots,W_N;x}(t) \triangleq \mathbb{P}[O_{b_1}(t) = W_1,\ldots,O_{b_N}(t) = W_N, X_R(t) = x]$.
3) Recall that the scheduling decision $B_j(t)$ for user $u_j \in \mathcal{U}(b_i)$ is a function of system state $X_R(t)$, the subset $O_{b_i}(t)$ chosen by its server (since users outside this subset are not scheduled), and current channel states $C_{O_{b_i}}(t)$. To indicate this, we explicitly write

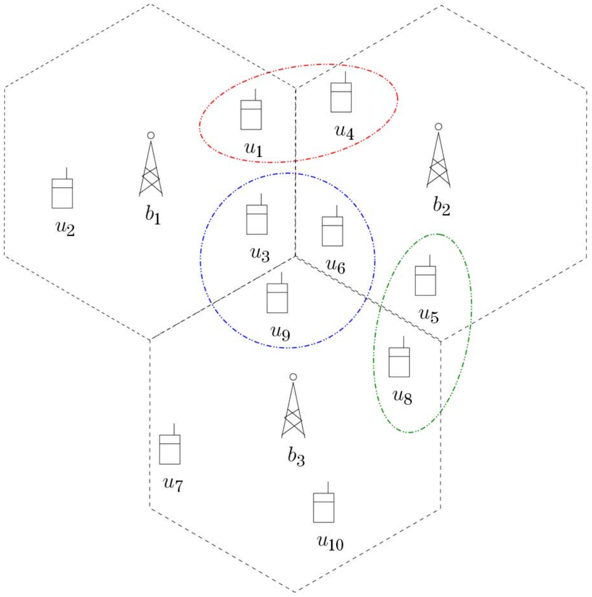


Fig. 6. Example of a system with 3 base stations and 10 users for numerical simulation. Interfering user groups, indicated by colored ellipses, are $\{u_1, u_4\}$, $\{u_3, u_6, u_9\}$ and $\{u_5, u_8\}$. Channels are Bernoulli (0.7) for u_1, u_5 and u_9 and constant 1 for all other users. Each base station can observe channel states for 3 of its users, which means b_3 must choose 3 of its 4 users at each time.

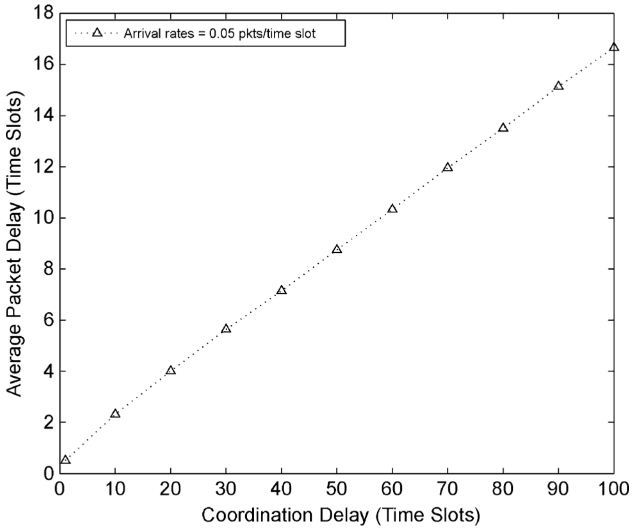


Fig. 7. Average packet delay of the scheduling algorithm $\mathcal{P}_T$ for various values of coordination delay T, for the 10-user 3-base station example.

$B_j(t) = f_j^t(X_R(t), O_{b_i}(t), C_{O_{b_i}}(t))$, where for every $j = 1, \ldots, M$ and $t = 1, 2, \ldots$, f_j^t is a function that maps $(X_R(t), O_{b_i}(t), C_{O_{b_i}}(t))$ into {0, 1}, with $f_j^t(X_R(t), O_{b_i}(t), C_{O_{b_i}}(t)) = 0$ whenever $j \notin O_{b_i}(t)$.

Let $u_j \in \mathcal{U}(b_i)$. To begin, note that at stationarity, $\mathbb{E}[Q_j(t+1)] = \mathbb{E}[Q_j(t)] \Rightarrow \mathbb{E}[A_j(t)] = \mathbb{E}[E_j(t)]$, and so

$$\begin{aligned}\lambda_j &= \mathbb{E}\left[A_j(t)\right] = \mathbb{E}\left[E_j(t)\right] \le \mathbb{E}\left[F_j(t)\right] \qquad (9)\\ &= \mathbb{E}\left[C_j(t)B_j(t)\prod_{u_k \in \mathcal{I}(u_j)}(1 - B_k(t))\right]\\ &= \sum_{\substack{W_1,\ldots,W_N\\ x}} \phi_{W_1,\ldots,W_N;x}(t)\mathbb{E}\end{aligned}$$

$$\times \left[C_j(t)B_j(t) \times \prod_{u_k \in \mathcal{I}(u_j)}(1 - B_k(t)) \middle| O(t) = (W_1, \ldots, W_N), X_R(t) = x\right]. \qquad (10)$$

Evaluating the expectation gives

$$\begin{aligned}&\mathbb{E}\left[C_j(t)B_j(t)\prod_{u_k \in \mathcal{I}(u_j)}(1 - B_k(t)) \middle| O(t) = (W_1, \ldots, W_N), X_R(t) = x\right]\\ &= \sum_{r \equiv (r_1,\ldots,r_M)} \times \mathbb{P}\left[C(t) = (r_1, \ldots, r_M) | O(t) = (W_1, \ldots, W_N)\right]\\ &\quad \times \mathbb{E}\left[C_j(t)B_j(t)\prod_{u_k \in \mathcal{I}(u_j)}(1 - B_k(t)) \middle| O(t) = (W_1, \ldots, W_N), X_R(t) = x, C(t) = (r_1, \ldots, r_M)\right]\\ &= \sum_{r \equiv (r_1,\ldots,r_M)} \pi(r_1, \ldots, r_M) r_j f_j^t\left(x, W_i, r|_{W_i}\right)\\ &\quad \times \prod_{u_k \in \mathcal{I}(u_j)}\left(1 - f_k^t\left(x, W_{\mathcal{B}(u_k)}, r|_{W_{\mathcal{B}(u_k)}}\right)\right) \qquad (11)\end{aligned}$$

since $C(t)$ is independent of $O(t)$. Using (11), (10) finally becomes

$$\begin{aligned}\lambda_j \le& \sum_{W_1,\ldots,W_N,x} \phi_{W_1,\ldots,W_N;x}(t)\\ &\times \left(\sum_{r \equiv (r_1,\ldots,r_M)} \pi(r_1, \ldots, r_M) r_j f_j^t\left(x, W_i, r|_{W_i}\right)\right.\\ &\left.\times \prod_{u_k \in \mathcal{I}(u_j)}\left(1 - f_k^t\left(x, W_{\mathcal{B}(u_k)}, r|_{W_{\mathcal{B}(u_k)}}\right)\right)\right). \qquad (12)\end{aligned}$$

The fact that

$$\sum_{W_1,\ldots,W_N,x} \phi_{W_1,\ldots,W_N;x}(t) = 1$$

together with the fact that (12) has the same form as that of (6) (relying in turn on (5)) for the long term rates of STS scheduling policies, shows that the vector $\lambda = (\lambda_1, \ldots, \lambda_M)$ can be dominated by a convex combination of rate vectors of SSS scheduling policies. This finishes the proof of the theorem. ■

Appendix B
Proof of Theorem 2

To avoid heavy notation, we prove the theorem assuming that each base station can pick all its users in the first scheduling step, viz. $\mathcal{O}_{b_i} = \{\mathcal{U}(b_i)\}\ \forall i = 1, \ldots, N$. The extension to the general case is straightforward.

First, we bound the amount that any queue in the system can grow in T time slots, using (1):

Lemma 1:

$$Q_j(t+T) \leq \max\left\{Q_j(t) - \sum_{\tau=t}^{t+T-1} F_j(\tau), 0\right\} + \sum_{\tau=t}^{t+T-1} A_j(\tau). \tag{13}$$

Proof: Consider two cases:

1) $Q_j(t) \geq \sum_{\tau=t}^{t+T-1} F_j(\tau)$: In this case, according to (1), both sides of (13) are equal.
2) $Q_j(t) < \sum_{\tau=t}^{t+T-1} F_j(\tau)$: For this case, let $t' \in \{t, \ldots, t+T-2\}$ be the first time that $Q_j(t') - F_j(t') < 0$ (if no such time exists, then $Q_j(t+T) = Q_j(t) - \sum_{\tau=t}^{t+T-1} F_j(\tau) + \sum_{\tau=t}^{t+T-1} A_j(\tau) \leq \sum_{\tau=t}^{t+T-1} A_j(\tau)$ and we are done). We must then have

$$Q_j(t+T) \leq \sum_{\tau=t'}^{t+T-1} A_j(\tau) \leq \sum_{\tau=t}^{t+T-1} A_j(\tau)$$

which finishes the proof. ■

Next, for the Markov chain $(X_T(t))_{t=1}^{\infty}$, let us introduce the quadratic Lyapunov function

$$L\left(X_T(t)\right) \triangleq \sum_{j=1}^{M} Q_j^2(t).$$

In what follows, we bound the expected drift in this Lyapunov function over an interval of T time slots when the system operates under the policy $\mathcal{P}_T$, and show that the expected drift can be bounded negatively away from zero. Consider

$$\begin{aligned}
&\Delta L\left(X_T(kT)\right)\\
&\quad \triangleq L\left(X_T\left((k+1)T\right)\right) - L\left(X_T(kT)\right)\\
&\quad = \sum_{j=1}^{M}\left(Q_j^2(kT+T) - Q_j^2(kT)\right)\\
&\quad \overset{(a)}{\leq} \sum_{j=1}^{M}\left(\left(\sum_{\tau=KT}^{kT+T-1} F_j(\tau)\right)^2 + \left(\sum_{\tau=KT}^{kT+T-1} A_j(\tau)\right)^2\right.\\
&\qquad\qquad \left. - 2Q_j(kT)\sum_{\tau=KT}^{kT+T-1}\left[F_j(\tau) - A_j(\tau)\right]\right)\\
&\quad \leq \sum_{j=1}^{M}\left(T^2 C_{\max}^2 + T^2 A_{\max}^2 - 2Q_j(kT)\right.\\
&\qquad\qquad \left. \times \sum_{\tau=KT}^{kT+T-1}\left[F_j(\tau) - A_j(\tau)\right]\right)\\
&\quad = M\left(T^2 C_{\max}^2 + T^2 A_{\max}^2\right)\\
&\qquad - 2\sum_{j=1}^{M} Q_j(kT) \sum_{\tau=KT}^{kT+T-1}\left[F_j(\tau) - A_j(\tau)\right]
\end{aligned}$$

where (a) follows from the fact that if V, U, μ, A are nonnegative real numbers with $V \leq \max\{U - \mu, 0\} + A$, then $V^2 \leq U^2 + \mu^2 + A^2 - 2U(\mu - A)$. Taking conditional expectations given $Q(kT) = q \equiv (q_1, \ldots, q_M)$ yields

$$\begin{aligned}
&\mathbb{E}\left[\Delta L\left(X_T(kT)\right) | Q(kT) = q\right]\\
&\quad \leq MT^2\left(C_{\max}^2 + A_{\max}^2\right)\\
&\qquad - 2\sum_{j=1}^{M} q_j \mathbb{E}\left[\sum_{\tau=KT}^{kT+T-1}\left[F_j(\tau) - A_j(\tau)\right] \middle| Q(kT) = q\right]\\
&\quad = MT^2\left(C_{\max}^2 + A_{\max}^2\right) + 2T\sum_{j=1}^{M} q_j \lambda_j\\
&\qquad - 2\sum_{j=1}^{M} q_j \mathbb{E}\left[\sum_{\tau=KT}^{kT+T-1} F_j(\tau) \middle| Q(kT) = q\right]\\
&\quad = MT^2\left(C_{\max}^2 + A_{\max}^2\right) + 2T\sum_{j=1}^{M} q_j \lambda_j\\
&\qquad - 2\sum_{j=1}^{M} q_j T \mathbb{E}\left[F_j(kT) \middle| Q(kT) = q\right]
\end{aligned} \tag{14}$$

where the last line follows because by definition, the scheduling choices of the policy $\mathcal{P}_T$ from time kT upto $kT+T-1$ depend only on the queue lengths $Q(kT)$ at time kT and the optimal binary vectors $z_1^*, \ldots, z_M^*$ computed at time slot kT, and are thus statistically identical from time kT upto $kT+T-1$.

By hypothesis, $\lambda = (\lambda_1, \ldots, \lambda_M) \in \mathcal{R}$. Hence, there exists $\epsilon > 0$ and a static time-sharing scheduling policy $\mathcal{P}_{TS}$ such that $\mu^{\mathcal{P}_{TS}} = (1+\epsilon)\lambda$. Let $\mathcal{P}_{TS}$ be a time-sharing (Bernoulli) combination of n SSS policies $\mathcal{P}_i$ with selection probabilities ϕ^i, respectively, $i = 1, \ldots, n$, where $\mathcal{P}_i = (W_1^i, \ldots, W_N^i, z_1^i, \ldots, z_N^i)$ (the superscript i indexes the SSS policy). We have, for $1 \leq j \leq M$

$$\begin{aligned}
\mu_j^{\mathcal{P}_{TS}} = \sum_{i=1}^{n} \phi_i \sum_{\substack{r \equiv (r_1, \ldots, r_M)\\ b_h = \mathcal{B}(u_j)}} \pi(r_1, \ldots, r_M) \cdot r_j \cdot \left(z_h\left(r|_{W_h^i}\right)\right)_j&\\
\times \prod_{\substack{u_k \in \mathcal{I}(u_j)\\ b_l = \mathcal{B}(u_k)}} \left(1 - \left(z_l\left(r|_{W_l}\right)\right)_k\right).&
\end{aligned} \tag{15}$$

We add and subtract $2T\sum_{j=1}^{M} q_j \mu_j^{\mathcal{P}_{TS}}$ to the right hand side of (14) to get

$$\begin{aligned}
&\mathbb{E}\left[\Delta L\left(X_T(kT)\right) | Q(kT) = q\right]\\
&\quad \leq MT^2\left(C_{\max}^2 + A_{\max}^2\right) + 2T\sum_{j=1}^{M} q_j\left(\lambda_j - \mu_j^{\mathcal{P}_{TS}}\right)\\
&\qquad + 2T\left(\sum_{j=1}^{M} q_j \mu_j^{\mathcal{P}_{TS}} - \sum_{j=1}^{M} q_j \mathbb{E}\left[F_j(kT) | Q(kT) = q\right]\right).
\end{aligned} \tag{16}$$

The crucial observation here is that the scheduling policy $\mathcal{P}_T$ is designed such that the last term above, in round brackets, is *always non-positive*:

Lemma 2: For the static time-sharing policy $\mathcal{P}_{TS}$, we have

$$\left(\sum_{j=1}^{M} q_j \mu_j^{\mathcal{P}_{TS}} - \sum_{j=1}^{M} q_j \mathbb{E}\left[F_j(kT) | Q(kT) = q\right]\right) \leq 0.$$

Proof: With the long term rates of the STS policy $\mathcal{P}_{TS}$ satisfying (15), we can write

$$\begin{aligned}
&\left(\sum_{j=1}^{M} q_j \mu_j^{\mathcal{P}_{TS}} - \sum_{j=1}^{M} q_j \mathbb{E}\left[F_j(kT) | Q(kT) = q\right]\right)\\
&= \Bigg(\sum_{j=1}^{M} q_j \sum_{i=1}^{n} \phi_i \sum_{\substack{r \equiv (r_1,\ldots,r_M)\\ b_h = \mathcal{B}(u_j)}} \\
&\quad \times \Bigg[\pi(r_1,\ldots,r_M) \cdot r_j \cdot \left(z_h\left(r|_{W_h^i}\right)\right)_j \\
&\quad \times \prod_{\substack{u_k \in \mathcal{I}(u_j)\\ b_l = \mathcal{B}(u_k)}} \left(1 - \left(z_l\left(r|_{W_l}\right)\right)_k\right)\Bigg] \\
&\quad - \sum_{j=1}^{M} q_j \mathbb{E}\left[F_j(kT) | Q(kT) = q\right]\Bigg) \\
&\leq \max_{i=1,\ldots,n} \Bigg(\sum_{j=1}^{M} q_j \sum_{\substack{r \equiv (r_1,\ldots,r_M)\\ b_h = \mathcal{B}(u_j)}} \pi(r_1,\ldots,r_M) \cdot r_j \\
&\quad \cdot \left(z_h\left(r|_{W_h^i}\right)\right)_j \times \prod_{\substack{u_k \in \mathcal{I}(u_j)\\ b_l = \mathcal{B}(u_k)}} \left(1 - \left(z_l\left(r|_{W_l}\right)\right)_k\right) \\
&\quad - \sum_{j=1}^{M} q_j \mathbb{E}\left[F_j(kT) | Q(kT) = q\right]\Bigg). \qquad (17)
\end{aligned}$$

We also have, by the definition of our proposed scheduling policy via (7), that

$$\begin{aligned}
&\sum_{j=1}^{M} q_j \mathbb{E}\left[F_j(kT) | Q(kT) = q\right] \\
&= \sum_{j=1}^{M} q_j \sum_{\substack{r \equiv (r_1,\ldots,r_M)\\ b_h = \mathcal{B}(u_j)}} \pi(r_1,\ldots,r_M) \cdot r_j \cdot \left(z_h^*(r)\right)_j \\
&\quad \times \prod_{\substack{u_k \in \mathcal{I}(u_j)\\ bl = \mathcal{B}(u_k)}} \left(1 - \left(z_l^*(r)\right)_l\right) \\
&\geq \sum_{j=1}^{M} q_j \sum_{\substack{r \equiv (r_1,\ldots,r_M)\\ b_h = \mathcal{B}(u_j)}} \pi(r_1,\ldots,r_M) \cdot r_j \cdot \left(z_h\left(r|_{W_h^i}\right)\right)_j \\
&\quad \times \prod_{\substack{u_k \in \mathcal{I}(u_j)\\ b_l = \mathcal{B}(u_k)}} \left(1 - \left(z_l\left(r|_{W_l}\right)\right)_k\right) = \sum_{j=1}^{M} q_j \mu_j^{\mathcal{P}_{TS}}
\end{aligned}$$

where the last line is due to the optimal choice of the z_h^*, $h = 1, \ldots, N$. Together with (17), this proves the lemma. ■

Using Lemma 2, (16) implies

$$\begin{aligned}
&\mathbb{E}\left[\Delta L\left(X_T(kT)\right) | Q(kT) = q\right] \\
&\leq MT^2\left(C_{\max}^2 + A_{\max}^2\right) + 2T \sum_{j=1}^{M} q_j \left(\lambda_j - \mu_j^{\mathcal{P}_{TS}}\right) \\
&= MT^2\left(C_{\max}^2 + A_{\max}^2\right) - 2\epsilon T \sum_{j=1}^{M} q_j \lambda_j \\
&\leq MT^2\left(C_{\max}^2 + A_{\max}^2\right) - 2\epsilon T (\min_j \lambda_j) \sum_{j=1}^{M} q_j. \qquad (18)
\end{aligned}$$

Without loss of generality, $\min_j \lambda_j > 0$. For a fixed $\delta > 0$, outside the finite set of vectors q for which $\sum_{j=1}^{M} q_j < ((\delta + MT^2(C_{\max}^2 + A_{\max}^2))/2\epsilon T(\min_j \lambda_j))$, we have $\mathbb{E}[\Delta L(X_T(kT)) | Q(kT) = q] \leq -\delta < 0$, so by Foster's theorem [29], $\{X_T(kT)\}_k$ is a positive recurrent Markov chain. This proves the first part of the theorem.

Turning to the second part, we can take expectations of both sides of (18) and sum over $k = 0, \ldots, K-1$, so that

$$\begin{aligned}
\mathbb{E}\left[L\left(X_T(KT)\right)\right] - \mathbb{E}\left[L\left(X_T(0)\right)\right] \leq MKT^2\left(C_{\max}^2 + A_{\max}^2\right) \\
-2\epsilon T (\min_j \lambda_j) \sum_{k=0}^{K-1} \sum_{j=1}^{M} \mathbb{E}\left[Q_j(kT)\right].
\end{aligned}$$

Rearranging terms and noting that $L(X_T(KT)) \geq 0$ gives

$$\begin{aligned}
\frac{1}{K} \sum_{k=0}^{K-1} \sum_{j=1}^{M} \mathbb{E}\left[Q_j(kT)\right] \leq \frac{MT\left(C_{\max}^2 + A_{\max}^2\right)}{2\epsilon(\min_j \lambda_j)} \\
+ \frac{\mathbb{E}\left[L\left(X_T(0)\right)\right]}{2\epsilon KT(\min_j \lambda_j)}.
\end{aligned}$$

The positive recurrence of $\{X_T(kT)\}_k$, from the previous part, implies that

$$\lim_{K \to \infty} \frac{1}{K} \sum_{k=0}^{K-1} \sum_{j=1}^{M} \mathbb{E}\left[Q_j(kT)\right] = \sum_{j=1}^{M} \mathbb{E}^{\pi}\left[Q_j(lT)\right]$$

for any l, and, together with the finiteness of $\mathbb{E}[L(X_T(0))]$, we get that

$$\sum_{j=1}^{M} \mathbb{E}^{\pi}\left[Q_j(lT)\right] \leq \frac{MT\left(C_{\max}^2 + A_{\max}^2\right)}{2\epsilon(\min_j \lambda_j)} \leq \alpha T$$

for any $\alpha \geq M(C_{\max}^2 + A_{\max}^2)/2\epsilon(\min_j \lambda_j)$. This proves the second part of the theorem. ■

References

[1] Motorola, "Long term evolution (LTE): Overview of LTE air-interface," *White Paper*, 2007 [Online]. Available: http://www.motorola.com/web/Business/Solutions/Industry%20Solutions/Service%20Providers/Wireless%20Operators/LTE/_Document/Static%20Files/6993_MotDoc_New.pdf

[2] D. Lopez-Perez, A. Valcarce, G. de la Roche, and J. Zhang, "OFDMA femtocells: A roadmap on interference avoidance," *IEEE Commun. Mag.*, vol. 47, no. 9, pp. 41–48, Sep. 2009.

[3] R. Madan, A. Sampath, N. Bhushan, A. Khandekar, J. Borran, and T. Ji, "Impact of coordination delay on distributed scheduling in LTE-A femtocell networks," in *Proc. GLOBECOM*, 2010, pp. 1–5.

[4] R. Madan, A. Sampath, A. Khandekar, J. Borran, and N. Bhushan, “Distributed interference management and scheduling in LTE-A femto networks,” in *Proc. GLOBECOM*, 2010, pp. 1–5.

[5] 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. 4, pp. 1936–1948, Dec. 1992.

[6] L. Tassiulas and A. Ephremides, “Dynamic server allocation to parallel queues with randomly varying connectivity,” *IEEE Trans. Inform. Theory*, vol. 39, pp. 466–478, Mar. 1993.

[7] M. Andrews, K. Kumaran, K. Ramanan, A. L. Stolyar, R. Vijayakumar, and P. Whiting, “Scheduling in a queueing system with asynchronously varying service rates,” *Prob. Eng. Informational Sci.*, vol. 14, pp. 191–217, 2004.

[8] S. Shakkottai and A. Stolyar, *Scheduling For Multiple Flows Sharing a Time-Varying Channel: The Exponential Rule*. Providence, RI: American Mathematical Society, 2002, vol. 7, pp. 185–201.

[9] A. Eryilmaz, R. Srikant, and J. R. Perkins, “Stable scheduling policies for fading wireless channels,” *IEEE/ACM Trans. Networking*, vol. 13, pp. 411–424, 2005.

[10] X. Lin and S. B. Rasool, “Constant-time distributed scheduling policies for ad hoc wireless networks,” in *Proc. Conf. Decision Control*, 2006, pp. 1258–1263.

[11] X. Wu and R. Srikant, “Scheduling efficiency of distributed greedy scheduling algorithms in wireless networks,” in *Proc. IEEE INFOCOM*, 2006, pp. 1–12.

[12] A. Eryilmaz, A. Ozdaglar, and E. Modiano, “Polynomial complexity algorithms for full utilization of multi-hop wireless networks,” in *Proc. IEEE INFOCOM*, 2007, pp. 499–507.

[13] S. Sanghavi, L. Bui, and R. Srikant, “Distributed link scheduling with constant overhead,” in *Proc. ACM SIGMETRICS*, 2007, pp. 313–324.

[14] M. J. Neely, “Dynamic Power Allocation and Routing for Satellite and Wireless Networks With Time Varying Channels,” Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge, MA, November 2004.

[15] L. Georgiadis, M. J. Neely, and L. Tassiulas, “Resource allocation and cross-layer control in wireless networks,” *Foundations and Trends in Networking*, vol. 1, pp. 1–144, 2006.

[16] 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.

[17] N. Chang and M. Liu, “Optimal channel probing and transmission scheduling for opportunistic spectrum access,” in *Proc. ACM Int. Conf. Mobile Comp. Netw. (MobiCom)*, Montreal, QC, Canada, Sep. 2007, pp. 27–38.

[18] A. Gopalan, C. Caramanis, and S. Shakkottai, “On wireless scheduling with partial channel state information,” *IEEE Trans. Inform. Theory*, vol. 58, no. 1, pp. 403–420, Jan. 2012.

[19] A. Pantelidou, A. Ephremides, and A. Tits, “Joint scheduling and routing for ad-hoc networks under channel state uncertainty,” in *Proc. Int. Symp. Modeling Optim. Mobile, Ad-Hoc Wireless Networks (WiOpt)*, Apr. 2007, pp. 1–8.

[20] M. Ouyang and L. Ying, “On scheduling in multi-channel wireless downlink networks with limited feedback,” in *Proc. Annu. Allerton Conf. Commun., Control Comp.*, Piscataway, NJ, USA, 2009, pp. 455–461, IEEE Press.

[21] L. Ying and S. Shakkottai, “On throughput optimality with delayed network-state information,” *IEEE Trans. Inform. Theory*, vol. 57, no. 8, pp. 5116–5132, Aug. 2011.

[22] L. Ying and S. Shakkottai, “Scheduling in mobile ad hoc networks with topology and channel-state uncertainty,” *IEEE Trans. Autom. Control*, to be published.

[23] A. Reddy, S. Banerjee, A. Gopalan, S. Shakkottai, and L. Ying, “On wireless scheduling with heterogeneously delayed network-state information,” *Queueing Syst.*, vol. 72, pp. 193–218, Dec. 2012.

[24] S. Das, H. Viswanathan, and G. Rittenhouse, “Dynamic load balancing through coordinated scheduling in packet data systems,” in *Proc. IEEE INFOCOM*, 1, Mar./Apr. 2003, vol. 1, pp. 786–796.

[25] A. Stolyar and H. Viswanathan, “Self-organizing dynamic fractional frequency reuse for best-effort traffic through distributed inter-cell coordination,” in *Proc. IEEE INFOCOM*, Apr. 2009, pp. 1287–1295.

[26] G. Fodor, M. Telek, and C. Koutsimanis, “Performance analysis of scheduling and interference coordination policies for OFDMA networks,” *Comput. Netw.*, vol. 52, no. 6, pp. 1252–1271, Jun. 2008.

[27] B. Rengarajan and G. de Veciana, “Architecture and abstractions for environment and traffic aware system-level coordination of wireless networks: The downlink case,” in *Proc. IEEE INFOCOM*, Apr. 2008, pp. 502–510.

[28] S. Shakkottai, R. Srikant, and A. L. Stolyar, “Pathwise optimality of the exponential scheduling rule for wireless channels,” *Adv. Appl. Prob.*, vol. 36, no. 4, pp. 1021–1045, 2004.

[29] P. Brémaud, *Markov Chains: Gibbs Fields, Monte Carlo Simulation, and Queues*, ser. Texts in applied mathematics. New York: Springer, 1999.

Aditya Gopalan received the Ph.D. degree from the Electrical and Computer Engineering Department, University of Texas at Austin in 2011.

He is a Postdoctoral Fellow with the Department of Electrical Engineering, Technion—Israel Institute of Technology, Haifa. His current research interests include network algorithms, stochastic control and machine learning.

Constantine Caramanis (M’06) received the Ph.D. degree in electrical engineering and computer science from the Massachusetts Institute of Technology, Cambridge, in 2006.

Since then, he has been on the faculty in Electrical and Computer Engineering, The University of Texas at Austin. His current research interests include robust and adaptable optimization, machine learning and high-dimensional statistics, with applications to large scale networks.

Dr. Caramanis received the NSF CAREER award in 2011.

Sanjay Shakkottai (M’02–SM’11) received the Ph.D. degree from the Electrical and Computer Engineering Department, University of Illinois at Urbana-Champaign in 2002.

He is with The University of Texas at Austin, where he is currently a Professor and the Engineering Foundation Centennial Teaching Fellow in the Department of Electrical and Computer Engineering. His current research interests include network architectures, algorithms and performance analysis for wireless and sensor networks.

Dr. Shakkottai received the NSF CAREER award in 2004.