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

Aditya Gopalan, Constantine Caramanis and Sanjay Shakkottai
Department of Electrical and Computer Engineering,
The University of Texas at Austin
Email: {gadit,caramanis,shakkott}@mail.utexas.edu

***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. If 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 the base stations to have complete 'instantaneous' channel-state information for all its own users, as well as the ability to communicate on the same time-scale 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 periodically (on a slower time-scale using a wired back-haul) communicate with other base stations in order to loosely coordinate scheduling decisions.**

In this setting, we first characterize the throughput capacity of the system. A key observation is that sharing (delayed) queue-length information enables loose coordination 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 own user queues. Based on this, we develop a distributed queue-aware scheduling (and information exchange) algorithm that provably stabilizes the system. We finally study the effect of inter-base-station coordination delay on the system packet delay performance.

I. Introduction

Next-generation cellular systems like 3GPP Long Term Evolution (LTE) [1] are based on the technique of Orthogonal Frequency Division Multiple Access (OFDMA), and promise high-speed packet-switched services for a variety of applications. In a typical downlink of such a system, base stations enable multiple mobile users to share common channel resources by assigning them different frequency bands or 'tones'. Base stations obtain channel quality measurements from the mobiles attached to them on a timescale of about every 2 ms; this helps them exploit local channel fluctuations to better schedule data transmissions to the mobiles.

A multicellular environment with several base stations, however, is prone to suffer from intercell interference, where transmissions to neighboring mobile users assigned the same frequencies in different cells collide resulting in loss of throughput. Static frequency planning among a collection of base stations helps reduce interference between such users at the cell edges. Yet, even after frequency planning and allocation, a significant amount of interference can persist between users. The fact that the base stations are connected through a common wired backhaul with low-bandwidth (which allows only limited communication on a slower timescale compared to channel scheduling) can then potentially be used to coordinate scheduling decisions across base stations. This slower and limited coordination timescale does not permit instantaneous channel state information to be shared across base stations but allows other slowly varying information to be propagated. An additional issue is that base stations cannot acquire even the complete 'local' instantaneous channel states for all their own users. This is because OFDM systems such as LTE have many sub-channels, and getting channel state information for all sub-channels for each user in every time-slot may be prohibitive in terms of feedback bandwidth.

How to effectively use this combination of network state information available to base stations – partial 'local' information such as instantaneous channel states from users on a fast timescale, and 'global' information such as channel statistics and accumulated queue lengths from other base stations on a slow timescale – to maximize throughput now becomes a challenging problem.

In this paper, we consider a collection of base stations each serving an exclusive set of users in a time-slotted system. Transmissions to interfering users collide if scheduled simultaneously. In every time slot, each base station picks a *subset* of its users and observes instantaneous channel states for those users in every time slot (thus, the base station has *partial local channel state*), and together with globally shared information (e.g., *delayed queue length state, channel statistics*) from other base stations schedules users in that subset. We first characterize the network throughput region under such a scenario. Using the key observation that common state information provided by global delayed queues allows coupling of decisions, we demonstrate the optimal way of using this coupled state in a multi-base-station scenario and develop a provably throughput-optimal scheduling algorithm. 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 queue lengths. We also examine how the packet delay performance of such an algorithm is related to the amount of delay in the shared queue length information, by way of analysis and simulations.

A. Main Contributions

The main contributions of this paper include the following:

1) We derive the throughput region of a system with coordinated scheduling using limited and local channel state information. Moreover, we show that the region can be parametrized using a simple class of static scheduling policies based on binary decision vectors for each user, in which each base station always picks a fixed subset of its users and for each user in that subset decides to schedule the user or not schedule it depending solely on its instantaneous channel state and a fixed binary vector associated with it.
2) We develop a two-tier distributed, throughput-optimal scheduling policy in which the base stations first communicate queue lengths of their respective users to each other once every T time slots where T is a fixed integer. The base stations then use this information along with knowledge about channel correlations and interference patterns to pick their user subsets and make scheduling decisions with limited local information in the next T slots.
3) We explore the relationship between the inter-base-station coordination delay, i.e., the 'staleness' of exchanged queue length information, and the system packet delay performance, and present analytical and simulation results to illustrate the degradation in the latter due to an increase in the former.

B. Related Work

Throughput-optimal scheduling for wireless networks dates back to the pioneering work of Tassiulas et al. [2], [3]. Since then, there has been much work on throughput-optimal wireless scheduling, both with a central scheduler having complete network-state information [4], [5], [6] and distributed implementations [7], [8], [9], [10]. Further references can be found in [11], [12]. Scheduling with partial or limited channel state information has been addressed in [13], where infrequent channel state information used to schedule, and [14], [15], [16] where scheduling is studied with partial or inaccurate observability of the aggregate channel state. In [17], [18], 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 [17], [18] 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 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 algorithms.

In the multiple base station setting, two-tiered interference mitigation through load balancing and base station coordination has been studied in [19], albeit 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 their channel state information. However, the authors do not investigate queue-stability or throughput-optimality. Further, as we will 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 (roughly because 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 [20] the authors propose a gradient power-control algorithm to mitigate intercell interference and dynamically reuse frequencies. [21] studies scheduling algorithms to effectively allocate subcarriers or frequencies to users in a multicell environment to maximize the sum throughput of the system. In [22] the authors assume coarse-grained communication among base stations along with a dynamic user model in which users enter and exit the network randomly, and extensively simulate scheduling strategies with the main metric being file transfer delay. None of these works, though, examine 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 [19] for a comprehensive survey.

II. A Motivating Example

In this section, we introduce an example to help understand how global coordination among base stations can improve the throughput region of a wireless system.

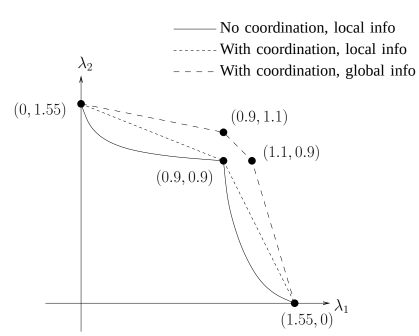

Fig. 1. Capacity regions under different scheduling information structures

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. Also, we assume that transmissions to the 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*). 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 Figure 1 (essentially, this is equivalent to saying that each base station decides independently to schedule its own user with some fixed probability). The first (resp. third) point represents the case when user u_1 (resp. 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.
2) Next, assume again that each base station knows only its own user's current channel state, but the base stations can additionally 'talk' and exchange their queue length values or other auxiliary information (but not their users' current channel states) before deciding to schedule their respective users. For instance they could toss a common coin and make their scheduling decisions depending on the outcome of the toss. This is the situation in which the base stations have *local channel state information with global coordination*, and we see that the throughput region expands to the convex hull of the earlier three points; intuitively *collaboration allows timesharing*.
3) Finally, if we assume the base stations can coordinate completely, i.e., get to know both users' channel states before making scheduling decisions (i.e., the base stations have *global channel state information*), the throughput region expands further to the convex hull of the points $(0, 1.55)$, $(0.9, 1.1)$, $(1.1, 0.9)$ and $(1.55, 0)$. This is because the second (resp. third) point can be achieved by scheduling *only* user u_1 (resp. u_2) when its channel state is 2 - the advantage in knowing the other user's channel state comes from the fact that a base station can 'back off' when both channels have state 2.

This example illustrates that there can be a significant difference in throughput depending on whether local or global channel state information or a combination of both is used. With this as a starting point, we develop a formal model of coordinated scheduling with local and limited channel state information and seek to understand what the throughput capabilities of such a system are, and what scheduling strategies can achieve maximum throughout.

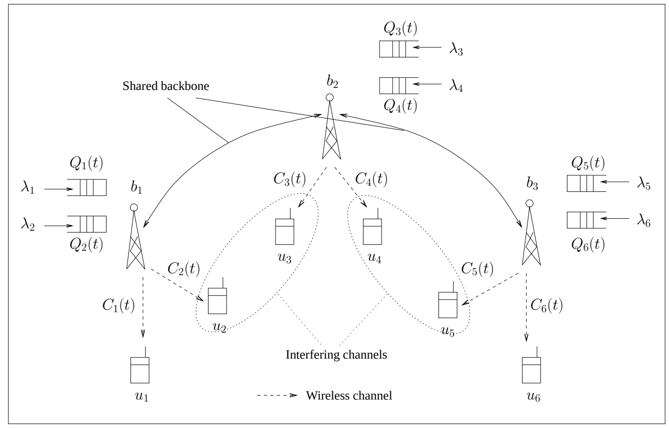

Fig. 2. Coordinated scheduling with local information

III. Model and Definitions

This section deals with establishing notation and definitions necessary for developing a formal model for coordinated wireless scheduling.

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.

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 $\mathbb{E}[A_j(t)] = \lambda_j$, $A_j(t) \leq A_{\max}$ $a.s.$. Let $A(t) \stackrel{\triangle}{=} (A_1(t), \ldots, A_M(t))$. The packets get queued if they are not immediately transmitted. The channel between user u_j and its associated base station is time-varying; -we denote the corresponding 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 $\mathcal{C} \subset \mathbb{Z}^+$. We explicitly assume that $\mathcal{C}$ consists of the values $c_1 < c_2 < \cdots < c_K = C_{\max}$. The aggregate channel state process $C(t) \stackrel{\triangle}{=} (C_j(t) : j = 1, \ldots, M)$ is assumed to be *iid* over time slots. Note however that the *channels can be correlated across users*. Let $\pi(\cdot)$ denote the probability mass function of the aggregate channel state $(C_1(t), \ldots, C_M(t))$, i.e., $\pi(r_1, \ldots, r_M) \stackrel{\triangle}{=} \mathbb{P}[C_1(t) = r_1, \ldots, C_M(t) = r_M]$, with $r_i \in \mathcal{C}$ for all i. Such a wireless system is shown in Figure 2.

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. When user $u_j \in \mathcal{U}(b_i)$ is *successfully scheduled* for data reception at time slot t (which we denote by letting a binary random variable $D_j(t) = 1$), 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

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

Another form of (1) which we will use later is

$$Q_j(t+1) = Q_j(t) + A_j(t) - E_j(t), \quad (2)$$

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.

Scheduling model: For each base station b_i, let a collection of subsets of $\mathcal{U}(b_i)$ - denoted by $\mathcal{O}(b_i)$ - be fixed. These subsets define which user channels the base station can observe and schedule in each time slot.

We model coordinated scheduling with local and limited channel state information in the network as follows. At any time slot t, let the *history* of the system be denoted by the random vector $H_T(t) \triangleq (Q(t-T), \ldots, Q(t), C(t-T), \ldots, C(t-1), A(t-T), \ldots, A(t-1))$ where $T \geq 1$ is a fixed integer throughout. $H_T(t)$ represents the cumulative queue length, channel state and arrival history for the previous T time slots. In addition assume that there exists a sequence of independent discrete random variables $G(t)$, $t = 1, 2, \ldots$ which can be thought of as a source of *common, auxiliary* scheduling information available globally to the base stations. Let the *state* of the system be defined by $X_T(t) \triangleq (H_T(t) : G(t-T), \ldots, G(t))$, i.e., a concatenation of the system history and auxiliary information history for the last T time slots. This is essentially the information that we allow each base station to know at the global level. For a random variable X and a σ-algebra $\mathcal{A}$, we write $X \in \mathcal{A}$ to mean that X is $\mathcal{A}$-measurable. A *scheduling policy* is defined as the following two-step procedure carried out at each time slot t:

1) Each base station b_i first picks a subset $O_{b_i}(t) \in \mathcal{O}(b_i)$ of its users depending on the system state (system history and auxiliary information history). Formally, we write $O_{b_i}(t) \in \sigma(X_T(t))$.
2) Let the chosen subset of users for b_i at time t be $O_{b_i}(t) = \{u_{j_1}, \ldots u_{j_L}\}$. For every user u_{j_l} in the chosen subset $O_{b_i}(t)$, base station b_i either schedules the user for data transmission or not by setting a decision variable $B_{j_l}(t)$ to 1 or 0 respectively. The remaining users associated to b_i are necessarily not scheduled, i.e., $B_j(t) = 0$ for all users $u_j \in \mathcal{U}(b_i) \backslash O_{b_i}(t)$. Furthermore, for each user u_{j_l} in the chosen subset $O_{b_i}(t)$, we require that $B_{j_l}(t) \in \sigma(X_T(t), C_{j_l}(t))$, i.e., the final scheduling decision for each user can potentially depend on *(i)* system history, *(ii)* auxiliary information in the past, *(iii)* whether the user is or is not in the subset chosen by its base station, and most importantly *(iv)* its currently observed channel state.

To summarize, every base station *first* picks a subset of its users to observe their current channel states, depending on queue lengths, channel states, arrivals and auxiliary information (global information) in the last T slots. After having observed channel states in that subset, it *next* decides whether to schedule those users depending on their current channel states (limited local information) and the global information it already has. We remark that this scheduling model captures two important aspects:

1) *Global coordination:* By letting the base stations pick subsets of users depending on the system history along with common randomness in Step 1, we essentially *let the base stations collaborate (using a common backhaul) to decide which subset each chooses for the current time slot*. This is a key feature in the scheduling model we use - instead of each base station deciding which subset of its users' channels it wants to locally sample independent of other base stations, it is allowed to 'loosely coordinate' with other basestations to choose its subset so as to further mitigate interference and improve throughput. The common auxiliary information $G(t)$ represents any other additional form of information that the base stations could use to decide which subsets to pick. Simple examples could be time-sharing depending on the time index, or time-sharing depending on a common 'coin toss'.
2) *Limited and local information:* Each base station must commit to choosing a subset of users in the first step whose channel states are then revealed to it, which models the fact that every base station has limited information about the channel states of its users. Further, each user whose channel state is observed in Step 1 is scheduled depending on common shared information and that users's current channel state. This says that the instantaneous channel state information available to the base station for scheduling is *local* - instantaneous channel states of *other* users are *not available* to a base station while scheduling its users.

Interference model: We model interference in the network by treating the aggregate channel between the base stations and users as a *collision channel*, i.e., we associate to each user $u_j \in \mathcal{M}$ its *interference set* $\mathcal{I}(u_j) \subset \mathcal{M}$, with the understanding that user u_j cannot receive any data packets in a time slot in which a user $u_k \in \mathcal{I}(u_j)$ is scheduled (we assume that $u_j \notin \mathcal{I}(u_j)$ for any j). We write this formally as

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

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

$$F_j(t) \triangleq C_j(t) D_j(t). \quad (4)$$

We remark that this 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.

Objective: 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. We say 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 $X_T(t)$ 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 $(\lambda_j : j = 1, \ldots, M)$ supported by at least onescheduling 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.

IV. The Stability Region

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 simpler class of scheduling policies that use just current local channel state information to make scheduling decisions, and show that the stability region can be described using just these policies. In other words, *any* scheduling policy can be thought of as mimicking some policy in this special class, in the sense of long-term service rates.

A. SSS Policies

We introduce a class of 'simple' scheduling policies which we will call *Static Service Split* (SSS) policies, inspired by [4]. Unlike the standard SSS policies used in literature in scheduling with complete channel state information, however, our SSS policies respect the two-tiered scheduling setup introduced in this work, and are seen to be specifications of *(i)* fixed subsets that base stations must always pick and *(ii)* fixed 'binary vectors' per user that dictate for which observed channel states the users must be scheduled.

Formally, an SSS policy $\mathcal{P}$ is defined by the tuple $\mathcal{P} = (W_1, \ldots, W_N, z_1, \ldots, z_M)$, where for each i, W_i is a permissible subset of users for base station b_i, and for each j, z_j is a binary decision vector of length $|\mathcal{C}|$. Equivalently, we will think of z_j as a map from $\mathcal{C}$ into $\{0, 1\}$. 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 to observe channel states for.
2) All the users for b_i not in W_i are not scheduled, i.e., $B_j(t) = 0$ for such users. For a user u_j in W_i, if its observed channel state is $C_j(t) = c$, u_j is scheduled if and only if $z_j(c) = 1$, i.e., $B_j(t) = z_j(c)$.

B. Static Time-sharing Policies

Extending the concept of an SSS policy to include a combination of SSS policies leads to the notion of a *Static Time-sharing* (STS) policy. An STS policy $\mathcal{P}$ is specified 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.

C. Characterization of the Stability Region

Towards an explicit characterization of the stability region, 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

$$\mu_j^{\mathcal{P}} \triangleq \mathbb{E}[F_j^{\mathcal{P}}(t)] = \sum_{r_1,\ldots,r_M} \Bigg[\pi(r_1, \ldots, r_M) \cdot r_j \cdot B_j(r_j) \times \prod_{u_k \in \mathcal{I}(u_j)} (1 - B_k(r_k)) \Bigg]. \tag{5}$$

Intuitively, $\mu_j^{\mathcal{P}}$ is the vector of long term 'service rates' that scheduling using the SSS policy $\mathcal{P}$ delivers to the flows to all the users in the system. In a similar manner, if $\mathcal{P}$ is an STS policy in that $\mathcal{P}$ is a combination of SSS policies $(\mathcal{P}_1, \ldots, \mathcal{P}_K)$ with weights $\phi_1, \ldots, \phi_K$, we define the rate vector $\mu^{\mathcal{P}}$ associated with $\mathcal{P}$ as

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

Essentially, the rate vector of an STS policy is just a convex combination of the rate vectors of its component SSS policies.

From the way SSS and STS policies are defined, they always choose fixed subsets and schedule users in those subsets using fixed binary decision vectors. Thus they are a class of valid scheduling policies. Hence, any arrival rate vector $\lambda = (\lambda_1, \ldots, \lambda_M)$ with $\lambda_i \geq 0, i = 1, \ldots, M$ dominated by such a vector $\mu_j^{\mathcal{P}}$ must be supported by some scheduling policy, namely $\mathcal{P}$ for one. Extending the argument, any arrival rate vector within the convex hull of all STS rate vectors must also be supported by some scheduling policy.

Let

$$\mathcal{R} \triangleq \text{int Co}(\{\mu^{\mathcal{P}} : \mathcal{P} \text{ an STS policy}\})$$

be the interior of the convex hull of the set of rate vectors corresponding to all possible STS scheduling policies. The heuristic argument in the previous paragraph 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). The following result formalizes that in fact, the stability region cannot be larger than $\mathcal{R}$:

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 a suitable time-sharing combination of SSS scheduling policies, in the sense of the long-term service rates it is able to provide. It will be useful when we are interested later in showing that a particular type of scheduling policy is throughput-optimal. The proof of this theorem is similar in spirit to the ones in [4], [17], [18] used to characterize the stability region, and intuitively relies on the fact that a system stable under a policy must have long term fractions which can be 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

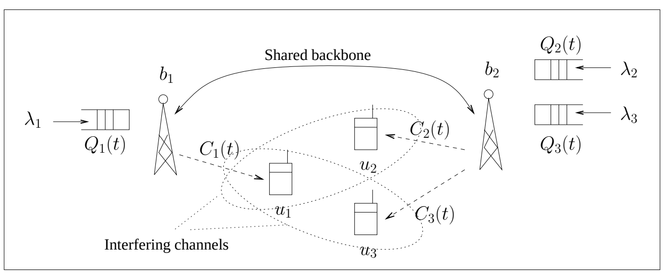

Fig. 3. Stability region example

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). Two users interfere if and only if they are associated to different base stations, i.e., $\mathcal{I}(u_1) = \{u_2\}$, $\mathcal{I}(u_2) = \mathcal{I}(u_3) = \{u_1\}$.

Let us assume that base station b_2 can pick at most one of its 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:

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	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{8}$

TABLE I
CHANNEL STATE DISTRIBUTION FOR 3-USER EXAMPLE

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 Figure 4. In this figure, the extreme points $(\frac{1}{4}, 0)$ and $(0, \frac{5}{8})$ 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 $(\frac{1}{8}, \frac{1}{2})$ 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 Figure 4 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.

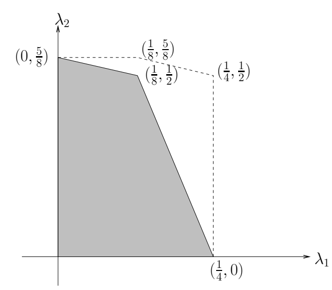

Fig. 4. Stability region for users u_1 and u_2

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 Figure 5. 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 Figure 6.

V. THROUGHPUT-OPTIMAL SCHEDULING

This section focuses on developing a throughput-optimal information exchange and scheduling policy in which base stations exploit infrequently exchanged queue length information from other base stations, together with instantaneous limited

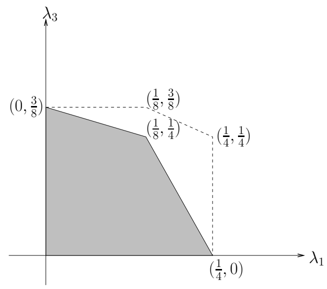

Fig. 5. Stability region for users u_1 and u_3

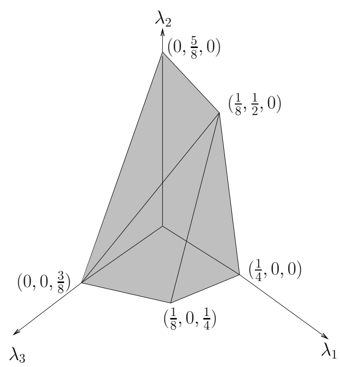

Fig. 6. Stability region for all three users

local information about their users' channels, to guarantee stability of the queues for any arrival rate in the stability region $\mathcal{R}$. We also show that in certain special cases of our network model, the throughput-optimal policy can be considerably simplified in form.

A. A Throughput-optimal Policy

Consider the following queue-length-aware, collaborative scheduling policy:

Scheduling policy $\mathcal{P}_T$:

- At every T-th timeslot $t = kT$ where $k \in \mathbb{Z}^+$,

  1) Each base station broadcasts the vector of queue lengths for its users to all other base stations.
  2) Let the global queue length vector at time slot kT be $Q(kT) = q \equiv (q_1, \ldots, q_M)$. Each base station b_i then solves the following (common) optimization problem:

$$\max_{\substack{S_{b_1},\ldots,S_{b_N},\\ z_1,\ldots,z_M}} \Bigg(\sum_{j=1}^{M} q_j \sum_{r_1,\ldots,r_M} \pi(r_1,\ldots,r_M) \times r_j z_j(r_j) \mathbb{1}_{u_j \in S_{\mathcal{B}(u_j)}} \times \prod_{u_k \in \mathcal{I}(u_j)} \Big(1 - z_k(r_k)\mathbb{1}_{u_k \in S_{\mathcal{B}(u_k)}}\Big) \Bigg) \quad (7)$$

$$\text{s.t.} \quad S_{b_i} \in \mathcal{O}(b_i),\ i = 1,\ldots,N,$$
$$z_j : \mathcal{C} \to \{0,1\},\ j = 1,\ldots,M.$$

- Let $(S^*_{b_1}, \ldots, S^*_{b_N}, z^*_1, \ldots, z^*_M)$ be a choice of arguments that solves (7). For the next T time slots, i.e., $t = kT, \ldots, kT+T-1$, each base station b_i picks $S^*_{b_i}$ as its chosen subset of users, and schedules a user u_j in that subset at time slot t if and only if $z^*_j(C_j(t)) = 1$.

In other words, the policy $\mathcal{P}_T$ works as follows:

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 service rates delivered to each queue weighted by its corresponding delayed queue length*.

The following theorem is a key result in this paper, and establishes that the scheduling policy $\mathcal{P}_T$ is in fact throughput-optimal *for any value of* T, i.e., it can support any arrival rate λ in the stability region $\mathcal{R}$:

Theorem 2. *For any value of T, the scheduling policy $\mathcal{P}_T$ is throughput-optimal, i.e., for an arrival rate vector $\lambda \in \mathcal{R}$, the system state $X_T(t)$ is a positive recurrent discrete time Markov chain under $\mathcal{P}_T$.*

The proof uses Lyapunov-function type arguments similar to the ones in [3], [13], [17], additionally incorporating the effect of the queue delay parameter T. Refer to Appendix B for the proof of the theorem.

B. Special Cases

The scheduling policy $\mathcal{P}_T$ introduced earlier was shown to be throughput-optimal under the network model described for this work. The network model is quite general in the sense of modeling scheduling with global system history and limited local channel states in a system with fading channels and inter-channel interference in the form of data collisions. In what follows, we consider some special cases of the network model and remark that the proposed scheduling policy $\mathcal{P}$ reduces to 'simpler'/known scheduling policies; in this way the scheduling model examined in this work naturally generalizes several ones in the scheduling literature.

1) Interference-free channels: When there is no (or practically negligible) interference or collision between any pair of channels/users in the system, i.e., when $\mathcal{I}(u_j) = \emptyset$ for all users u_j, we see that the optimization suggested by (7) becomes separable in the users, and hence across base stations. There is no need for collaboration among the base stations; at time kT each base station b_i picks the subset W_i of its users that maximizes $\sum_{j:u_j \in W_i} Q_j(kT)\,\mathbb{E}[C_j(kT)]$ and for the next T time slots schedules all the users in that subset.

An interesting case is when there is interference only within users associated to the same base station, and in a way that any two such users always interfere. In this situation, in addition to picking a subset W_i as earlier, each base station b_i in the second step uses the *Max-Weight/M-LWWF* scheduling rule [4] to schedule a user $u_j \in W_i$ that maximizes $Q_j(kT)C_j(kT)$. This is an earlier result presented in [15].

2) Fading-free symmetric system, singleton subsets: Consider the case when every channel state is constant over time and interferes with every other channel, i.e., $C_j(t) = \beta_j$ irrespective of t and $\mathcal{I}(u_j) = \mathcal{M} - \{u_j\}$. In addition, if the joint channel state distribution is assumed to be symmetric, i.e., $\pi(r_1, \ldots, r_M) = \pi(r_{\sigma(1)}, \ldots, r_{\sigma(M)})$ for any permutation σ on M letters, and the observable subsets for every base station b_i consist of all the singleton users, then each base station merely has to pick a user at every time slot to schedule or not. In this case, it can be shown that the policy $\mathcal{P}_T$ reduces to all the base stations together picking the user with the largest product of queue length and (constant) channel state in the system; in effect *picking the user with the longest queue is throughput-optimal*.

3) Single independent user per base station: Consider the case when every base station has just one user, with every user's channel state independent of every other user's channel state. This is essentially the model treated in [17], this time viewing each base station-user pair as an *uplink* user with the additional restriction that channel states are *iid* across time slots.

In this case, with some manipulation, it is not hard to show that the proposed throughput-optimal scheduling policy with $T = 1$ is exactly the *threshold* scheduling policy shown to be throughput-optimal in [17], i.e., if the optimum binary vector $z_j^*(t)$ for user j resulting from (7) has $z_j^*(t)(c_k) = 1$, then $z_j^*(t)(c_{k'}) = 1$ for all $c_{k'} > c_k$. In other words, each user u_j sets a threshold $\eta_j(t)$, and decides to contend to transmit (uplink) data if and only if its channel state at time t is at least $\eta_j(t)$.

VI. Packet delay performance

For every T, we saw that the scheduling policy $\mathcal{P}_T$ introduced in the previous section is throughput-optimal. The parameter T can be interpreted as a lag or delay in the base stations exchanging queue length information. It is natural to expect that with increasing lag T, the queueing delays seen by incoming arrivals grow. In this section we show that this is indeed the case by way of an analytical bound.

The following result states that the average queue lengths in the system grow at most *linearly* with the information lag T. By Little's Law, the average packet delays are linear in the average queue lengths for fixed arrival rates, and hence must also grow at most linearly in T. Though it is a direct consequence of the proof of Theorem 2, we state it as a separate theorem for its significance.

Theorem 3. *Let $\lambda \in \mathcal{R}$ be the average arrival rate vector for the system, and let $Q_j^{\mathcal{P}_T}(\tau)$ denote the queue length for user u_j at time slot τ when the arrival rates are λ and the scheduling policy is $\mathcal{P}_T$. Then, $\mathbb{E}[Q_j^{\mathcal{P}_T}(\tau)] = O(T)$.*

The proof follows from the proof of Theorem 2. It uses Lyapunov techniques similar to the ones in [11] to establish delay bounds for opportunistic scheduling. The reader is referred to Appendix B for details.

VII. Simulation results

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.

A. Simulation Setup

The network model we consider for the purpose of simulation is the one presented and discussed in Section IV-D. As per the throughput region of the system shown in Figure 6, consider the rate vector $\hat{\lambda} = (\frac{1}{16}, \frac{1}{4}, \frac{3}{16})$ which is the midpoint of the edge joining the corner points $(\frac{1}{8}, \frac{1}{8}, 0)$ and $(0, 0, \frac{3}{8})$, 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:

B. 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 Figure 7. We observe that the growth in average packet delay is linear with T which is in accordance with the result of Theorem 3, since by Little's law the average delay in the network is proportional to the average queue lengths for a fixed net arrival rate.

C. 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%.

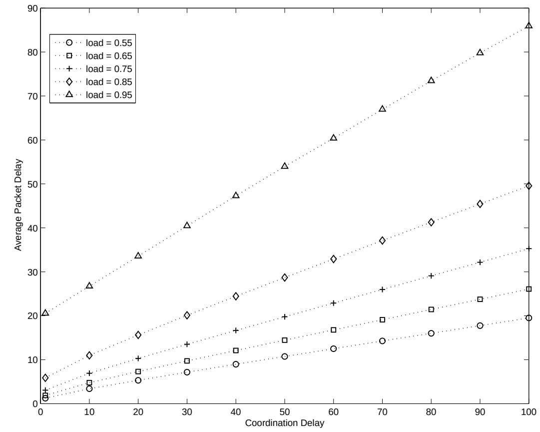

Fig. 7. Plot of average packet delay with lag T for various loads

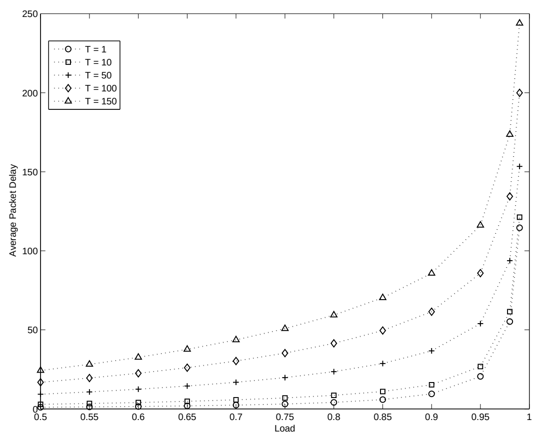

Fig. 8. Plot of average packet delay with load for various lags T

VIII. CONCLUSION

In this work, we studied multi-base-station wireless downlink scheduling with 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, loose coordination between the base stations in 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)* considering greedy, low-complexity scheduling strategies and evaluating their throughput, and *(ii)* refining the packet delay estimates in the system using large-deviations or heavy-traffic analysis.

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 $X_T(t)$ 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 $O(t) \stackrel{\triangle}{=} (O_{b_1}(t), \ldots, O_{b_N}(t))$ 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 $X_T(t)$, let $\phi_{W_1,\ldots,W_N;x}(t) \stackrel{\triangle}{=} \mathbb{P}[O_{b_1}(t) = W_1, \ldots, O_{b_N}(t) = W_N, X_T(t) = x]$.
3) Recall that the scheduling decision $B_j(t)$ for user $u_j \in \mathcal{U}(b_i)$ is a function of system state $X_T(t)$, the subset $O_{b_i}(t)$ chosen by its server (since users outside this subset are not scheduled), and current channel state $C_j(t)$. To indicate this, we explicitly write $B_j(t) = f_j^t(X_T(t), O_{b_i}(t), C_j(t))$, where for every $j = 1, \ldots, M$ and $t = 1, 2, \ldots$, f_j^t is a function that maps $(X_T(t), O_{b_i}(t), C_j(t))$ into $\{0, 1\}$, with $f_j^t(X_T(t), O_{b_i}(t), C_j(t)) = 0$ whenever $j \notin O_{b_i}(t)$.

Let $u_j \in \mathcal{U}(b_i)$. To begin, we note that

$$\begin{aligned}
\lambda_j &= \mathbb{E}[A_j(t)] = \mathbb{E}[E_j(t)] \leq \mathbb{E}[F_j(t)] \\
&= \mathbb{E}\left[C_j(t)B_j(t)\prod_{u_k \in \mathcal{I}(u_j)}(1 - B_k(t))\right] \\
&= \sum_{W_1,\ldots,W_N,x}\phi_{W_1,\ldots,W_N;x}(t)\mathbb{E}\Bigg[C_j(t)B_j(t) \times \\
&\qquad \prod_{u_k \in \mathcal{I}(u_j)}(1 - B_k(t))\Bigg| O(t) = (W_1, \ldots, W_N), \\
&\qquad\qquad X_T(t) = x\Bigg]\Bigg].
\end{aligned} \tag{8}$$

Evaluating the expectation gives

$$\begin{aligned}
&\mathbb{E}\left[C_j(t)B_j(t)\prod_{u_k\in\mathcal{I}(u_j)}(1-B_k(t))\middle| O(t)=(W_1,\ldots,W_N),\right.\\
&\qquad\left. X_T(t)=x\right]\\
&=\sum_{r_1,\ldots,r_M}\mathbb{P}[C(t)=(r_1,\ldots,r_M)|O(t)=(W_1,\ldots,W_N)]\times\\
&\quad\mathbb{E}\left[C_j(t)B_j(t)\prod_{u_k\in\mathcal{I}(u_j)}(1-B_k(t))\middle|\right.\\
&\quad\left. O(t)=(W_1,\ldots,W_N),X_T(t)=x,C(t)=(r_1,\ldots,r_M)\right]\\
&=\sum_{r_1,\ldots,r_M}\pi(r_1,\ldots,r_M)r_jf_j^t(x,W_i,r_j)\\
&\qquad\prod_{u_k\in\mathcal{I}(u_j)}(1-f_k^t(x,W_{\mathcal{B}(u_k)},r_k)),
\end{aligned}\tag{9}$$

since $C(t)$ is independent of $O(t)$. Using (9), (8) finally becomes

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

The fact that this equation has the same form as the one expressed by (5) and (6) for the long term rates of STS scheduling policies, together with the fact that

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

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

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

Lemma 1.

$$Q_j(t+T)\le\max\left\{Q_j(t)-\sum_{\tau=t}^{t+T-1}F_j(\tau),0\right\}+\sum_{\tau=t}^{t+T-1}A_j(\tau).\tag{10}$$

Proof: Consider two cases:

- $Q_j(t)\ge\sum_{\tau=t}^{t+T-1}F_j(\tau)$: In this case, according to (1), both sides of (10) are equal.
- $Q_j(t)<\sum_{\tau=t}^{t+T-1}F_j(\tau)$: 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 $Q_j(t+T)=Q_j(t)-\sum_{\tau=t}^{t+T-1}F_j(\tau)+\sum_{\tau=t}^{t+T-1}A_j(\tau)\le\sum_{\tau=t}^{t+T-1}A_j(\tau)$ and we are done). We must then have

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

which finishes the proof.

∎

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

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

In what follows, we will 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

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

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

$$\mathbb{E}[\Delta L(X_T(kT))|Q(kT) = q] \leq MT^2\left(C_{\max}^2 + A_{\max}^2\right) - \tag{11}$$

$$2\sum_{j=1}^{M} q_j \mathbb{E}\left[\sum_{\tau=KT}^{kT+T-1}[F_j(\tau) - A_j(\tau)]\middle| Q(kT) = q\right]$$

$$= MT^2\left(C_{\max}^2 + A_{\max}^2\right) + 2T\sum_{j=1}^{M} q_j\lambda_j$$

$$- 2\sum_{j=1}^{M} q_j \mathbb{E}\left[\sum_{\tau=KT}^{kT+T-1} F_j(\tau)\middle| Q(kT) = q\right]$$

$$= MT^2\left(C_{\max}^2 + A_{\max}^2\right) + 2T\sum_{j=1}^{M} q_j\lambda_j$$

$$- 2\sum_{j=1}^{M} q_j T\mathbb{E}\left[F_j(kT)| Q(kT) = q\right], \tag{12}$$

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 $z_1^*, \dots, z_M^*$ computed at time slot kT, and are thus statistically identical from time kT upto $kT + T - 1$.

By hypothesis, $\lambda = (\lambda_1, \dots, \lambda_M) \in \mathcal{R}$. Hence there must exist a small $\epsilon > 0$ 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, $i = 1, \dots, n$, where $\mathcal{P}_i = (W_1^i, \dots, W_N^i, z_1^i, \dots, z_M^i)$ (the superscript i indexes the SSS policy). We have, for $1 \leq j \leq M$,

$$\mu_j^{\mathcal{P}_{TS}} = \sum_{i=1}^{n}\phi_i \sum_{r_1,\dots,r_M}\left[\pi(r_1, \dots, r_M)\cdot r_j \cdot z_j^i(r_j) \times \prod_{u_n \in \mathcal{I}(u_j)}(1 - z_n^i(r_n))\right]. \tag{13}$$

We add and subtract $2T\sum_{j=1}^{M} q_j\mu_j^{\mathcal{P}_{TS}}$ to the right hand side of (12) to get

$$\mathbb{E}[\Delta L(X_T(kT))|Q(kT) = q] \leq MT^2\left(C_{\max}^2 + A_{\max}^2\right) + 2T\sum_{j=1}^{M} q_j(\lambda_j - \mu_j^{\mathcal{P}_{TS}}) + 2T\left(\sum_{j=1}^{M} q_j\mu_j^{\mathcal{P}_{TS}} - \sum_{j=1}^{M} q_j\mathbb{E}\left[F_j(kT)| Q(kT) = q\right]\right). \tag{14}$$

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 (13), we can write

$$\begin{aligned}
&\left(\sum_{j=1}^{M} q_j\mu_j^{\mathcal{P}_{TS}} - \sum_{j=1}^{M} q_j\mathbb{E}\left[F_j(kT)| Q(kT) = q\right]\right) \\
&= \left(\sum_{j=1}^{M} q_j \sum_{i=1}^{n}\phi_i \sum_{r_1,\dots,r_M}\left[\pi(r_1, \dots, r_M)\cdot r_j \cdot z_j^i(r_j) \times \prod_{u_k \in \mathcal{I}(u_j)}(1 - z_k^i(r_k))\right] - \sum_{j=1}^{M} q_j\mathbb{E}\left[F_j(kT)| Q(kT) = q\right]\right) \\
&\leq \left(\max_{i=1,\dots,n}\left(\sum_{j=1}^{M} q_j \sum_{r_1,\dots,r_M}\left[\pi(r_1, \dots, r_M)\cdot r_j \cdot z_j^i(r_j) \times \prod_{u_k \in \mathcal{I}(u_j)}(1 - z_k^i(r_k))\right]\right) - \sum_{j=1}^{M} q_j\mathbb{E}\left[F_j(kT)| Q(kT) = q\right]\right).
\end{aligned} \tag{15}$$

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

$$\begin{aligned}
&\sum_{j=1}^{M} q_j\mathbb{E}\left[F_j(kT)| Q(kT) = q\right] \\
&= \sum_{j=1}^{M} q_j \sum_{r_1,\dots,r_M}\pi(r_1, \dots, r_M) r_j z_j^*(r_j)\prod_{u_k \in \mathcal{I}(u_j)}(1 - z_k^*(r_k)) \\
&\geq \sum_{j=1}^{M} q_j \sum_{r_1,\dots,r_M}\pi(r_1, \dots, r_M) r_j z_j^i(r_j)\prod_{u_k \in \mathcal{I}(u_j)}(1 - z_k^i(r_k))
\end{aligned}$$

for $i = 1, \dots, n$, by the optimal choice of the z_k^*. Together with (15), this proves the lemma. ■

Using Lemma 2, (14) implies

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

Taking expectations over both sides of (16) and summing from $k = 1, \dots, K$ leads to

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

Rearranging terms and noting that $L(Q((K+1)T)) \geq 0$ yields

$$\frac{1}{K}\sum_{k=1}^{K}\sum_{j=1}^{M}\mathbb{E}[Q_j(kT)] \leq \frac{L(X_T(T))}{2\epsilon TK} + \frac{TM(A_{\max}^2 + C_{\max}^2)}{2\epsilon}$$

$$\Rightarrow \quad \limsup_{K\to\infty}\frac{1}{K}\sum_{k=1}^{K}\sum_{j=1}^{M}\mathbb{E}[Q_j(kT)] \leq \frac{TM(A_{\max}^2 + C_{\max}^2)}{2\epsilon} \tag{17}$$

$$< \infty.$$

Together with the fact that arrivals and channel states are bounded, this means that the (irreducible and aperiodic) system state process $X_T(t), t = 1, 2\ldots$ is positive recurrent. This completes the proof. ■

References

[1] Motorola, "Long term evolution (LTE): Overview of LTE air-interface," *White paper*, 2007.

[2] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," *IEEE Transactions on Automatic Control*, vol. 4, pp. 1936–1948, December 1992.

[3] L. Tassiulas and A. Ephremides, "Dynamic server allocation to parallel queues with randomly varying connectivity," *IEEE Transactions on Information Theory*, vol. 39, pp. 466–478, March 1993.

[4] M. Andrews, K. Kumaran, K. Ramanan, A. L. Stolyar, R. Vijayakumar, and P. Whiting, "Scheduling in a queueing system with asynchronously varying service rates," *Probability in Engineering and Informational Sciences*, vol. 14, pp. 191–217, 2004.

[5] S. Shakkottai and A. Stolyar, "Scheduling for multiple flows sharing a time-varying channel: The exponential rule," *American Mathematical Society Translations, Series 2, a volume in memory of F. Karpelevich, Yu. M. Suhov, Editor*, vol. 207, 2002.

[6] A. Eryilmaz, R. Srikant, and J. R. Perkins, "Stable scheduling policies for fading wireless channels," *IEEE/ACM Transactions on Networking*, vol. 13, pp. 411–424, 2005.

[7] X. Lin and S. B. Rasool, "Constant-time distributed scheduling policies for ad hoc wireless networks," in *in Proceedings of IEEE Conference on Decision and Control*, 2006.

[8] X. Wu and R. Srikant, "Scheduling efficiency of distributed greedy scheduling algorithms in wireless networks," in *INFOCOM*, 2006.

[9] A. Eryilmaz, A. Ozdaglar, and E. Modiano, "Polynomial complexity algorithms for full utilization of multi-hop wireless networks," in *Proceedings of IEEE Infocom*, (Anchorage, AK), 2007.

[10] S. Sanghavi, L. Bui, and R. Srikant, "Distributed link scheduling with constant overhead," in *SIGMETRICS '07: Proceedings of the 2007 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, pp. 313–324, 2007.

[11] M. J. Neely, *Dynamic Power Allocation and Routing for Satellite and Wireless Networks with Time Varying Channels*. PhD thesis, Massachusetts Institute of Technology, LIDS, November 2004.

[12] 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.

[13] K. Kar, X. Luo, and S. Sarkar, "Throughput-optimal scheduling in multichannel access point networks under infrequent channel measurements," *IEEE Transactions on Wireless Communications*, vol. 7, no. 7, pp. 2619–2629, 2008.

[14] N. Chang and M. Liu, "Optimal channel probing and transmission scheduling for opportunistic spectrum access," in *Proc. ACM International Conference on Mobile Computing and Networking (MobiCom)*, (Montreal, Canada), September 2007.

[15] A. Gopalan, C. Caramanis, and S. Shakkottai, "On wireless scheduling with partial channel state information," in *Proc. Allerton Conference on Communication, Control and Computing*, 2007.

[16] A. Pantelidou, A. Ephremides, and A. Tits, "Joint scheduling and routing for ad-hoc networks under channel state uncertainty," in *Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks and Workshops, 2007. WiOpt 2007. 5th International Symposium on*, pp. 1–8, April 2007.

[17] L. Ying and S. Shakkottai, "On throughput optimality with delayed network-state information," in *Proc. of the Information Theory and Applications Workshop*, pp. 339–344, January 2008.

[18] L. Ying and S. Shakkottai, "Scheduling in mobile wireless networks with topology and channel-state uncertainty," in *Proc. IEEE Infocom*, 2009.

[19] S. Das, H. Viswanathan, and G. Rittenhouse, "Dynamic load balancing through coordinated scheduling in packet data systems," in *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies. IEEE*, vol. 1, pp. 786–796 vol.1, March-3 April 2003.

[20] A. Stolyar and H. Viswanathan, "Self-organizing dynamic fractional frequency reuse for best-effort traffic through distributed inter-cell coordination," in *INFOCOM 2009. The 28th Conference on Computer Communications. IEEE*, pp. 1287–1295, April 2009.

[21] 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, 2008.

[22] B. Rengarajan and G. de Veciana, "Architecture and abstractions for environment and traffic aware system-level coordination of wireless networks: The downlink case," in *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, pp. 502–510, April 2008.