Noname manuscript No.
(will be inserted by the editor)

On Distributed Scheduling with Heterogeneously Delayed Network-State Information

Akula Aneesh Reddy · Siddhartha Banerjee · Aditya Gopalan · Sanjay Shakkottai · Lei Ying

Received: date / Accepted: date

Abstract We study the problem of distributed scheduling in wireless networks, where each node makes individual scheduling decisions based on heterogeneously delayed network state information (NSI). This leads to inconsistency in the views of the network across nodes, which, coupled with interference, makes it challenging to schedule for high throughputs.

We characterize the network throughput region for this setup, and develop optimal scheduling policies to achieve the same. Our scheduling policies have a threshold-based structure and, moreover, require the nodes to use only the "smallest critical subset" of the available delayed NSI to make decisions. In addition, using Markov chain mixing techniques, we quantify the impact of delayed NSI on the throughput region. This not only highlights the value of extra NSI for scheduling, but also characterizes the loss in throughput incurred by lower complexity scheduling policies which use homogeneously delayed NSI.

Keywords Wireless Networks · Scheduling Algorithms · Delayed Information

1 Introduction

Modern data networks are increasingly being supported on the wireless medium. In this regard, there are two primary trends which emerge. Firstly, there is an

An earlier version of this work has appeared in the Proceedings of the 48th Annual Allerton Conference on Communication, Control, and Computing, 2010 [1].

A. A. Reddy, S. Banerjee, A. Gopalan, S. Shakkottai
The University of Texas at Austin
Austin, Texas 78712
E-mail: aneesh, sbanerjee, gadit, shakkott@mail.utexas.edu

L. Ying
Iowa State University
Ames, Iowa 50011
E-mail: leiying@iastate.edu

ever increasing demand for higher data rates, which is caused both due to increasing number of users on such networks as well as a trend towards applications that are more data-intensive. Over the last few years, we have moved from cellular networks dedicated to voice traffic to WiFi networks supporting internet traffic over a small geographic area to the bandwidth demands posed by a plethora of applications on modern 'smartphones', and this trend shows no signs of abating. Due to the nature of the wireless medium, the resources available to support this extra traffic are limited, and this puts added importance on the need for optimizing the protocols that are used for scheduling and routing the information. A more subtle trend in wireless communications is a move towards decentralization. The old paradigms of cellular networks with a centralized controller are increasingly giving way to more distributed network architectures like those seen in wireless sensor networks (WSNs), wireless mesh networks (WMNs) and mobile ad-hoc networks (MANETs). We thus require network algorithms that are not only capable of supporting high data-rates, but also do so in a distributed manner.

Managing data in wireless networks, as opposed to traditional wireline networks, is complicated by two effects unique to the wireless medium – channel fading and interference. Channel fading, at a high level, refers to the fact that the wireless channel between two users is not constant (like in corresponding wireline systems), but fluctuates in time; knowledge of these fluctuations, by means of channel sensing, allows an algorithm to schedule transmissions in an opportunistic manner (i.e., transmit more when the channel quality is good, and remain silent when not). Due to the shared nature of the medium, the successful reception of a user's transmissions, even when the channel quality is high, depends on its interactions with transmissions from other users. This phenomenon is known as interference, and naturally necessitates a centralized scheduling approach in order to coordinate transmissions to/from various users.

With this background in mind, the fundamental wireless scheduling problem can be viewed as one of scheduling transmissions in the network in the presence of fading and interference in order to support data flows with as high rates as possible. This problem is tackled by a long line of work, started by Tassiulas and Ephremides [2] and extensively followed up by others [3–9], resulting primarily in the celebrated Back-Pressure network scheduling algorithm [2]. This algorithm schedules network links to maximize throughput in an opportunistic fashion using instantaneous network state information (NSI), i.e., queue and channel state knowledge across the entire network.

Though Back-Pressure scheduling guarantees the best possible throughput performance for flows in networks, it suffers from two drawbacks – (a) the algorithm requires solving a global optimization problem at each time to determine the schedule, making it highly centralized, and (b) it requires knowledge of instantaneous NSI from the whole network, i.e. feedback about time-varying channel and queue states from all links of the network. Towards addressing these issues, researchers have developed distributed implementations of the Max-weight algorithm [10–16], which use local NSI to achieve optimal/nearoptimal throughput performance. Additionally, there have been studies on scheduling in the presence of partial, noisy or delayed channel state information (CSI). These include scheduling with limited channel sensing capabilities and channel-probing costs [17–19], and scheduling with limited/uncertain channel-state feedback [20–26].

An important problem which arises while scheduling in the presence of channel fading and network interference, and which remains unexplored by these works, is the fact that there is often a widespread *mismatch* in the information that nodes possess. Each node has complete information about its own queue and channel state, but has progressively "coarser" information about other nodes' NSI as the distance to these node increases. This happens because: *(i)* prohibitive overheads in measuring and communicating NSI, *(ii)*fading occurring faster than communicating NSI, leading to delayed channel-state information and/or *(iii)* propagation delays due to geographic separation of nodes. In this regard, the work of Ying and Shakkottai [27,28] investigates distributed scheduling with *delayed* network state information, i.e., with delayed topology [28] and delayed wireless channel state information [27]. In particular, the latter paper considers networks with symmetric delays in channel state and queue information, i.e., every node has instantaneous CSI for itself, and CSI from other nodes delayed by a globally fixed number of time slots. In this setting, all the nodes share a common view of the network – i.e., the network state with a fixed, uniform delay – which the nodes can use to implement threshold-type scheduling based on individual instantaneous CSI and achieve throughput-optimality.

The assumption of symmetric delayed state information is often not satisfied in general networks which could have *heterogeneous* delays in channel state information. For instance, two nodes in a network could possess channel state information from a third node delayed by different amounts. This can easily result in widely differing estimates at the first two nodes for the third node's network/channel state. A challenging problem thus is how to use the *heterogeneously* delayed NSI to schedule. Unlike the case of homogeneous delayed CSI with additional individual CSI [27], the scheduling algorithm now needs to account for the fact that the nodes can possess *inconsistent* network state information – *each node can potentially have a completely different view of the network state*. It is a priori unclear how distributed scheduling can be performed when nodes have such *inconsistent* (i.e., heterogeneously delayed) channel information. This paper aims to both (a) characterize the throughput region with inconsistent NSI, and (b) develop scheduling algorithms that use a minimal amount of heterogeneously delayed network state information and are yet throughput-optimal. Having done this, it also examines the "value" or "cost" of network state information, in regard to throughput, by quantitatively estimating throughput improvement/degradation when the nodes have "finer/coarser" delayed NSI structures respectively.

1.1 Our Contributions

In this work, we consider the problem of distributed wireless scheduling in the presence of arbitrary interference set constraints and Markovian channel fading, where each transmitter knows the other transmitters' NSI with arbitrary, *heterogeneous* delays. This disparity in the delays of NSI available to the transmitters can potentially result in inconsistent views of the global current network state, causing conflicting/poor local scheduling decisions among the transmitters. Given such a NSI structure, how can all the transmitters in the network use their possibly inconsistent individual information to make scheduling decisions for good overall throughout? Our main contributions in this regard are as follows:

1. We characterize the network throughput region when each transmitter possesses instantaneous local NSI (i.e., NSI from itself) and *heterogeneously delayed* NSI from other transmitters. For this purpose, we introduce a special, restricted class of *static-split* scheduling policies, in which each transmitter uses only *critical* delayed CSI from other nodes, along with its own channel state information, to make transmission decisions. An important observation here is that these static-split scheduling rules, in the conventional sense, are *not necessarily* throughput-optimal − deterministic scheduling at all nodes still achieves corner points of the rate region, but time sharing across the corner points is no longer possible with each node using only local information. Rather, the throughput region results by time sharing using *global, common* randomness together with static-split strategies.
2. We develop a decentralized, threshold-based throughput-optimal scheduling algorithm for the network, in which nodes use only critical NSI to schedule. In every time slot, each node uses (delayed) network queue length information along with critical delayed CSI from other nodes to compute a suitable *local threshold*, and decides to schedule transmission by comparing the threshold with its own channel state. Further, we show that delayed queue length and channel state information, when used at each node to dynamically pick local threshold-scheduling rules, acts as a source of global, common randomness for all the transmitters, helping to achieve stability across the entire throughput region.
3. With respect to the canonical heterogeneous NSI setting, we quantify the loss (gain) in throughput that results from all transmitters having the maximum (minimum) possible *homogeneously* delayed NSI from other transmitters. This quantifies the value of delayed NSI in terms of its impact on the system throughput region, and is accomplished using techniques from mixing of Markov chains.

2 Scheduling with Heterogeneously Delayed NSI: An Example

Let us consider an illustrative example to help understand the essential difficulties and challenges in scheduling when the NSI available to each user is delayed in a heterogeneous fashion. Suppose we have three wireless users A, B and C, attempting to transmit packet data to a common receiver in a time-slotted manner. We assume that the users are located sufficiently close to each other so as to make their transmissions interfere, i.e., if the number of users attempting to transmit in a time slot is more than one, no packets reach the receiver. The channel between each user and the receiver is time-varying, and in the event of a successful transmission, the channel state or rate of the lone attempting user specifies how many packets can be sent to the receiver in that time slot.

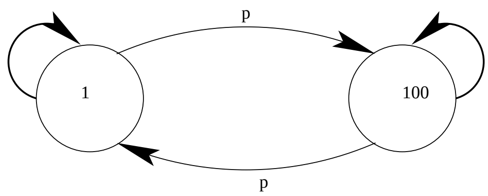

Fig. 1 User C channel Markov chain

Each user possesses instantaneous channel (and queue backlog) state information about its own channel and receives delayed channel (and queue backlog) state information from other users for the purpose of making transmit/no-transmit decisions. Let us assume for simplicity that the channels for users A and B take rates 1 or 100 (packets per time slot) each with probability $\frac{1}{2}$ *independently* in each time slot; however user C's channel state evolves as a *Markov chain* between rates 1 and 100 with crossover probability $p = \frac{1}{4}$ (Fig. 1). User A gets channel state information from users B and C delayed by 1 time slot, user B gets channel state information from users A and C delayed by 1 and 2 time slots respectively, and user C gets channel state information from users A and B delayed by 1 time slot. Fig. 2 depicts this NSI structure at time t – a circle in the row of Tx A at time $t-1$ indicates that it is the latest information B has about A's channel state, and so on.

Note that due to this information structure, at each time users A and B have different "views" of user C's current channel state owing to disparate channel state information delays. For instance, if user C's channel two time slots ago was at rate 100 and one time slot ago was at rate 1, user A is led to believe that user C's current channel is very likely to have rate 1, whereas user B's belief would be that user C's channel is most probably at rate 100. In such events, how must the users act so that they can avoid excessive collision and achieve desired data transmission rates? It turns out, as we show later on, that

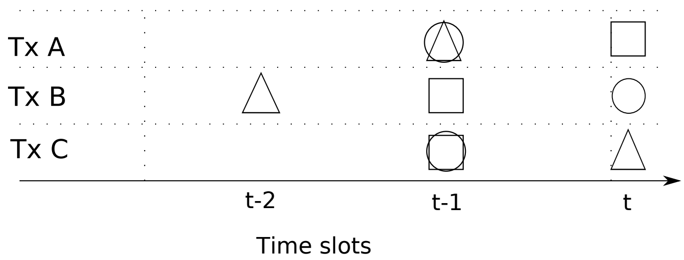

Fig. 2 Heterogeneous NSI for the 3-user network: Squares, circles and triangles represent the *most recent* channel state information available to user A, B and C respectively.

the following *threshold*-based transmission rule, for each user, is a throughput *optimal* scheduling strategy: In every time slot,

1. All the three users compute individual "threshold values" (to be used later) as functions of their respective delayed queue length information and certain "critical" subsets of their available delayed channel state information – user A works out a threshold value as a function of the one-step delayed channel states of user B and user C, and so on.
2. Each user looks at the value(s) of its critical set of delayed NSI, compares the corresponding threshold value and its own current channel state, and *attempts transmission only if its current channel state exceeds the threshold*.

Now, consider the case when *both* user A and user B have user C's channel state information with a delay of 2 time slots. Compared to the earlier set of delays, user A has one step "coarser" channel state information about user C, so we expect a degradation in the overall set of achievable data rates that all the users can support. In fact, it can be shown that

1. *The best average sum rate achievable in the latter system is* 56.69 *packets/time slot*, whereas
2. *The best average sum rate achievable in the former system is* 62.88 *packets/time slot* – an increase of about 11% in the sum rate with one additional step of channel state information.

In this work, we provide a theory for wireless scheduling with heterogeneously delayed channel state information that answers the following useful questions:

1. What are all the long-term average rates (i.e., the throughput region) that such a wireless system with an arbitrary delayed NSI structure can support?
2. How can each user make scheduling (transmission) decisions – just based on its limited amount of delayed information about other users' channel states – to be able to support *any* given feasible data rate? Moreover, which are the time slots whose channel state information is "crucial" or "essential" for making throughput-optimal decisions?
3. By how much does the throughput region of the system change with better or worse delayed channel state information?

3 System Model

This section is concerned with setting up the system model we use to develop our results. This includes describing the network model, traffic model and the structure of interference between wireless users. A key component of the model is the information structure of delayed network state information available to each user to schedule transmissions, which is described here. We conclude by defining the performance metric of throughput that we consider in this paper.

– **Network Model:** We consider a wireless network consisting of L transmitter-receiver pairs denoted by $\mathcal{L}$. We model the (time-varying) capacity of each link l using a discrete-time Markov chain, denoted by $\{C_l[t]\}$, on the finite state space $\mathcal{C} = \{c_1, c_2, c_3, ..., c_M\}$, where $c_1 \leq \cdots \leq c_M$ are nonnegative integers. Furthermore, we require that the link's capacity is independent and identically distributed, with transition probabilities $P_{ij} := \Pr[C_l[t+1] = c_j | C_l[t] = c_i]$ for the respective Markov chain. The above channel model is assumed for notational simplicity, and our results hold even for the case of networks where each link can be modeled by a separate Markov chain (different state space and different transition probabilities). The only condition for our results to hold is that channels are independent across various transmitter-receiver pairs (users).
  We assume that the channel state Markov chain parameterized by the transition probabilities $\{P_{ij}\}_{i,j}$ is irreducible and aperiodic [1]. Thus the channel state process has a stationary distribution and we denote the stationary probability of being in a state $c_j, j \in \{1, 2, 3..., M\}$ by π_j.

  Finally, each link l has an associated queue of length $Q_l[t]$, which holds data packets to be transmitted across the link.

– **Interference Model:**
  We model radio interference in the network using a *packet capture* model. Specifically, for each link l, let I_l denote the set of links in the network that interfere with l. Note that I_l can be an arbitrary but fixed set of interfering links for link l, which can be used to model geographically close transmitters, transmitters using the same shared time/frequency resource etc. We say that a *collision* occurs with a transmission scheduled on link l if, in the same time slot, a transmission is scheduled on a link $l' \in I_l$. When there is no collision at link l in time slot t, then $\min(C_l[t], Q_l[t])$ packets are successfully received across the link. However, when a collision occurs on link l, we assume that $\min(\gamma_l C_l[t], Q_l[t])$ packets are received successfully across the link. For each l, we assume there exists $\gamma_l \in [0, 1]$ such that $\{\gamma_l c_1, \ldots, \gamma_l c_M\}$ are all integers (i.e., at each time t, $\gamma_l C_l[t]$ is an integer). In general, it suffices to have all the $\gamma_l c_i$ be rational numbers, for

[1] This assumption is to ensure that the system state Markov chain (defined in Section 3.2) is irreducible and aperiodic, by suitably augmenting the state space.

then the notion of a packet (equivalently, the queue length) can be suitably redefined to satisfy this assumption.

We can consider an alternative model where if a collision occurs on link l, then $C_l[t]$ packets are successfully received *with probability* γ_l, else no packets are received. In this case, $\gamma_l C_l[t]$ need not be integer since in any event, an integer number of packets (0 or $C_l[t]$) is successfully received. Setting $\gamma_l = 0$ for all l corresponds to a "perfect collision" interference model, where no packets get through in the event of simultaneous transmissions, whereas $\gamma_l > 0$ models reception of packets in a probabilistic manner. Though the results in this paper are proved for the former, deterministic interference model, all of them can be shown to hold for the latter, probabilistic interference model as well.

– **Traffic Model:** We assume single-hop flows in the network, and that each node does not have multiple simultaneous connections. Each link in the network has a traffic process denoted by $A_l[t]$, that describes the number of packets that arrive at sender node of link at time t. For every link l, we assume that $A_l[t]$ is an integer-valued process independent across time slots t, with $0 \leq A_l[t] \leq A_{\max} < \infty$ almost surely, and set $\lambda_l := E[A_l[t]] < \infty$. We further assume that $\Pr[A_l[t] = 0] > 0$ and $\Pr[A_l[t] = 1] > 0$.[2]

3.1 NSI Structure and Scheduling Policies

We assume that each transmitter accesses network state information parameterized in terms of its information delays from other transmitters. Specifically, at time t, transmitter l has channel and queue state information history of link l upto and including time t, but has only *delayed* channel state information and queuing history of other links in the network. Let $\tau_l(h)$ denote the delay incurred in communicating the channel and queue state information of link h to the transmitter node of link l. Thus, each transmitter node l has a vector of delay values $\boldsymbol{\tau_l}$ that characterizes the available delayed NSI at l. We denote by τ_{min} and τ_{max} the minimum and maximum channel (and queue) state information delay across the network, i.e.,

$$\tau_{min} = \min_{l,h \in \mathcal{L}: l \neq h} \tau_l(h); \tau_{max} = \max_{l,h \in \mathcal{L}: l \neq h} \tau_l(h).$$

We denote the set $\{C_l[t-\tau], C_l[t-\tau+1], ..., C_l[t]\}$ by $C_l[t](0:\tau)$ and the set $\{C_l[t]\}_{l \in \mathcal{L}}$ by $\mathbf{C}[t]$. We denote the information available at transmitter l by $\{\mathcal{P}_l(\mathbf{C}[t](0:\tau_{max})), \mathcal{P}_l(\mathbf{Q}[t](0:\tau_{max}))\}$, where

$$\mathcal{P}_l(\mathbf{C}[t](0:\tau_{max})) := \{\boldsymbol{\mathcal{P}}_{lm}(\mathbf{C}[t](0:\tau_{max}))\}_{m \in \mathcal{L}}, \text{ with}$$
$$\boldsymbol{\mathcal{P}}_{lm}(\mathbf{C}[t](0:\tau_{max})) := \{C_m[t-\tau]\}_{\tau=\tau_{max}}^{\tau_l(m)},$$

and likewise for $\mathcal{P}_l(\mathbf{Q}[t](0:\tau_{max}))$. A *scheduling policy* is a map for each link l that maps its network state information $\{\mathcal{P}_l(\mathbf{C}[t](0:\tau_{max})), \mathcal{P}_l(\mathbf{Q}[t](0:\tau_{max}))\}$ to a transmit/no-transmit scheduling decision.

[2] These assumptions are to ensure that the system state Markov chain (defined in Section 3.2) is irreducible and aperiodic, by suitably augmenting the state space.

3.2 Performance Objective: Throughput/Stability

We define the *state* of the network at time t as the process $\mathbf{Y}[t] = \{Q_l[t](0 : \tau_{max}), C_l[t](0 : \tau_{max})\}_{l\in\mathcal{L}}$, and specifically denote this state process under a scheduling policy $\mathcal{F}$ by $\mathbf{Y}^{\mathcal{F}}[t]$.

Given the arrival rate vector $\{\lambda_l\}_{l\in\mathcal{L}}$ and a scheduling policy $\mathcal{F}$, we say that the network is *stochastically stable* if the system state Markov chain $\mathbf{Y}^{\mathcal{F}}[t]$ is positive recurrent. We say that an arrival rate vector $\{\lambda_l\}_{l\in\mathcal{L}}$ is *supportable* if there exists a scheduling policy that makes the network stochastically stable.

4 Distributed Scheduling with Heterogeneously Delayed NSI

In this section, we first characterize the *throughput region* of the wireless system, i.e., the set of all supportable arrival rates. Traditionally the throughput region is the set of arrival rates that can be supported by *Static Service Split (SSS)* scheduling rules – a restricted class of queue-length oblivious and channel-state aware strategies [3,5,27,20]. Although we use a similar approach, a crucial distinction arises when considering static-split scheduling in our setting. In the classical framework of static-split rules, deterministic scheduling rules achieve the *corner points* of the throughput region, and time sharing using *randomized* static-split rules then attains the entire region. However, in our decentralized setting, though deterministic scheduling using local information at each node still achieves all the corner points of the rate region, time sharing among these corner points is *not* possible using only *local* information at each node. Instead, *global, common randomness* is required for time-sharing and for achieving the entire throughput region. Thus, *static-split scheduling, in the conventional sense, is not necessarily throughput-optimal for our setting.*

Consider a simple example of two nodes sharing a unit-rate collision channel – in each time slot, one packet can be transmitted by each node, but a collision occurs if both nodes simultaneously transmit. As shown in Figure 3, the rate point $(1, 0)$ (resp. $(0, 1)$) can be achieved by deterministic scheduling at the nodes, i.e., if node 1 (resp. node 2) always transmits and node 2 (resp. node 1) always stays silent. The dotted line denotes rate pairs $(\alpha, 1-\alpha)$ achieved by time sharing between the corner points $(1, 0)$ and $(1, 0)$. This is possible when the nodes use global, common randomness (e.g., a common sequence of coin tosses with the probability of heads being α), and captures the traditional notion of randomized static split rules.

On the other hand, when the nodes can only use local information (e.g., individual, independent coin flips), it is not hard to see that points beyond the curved line in Figure 3 cannot be achieved. Indeed, a point on this curved line results when each node $i \in \{1, 2\}$ transmits independently with probability p_i, which represents static split scheduling carried out individually at each node.

Note that SSS rules need not necessarily preclude joint decisions via common randomness. Yet, the point of the above example is to emphasize the fact that common randomness is, in a sense, indispensable when performing distributed scheduling. In other words, one cannot hope to achieve the entire throughput region by applying static split rules using only local coin flips at each node; rather, the SSS rules need to be able to access global, common randomness.

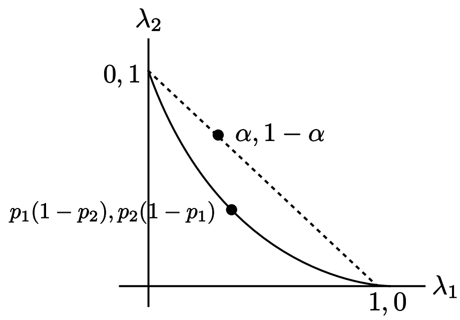

Fig. 3 Static-split scheduling rules, in our setting, are not necessarily throughput-optimal in the conventional sense. For two transmitters sharing a unit-rate collision channel, the corner points $(1, 0)$ and $(0, 1)$ are achieved by individual static-split scheduling at each transmitter. However, with just local information and randomization, rates beyond the curved boundary cannot be achieved. To time share between the corner points and attain rates on the dotted line requires *global common* randomness.

Given this distinguishing feature of static-split scheduling in our setting, we show in Section 4.1 that by combining appropriate "static" scheduling at each transmitter with the use of global common randomness, we show that all points in the throughput region can be achieved. Next, in Section 4.2, we further simplify the structure of the static scheduling policies, by identifying the "critical set" of available delayed NSI sufficient for each transmitter to achieve any valid rate point.

Finally, in Section 4.3, we give a throughput-optimal, distributed scheduling algorithm for all transmitters, that uses critical delayed queue and channel states as a source of global common randomness along with scheduling with suitable static rules at the transmitters. In this regard, the idea leveraged from the above example is the following: if both nodes can access delayed queue length information, say $(Q_1(t-10), Q_2(t-10))$, at every time slot t, it is possible to time share between the two corner points. This can be done, for instance, when node 1 transmits whenever $Q_1(t-10) \geq Q_2(t-10)$ and node 2 transmits whenever $Q_1(t-10) < Q_2(t-10)$. The key advantage of a queue-based policy, as opposed to a fixed common random coin, is that the joint distribution of the queues automatically adapts, and the resulting algorithm achieves any point in the interior of the throughput region. Thus, this is in the spirit of traditional Back-Pressure algorithms, but in the context of deriving the "correct" common randomness.

4.1 Throughput Characterization

Towards describing the throughput region, i.e., the set of all supportable arrival rate vectors $\{\lambda_l\}_{l\in\mathcal{L}}$, consider a collection of functions $\{f_l\}$, one for each link/transmitter $l \in \mathcal{L}$, where each $f_l : \mathcal{P}_l(\mathbf{C}[t](0 : \tau_{max})) \to \{0,1\}$. These maps $\{f_l\}_{l\in\mathcal{L}}$ parameterize a *static-split* or *stationary* scheduling policy – oblivious of queue state information, and of channel state information past τ_{max} – as follows: at each time t, every link l computes the binary value $f_l(\mathcal{P}_l(\mathbf{C}[t](0 : \tau_{max})))$ and attempts to transmit (i.e., schedule itself) whenever this binary value is 1.

If the delayed channel state information at time t is $\mathbf{C}[t-\tau_{max}] = \mathbf{c}$, then the expected rates at time t that all links receive when each transmitter l applies the static-split scheduling policy f_l is defined to be $\mathbf{S}(\mathbf{c},\mathbf{f}) = \{S_l(\mathbf{c},\mathbf{f})\}_{l\in\mathcal{L}}$, as follows:

$$S_l(\mathbf{c},\mathbf{f}) = E\Big[C_l[t]f_l(\mathcal{P}_l(.))\big(\gamma_l + (1-\gamma_l)\prod_{m\in I_l}(1-f_m(\mathcal{P}_m(.)))\big) \mid \mathbf{C}[t-\tau_{max}] = \mathbf{c}\Big],$$

where $\mathcal{P}_l(.) = \mathcal{P}_l(\mathbf{C}[t](0 : \tau_{max}))$. We now define $\eta(\mathbf{c})$ as follows,

$$\eta(\mathbf{c}) = \mathcal{CH}_f\big(S(\mathbf{c},\mathbf{f})\big).$$

Thus, $\eta(\mathbf{c}) \subset \mathbf{R}^L$ is the convex hull of all the possible expected transmission rates that achieved by static-split scheduling policies in time slot t, when the common NSI up to time $t-\tau_{max}$ is c. Finally, our candidate for the throughput region of the system is the region $\Lambda \in \mathbf{R}^L$ defined by

$$\Lambda = \big\{\lambda : \lambda = \sum_{\mathbf{c}\in\mathcal{C}^L}\pi(\mathbf{c})x(\mathbf{c}), x(\mathbf{c}) \in \eta(\mathbf{c})\ \forall \mathbf{c}\in\mathcal{C}^L\big\}.$$

In other words, Λ is the Minkowski sum of the sets $\{\eta(\mathbf{c})\}_{\mathbf{c}\in\mathcal{C}^L}$ weighted by the respective probabilities $\pi(\mathbf{c})$. The corner points of Λ correspond directly to static-split scheduling rules, and in general, each point in Λ represents the expected rates delivered to all links obtained by time sharing across static-split scheduling rules. Note that this time sharing across nodes' scheduling decisions, as described in the example above, can be achieved with *global, common* randomization, e.g., a common sequence of coin flips available to all the nodes. Thus, Λ is an inner bound for the throughput region of the system. However, the following result establishes that the throughput region is no more than Λ.

Lemma 4.1. *Under the above NSI structure, the traffic process* $\{A[t]\}_t$ *is supportable if* $(1+\epsilon)E[A[t]] \in \Lambda$ *for some* $\epsilon > 0$*, and only if* $E[A[t]] \in \Lambda$*.*

The key step in the proof of Lemma 4.1 (similar to Lemma 7 in [27]) is to build a time shared stationary policy corresponding to any given rate point $\lambda \in \Lambda$. This is carried out by using the steady-state queue-length distribution, of an arbitrary scheduling policy that stabilizes λ, as the distribution of a source of global, common randomness. This randomization is then used by each transmitter to pick a suitable static-split scheduling rule, and enable the transmitters to appropriately time share their transmit decisions to stabilize λ. The proof technique also hints at the fact that shared delayed queue and channel state information thus can, in fact, act as a source of common randomness – a fact that is exploited crucially in Section 4.3 to design a throughput-optimal scheduling policy. We refer the reader to the appendix for the detailed proof of the lemma.

4.2 Critical NSI

As defined in the system model (Section 3), $\tau_l(h)$ represents the delay with which the latest queue state and channel state information of link h is available at link l. We expect that for link l at time slot t, all the latest delayed channel state information from other users (i.e., $\{C_k[t-\tau_k(l)] : k \in \mathcal{L}, k \neq l\}$) is the information most useful with regard to the current channel states of the other users. In what follows, we introduce the important concept of critical NSI for the network – essentially *all the latest delayed channel state information observed by every user in the network* – which is later used to develop a throughput-optimal scheduling policy in which each user makes scheduling decisions just based on the critical NSI available to itself.

Given $\mathbf{C}[t](0 : \tau_{max})$, the critical set of information related to link l is defined as the the channel state information at times $\{t - \tau_k(l)\}_{k \in \mathcal{L}: k \neq l}$. Let us denote the critical NSI of the network at time t as $\mathcal{CS}(.)$, which can be expressed mathematically as follows

$$\mathcal{CS}(\mathbf{C}[t](0 : \tau_{max})) := \{\{C_l[t - \tau_k(l)]\}_{k \in \mathcal{L}: k \neq l}\}_{l \in \mathcal{L}}.$$

For every $l \in \mathcal{L}$, we define the critical NSI available at transmitter l as follows:

$$\mathcal{CS}_l(\mathbf{C}[t](0 : \tau_{max})) := \mathcal{CS}(\mathbf{C}[t](0 : \tau_{max})) \bigcap \mathcal{P}_l(\mathbf{C}[t](0 : \tau_{max})).$$

Recalling the example in Section 2, we have the $\tau_{max} = 2$, and critical set at time t is $\{C_A[t-1], C_B[t-1], C_C[t-1], C_C[t-2]\}$. Thus at time t, the critical set available at transmitter A is $\{C_A[t-1], C_B[t-1], C_C[t-1], C_C[t-2]\}$, at B is $\{C_A[t-1], C_B[t-1], C_C[t-2]\}$, and at C is $\{C_A[t-1], C_B[t-1], C_C[t-1], C_C[t-2]\}$.

We now describe the queue dynamics at each transmitter node. Each transmitter maintains a queue of packets corresponding to its destination. Once a packet is sent, this node does not flush the packets from its queues until an acknowledgment is received indicating successful reception. This acknowledgment (ACK) is received with some delay, and this delay is consistent with the critical channel state information delays. By this, we mean that the information contained in the acknowledgment, either explicitly (in the header) or implicitly (via the observation that presence of the ACK/NACK "encodes" the interfering links' critical NSI) does not contain additional NSI as compared to the nodes' critical NSI. This is to ensure that by learning based on queue lengths and ACKs, nodes cannot get more NSI than the critical NSI. This consistency of ACK "state" information can be characterized explicitly where each transmitter node has potentially a different ACK delay, which is "naturally" consistent with the critical NSI in the system. However, in this paper, for notational simplicity, we assume that the acknowledgment is received only after τ_{max} time slots (thus trivially ensuring that the ACK information is consistent with the critical NSI). The queue dynamics therefore is represented as follows,

$$Q_l[t+1] = (Q_l[t] + A_l[t] - S_l[t-\tau_{max}])^+,$$

where $S_l[t]$ denotes the number of packets successfully transmitted at time t.

4.3 A Threshold-based Throughput-optimal Scheduling Algorithm:

The two ideas discussed so far – (a) that global, common randomness helps span the stability region (Section 4.1), and (b) that critical delayed NSI at each transmitter is as good as all available delayed NSI (Section 4.2), are used in this section to design a threshold-based decentralized scheduling algorithm. This algorithm uses shared, delayed queue-length information as a source of common randomness, and along with local threshold-type static scheduling with only critical NSI at each transmitter, achieves throughput-optimality, i.e., stabilizes the network for all arrival rates in the interior of the throughput region Λ. Note that this is done without any explicit knowledge of the arrival rates; thus the shared queue lengths distribute themselves in such a way as to provide the "right" time sharing fractions necessary to stabilize any valid vector of arrival rates.

The algorithm we propose consists of two steps. At each time slot,

- **Step 1:** All the transmitters compute threshold functions based on common NSI available at all transmitters. These threshold functions, one for each transmitter, map the respective transmitter's critical NSI to a corresponding threshold value, and are computed by solving the following optimization problem:

$$\arg\max_{\mathbf{T}} \sum_{l \in \mathcal{L}} Q_l(t-\tau_{max}) R_{l,\tau_{max}}(\mathbf{T}), \tag{1}$$

  where

$$R_{l,\tau}(\mathbf{T}) := E\big[C_l[t] 1_{C_l[t] \geq T_l(.)} \big(\gamma_l + (1-\gamma_l) \prod_{m \in I_l} 1_{C_m[t] < T_m(.)}\big) | \mathbf{C}[t-\tau]\big], \tag{2}$$

  and $T_l(.) := T_l(\mathcal{CS}_l(\mathbf{C}[t](0:\tau_{max})))$.

– **Step 2:** Each transmitter observes its current critical NSI, evaluates its threshold function (found in Step 1) at this critical NSI, and attempts to transmit if and only if its current channel rate exceeds the threshold value, i.e., when

$$C_l[t] \geq T_l(\mathcal{CS}_l(\mathbf{C}[t](0:\tau_{max}))).$$

The main result of this section is the following, which states that the above distributed scheduling algorithm stabilizes *any* arrival rate vector in the system throughput region Λ.

Theorem 4.2. *The proposed algorithm is throughput-optimal.*

Proof outline. We provide a sketch of the proof here – the detailed proof can be found in the appendix. The crux of the proof lies in the following lemma, which shows that solving an optimization problem locally in each time slot results in (globally) throughput-optimal scheduling.

Lemma 4.3. *Consider the optimization problem*

$$\arg\max_{\mathbf{F}(.)} \sum_{l \in \mathcal{L}} Q_l(t-\tau_{max}) R_{l,\tau_{max}}(\mathbf{F}(.)), \tag{3}$$

where

$$R_{l,\tau}(\mathbf{F}(.)) := E\big[C_l[t]F_l(.)\big(\gamma_l + (1-\gamma_l)\prod_{m \in I_l}(1-F_m(.))\big)|\mathbf{C}[t-\tau]\big],$$

and $F_l(.) := F_l(\mathcal{P}_l(\mathbf{C}[t](0:\tau_{max}))) \in \{0,1\}$ *for each* $l \in \mathcal{L}$*. If each transmitter* l *at time* t *is scheduled to transmit whenever the optimizing* $F_l^*(\mathcal{P}_l(\mathbf{C}[t](0:\tau_{max}))) = 1$*, then any* $\boldsymbol{\lambda}$ *that satisfies* $(1+\epsilon)\boldsymbol{\lambda} \in \Lambda$ *for* $\epsilon > 0$ *is supportable.*

Next, we show that the optimizing solution (i.e., the functions $F_l^*(.)$ of the individual NSI for all $l \in \mathcal{L}$)

1. Satisfies a *threshold* property, i.e.,

$$F_l^*(\mathcal{P}_l(\mathbf{C}[t](0:\tau_{max}))) = 1_{C_l[t] \geq T_l^*(\mathcal{P}_l(\mathbf{C}[t](0:\tau_{max})))},$$

2. Depends *only on the critical set* of NSI for each $l \in \mathcal{L}$, i.e.,

$$T_l^*(\mathcal{P}_l(\mathbf{C}[t](0:\tau_{max}))) = T_l^*(\mathcal{CS}_l(\mathbf{C}[t](0:\tau_{max}))).$$

The proof is completed by first noting that the proposed algorithm finds the best threshold-based scheduling decisions where each transmitter's thresholds are based only on its currently available NSI. And then using the two key properties of the time-varying channels - Markov property across time and independence property across the links in the network. □

5 Impact of Delayed NSI on the Throughput Region

With increasing delays in NSI between users, the information structure available to the users for scheduling becomes "coarser", hence we expect that system throughput is degraded. In this section, we present our second main result, which describes the extent to which the throughput region shrinks with larger delays in acquiring NSI from other users.

Let us denote the throughput region with NSI delays $\{\boldsymbol{\tau_l}\}_{l\in\mathcal{L}}$ (which we call our "canonical heterogeneous case") by Λ. For an integer $\tau \geq 0$, let Λ_τ denote the throughput region assuming that each link has its own instantaneous NSI and knows the NSI of other links in the network with a fixed delay of τ. We note that

$$\Lambda_{\tau_{max}} \subseteq \Lambda \subseteq \Lambda_{\tau_{min}}.$$

The following theorem – our second main result – quantifies the loss (gain) in the interior of the throughput region by using the minimum (maximum) homogeneously delayed NSI compared to the canonical heterogeneous case.

Theorem 5.1. *For integers $\tau_1, \tau_2 \geq 0$, let*

$$\alpha(\tau_1,\tau_2) := \frac{2Lk_o\beta(\tau_1,\tau_2)}{\sum_j c_j \min_i P_{ij}^{\tau_1}}, \tag{4}$$

where $k_o = (1 + M|I|(1-\gamma))(\sum c_i)$, $\beta(\tau_1,\tau_2) = max|P_{ij}^{\tau_1} - P_{kj}^{\tau_2}|$, $\gamma = min\ \gamma_l$ and $|I|$ denotes the maximum size of an interfering set of transmitters. Then,

$$(1-\alpha)\Lambda_{\tau_{min}} \subseteq \Lambda \subseteq (1-\alpha)^{-1}\Lambda_{\tau_{max}},$$

where $\alpha := \alpha(\tau_{min}, \tau_{max})$.

Theorem 5.1 is important for the following reasons:

1. It provides a lower bound on the fraction of the best-NSI throughput that can be attained as delays in NSI increase. Furthermore, the bound depends in a straightforward manner on the probability transition matrices of the system channels and the maximum number of interfering channels.

2. From the perspective of system design, the result of the theorem is useful since it specifies how much delay in the NSI can be tolerated while guaranteeing a minimum desired throughput capability for the system.

Proof. We prove a more general result which implies the above theorem: Given τ_1 and τ_2 such that $\tau_1 \leq \tau_2$, we have $\Lambda_{\tau_2} \supseteq (1-\alpha(\tau_1,\tau_2))\Lambda_{\tau_1}$.

For a NSI structure where each transmitter knows its current information and delayed information (by τ_1) of other links in the network, we have a scheduling policy based on thresholds (from Theorem 4.2) that is throughput-optimal. We will need the following useful lemma [7].

Lemma 5.2. *(Adapted from [7]) At any time* t*, given the common NSI* $(\mathbf{Q}[\mathbf{t}](\tau_1 : t), \mathbf{C}[\mathbf{t}](\tau_1 : t))$*, let* $\mathbf{T_1^*}$ *be the optimal set of thresholds calculated using the proposed algorithm and* $\mathbf{T_2}$ *be set of thresholds computed using a scheduling policy* S_ρ *such that the following condition holds (for some* $\rho \in [0, 1]$*):*

$$\sum_{l \in \mathcal{L}} Q_l(t-\tau_2) R_{l,\tau_1}(\mathbf{T_2}) \geq (1-\rho) \sum_{l \in \mathcal{L}} Q_l(t-\tau_2) R_{l,\tau_1}(\mathbf{T_1^*}).$$

Then, the scheduling policy S_ρ *can stabilize any arrival rate* $\boldsymbol{\lambda} \in (1-\rho)\Lambda_{\tau_1}$*.*

Let $\mathbf{T_2^*}$ be the set of thresholds computed using the proposed algorithm with the "degraded" NSI $(\mathbf{Q}[\mathbf{t}](\tau_2 : t), \mathbf{C}[\mathbf{t}](\tau_2 : t))$. Thus, $\mathbf{T_2^*}$ need not be an optimal set of thresholds for scheduling with the "non-degraded" partial NSI $(\mathbf{Q}[\mathbf{t}](\tau_1 : t), \mathbf{C}[\mathbf{t}](\tau_1 : t))$. Also, the proposed algorithm which uses only degraded partial NSI (τ_2 instead of τ_1) can stabilize the system for all arrival rates $\boldsymbol{\lambda} \in \Lambda_{\tau_2}$. We can write

$$R_{l,\tau_1}(\mathbf{T_2^*}) = E\left[C_l[t] 1_{C_l[t] \geq T^*_{2,l}} \big(\gamma_l + (1-\gamma_l) \prod_{m \in I_l} 1_{C_m[t] < T^*_{2,m}}\big) |\mathbf{C}[t-\tau_1]\right].$$

Since the random variables $C_l[t]$ and $C_m[t]$ are independent, we can rewrite the above expression as

$$\begin{aligned} R_{l,\tau_1}(\mathbf{T_2^*}) =& \gamma_l E\big[C_l[t] 1_{C_l[t] \geq T^*_{2,l}} |C_l[t-\tau_1]\big] + (1-\gamma_l) E\big[C_l[t] 1_{C_l[t] \geq T^*_{2,l}} |C_l[t-\tau_1] \times \\ & \prod_{m \in I_l} E\big[1_{C_m[t] < T^*_{2,m}} |C_m[t-\tau_1]\big]. \end{aligned}$$

Let P^τ_{ij} denote the τ-step transition probability of the channel state Markov chain from state c_i to state c_j. Rewriting the above expression in terms of P^τ_{ij}, we have

$$\begin{aligned} R_{l,\tau_1}(\mathbf{T_2^*}) =& \gamma_l \left(\sum_{i=1}^{M} c_i P^{\tau_1}_{.i} 1_{c_i \geq T^*_{2,l}} \right) + \\ & (1-\gamma_l) \left(\sum_{i=1}^{M} c_i P^{\tau_1}_{.i} 1_{c_i \geq T^*_{2,l}} \right) \prod_{m \in I_l} \left(\sum_{i=1}^{M} P^{\tau_1}_{.i} 1_{c_m \geq T^*_{2,l}} \right). \end{aligned} \tag{5}$$

We now state another lemma that bounds the difference between $R_{l,\tau_1}(\mathbf{T_2^*})$ and $R_{l,\tau_2}(\mathbf{T_2^*})$.

Lemma 5.3. $|R_{l,\tau_1}(\mathbf{T_2^*}) - R_{l,\tau_2}(\mathbf{T_2^*})| < k_o \beta(\tau_1, \tau_2)$.

Using Lemma 5.3, we have that

$$\begin{aligned} \sum_{l \in \mathcal{L}} Q_l(t-\tau_2) R_{l,\tau_1}(\mathbf{T_2^*}) \geq & \sum_{l \in \mathcal{L}} Q_l(t-\tau_2) \times \\ & (R_{l,\tau_2}(\mathbf{T_2^*}) - k_o \beta(\tau_1, \tau_2)). \end{aligned}$$

With the fact that $\mathbf{T_2^*}$ is an optimal set of thresholds for the proposed algorithm with NSI $(\mathbf{Q[t]}(\tau_2 : t), \mathbf{C[t]}(\tau_2 : t))$, we have

$$\sum_{l\in\mathcal{L}} Q_l(t-\tau_2)R_{l,\tau_1}(\mathbf{T_2^*}) \geq \sum_{l\in\mathcal{L}} Q_l(t-\tau_2) \times (R_{l,\tau_2}(\mathbf{T_1^*}) - k_o\beta(\tau_1,\tau_2)).$$

Employing Lemma 5.3 once again, we have

$$\begin{aligned}
&\sum_{l\in\mathcal{L}} Q_l(t-\tau_2)R_{l,\tau_1}(\mathbf{T_2^*}) \\
&\geq \sum_{l\in\mathcal{L}} Q_l(t-\tau_2)\big(R_{l,\tau_1}(\mathbf{T_1^*}) - 2k_o\beta(\tau_1,\tau_2)\big) \\
&\geq \sum_{l\in\mathcal{L}} Q_l(t-\tau_2)R_{l,\tau_1}(\mathbf{T_1^*}) - (LQ_{max})2k_o\beta(\tau_1,\tau_2) \\
&= \sum_{l\in\mathcal{L}} Q_l(t-\tau_2)R_{l,\tau_1}(\mathbf{T_1^*}) \left(1 - \frac{(LQ_{max})2k_o\beta(\tau_1,\tau_2)}{\sum_{l\in\mathcal{L}} Q_l(t-\tau_2)R_{l,\tau_1}(\mathbf{T_1^*})}\right).
\end{aligned}$$

Note that

$$\sum_{l\in\mathcal{L}} Q_l(t-\tau_2)R_{l,\tau_1}(\mathbf{T_1^*}) \geq \sum_{l\in\mathcal{L}} Q_l(t-\tau_2)(\min R_{l,\tau_1}(.)) \geq Q_{max}\sum_j c_j \min P_{ij}^{\tau_1}.$$

where the second inequality follows from the fact that summation is larger than maximum and $R_{l,\tau_1}(.)$ can be lower bounded by $\sum_j c_j \min_i P_{ij}^{\tau_1}$. Using the above inequality, we have that

$$\begin{aligned}
&\sum_{l\in\mathcal{L}} Q_l(t-\tau_2)R_{l,\tau_1}(\mathbf{T_2^*}) \\
&\quad\geq \sum_{l\in\mathcal{L}} Q_l(t-\tau_2)R_{l,\tau_1}(\mathbf{T_1^*}) \left(1 - \frac{2Lk_o\beta(\tau_1,\tau_2)}{\sum_j c_j \min P_{ij}^{\tau_1}}\right) \\
&\quad= (1-\alpha(\tau_1,\tau_2)) \sum_{l\in\mathcal{L}} Q_l(t-\tau_2)R_{l,\tau_1}(\mathbf{T_1^*}).
\end{aligned}$$

Using Lemma 5.2 now yields $\Lambda_{\tau_2} \supseteq (1-\alpha(\tau_1,\tau_2))\Lambda_{\tau_1}$ as desired. □

Finally, as a corollary of Theorem 5.1, we characterize the throughput region Λ_∞ as a fraction of the canonical throughput region Λ_τ. This represents the throughput in the "worst" possible delayed NSI case when each user has no NSI from any other user. For the sake of simplicity, we assume $P_{ij} > 0$ for all i and j. Even if P_{ij} are not all positive, we can find an integer m_o (since the Markov chain is aperiodic, irreducible and finite) such that $P_{ij}^{m_o} > 0$ for all i and j.

Corollary 5.4.

$$a) \quad \alpha(\tau_{min}, \tau_{max}) \leq \frac{4Lk_o(1-M\delta)^{\tau_{min}}}{\sum_j c_j \min_i P_{ij}^{\tau_{min}}},$$

$$b) \lim_{\tau_{max} \to \infty} \alpha(\tau_{min}, \tau_{max}) \leq \frac{2Lk_o(1-M\delta)^{\tau_{min}}}{\sum_j c_j \min_i P_{ij}^{\tau_{min}}},$$

where $\delta = \min_{ij} P_{ij}$.

Proof. The proof is based on the exponential convergence property [29] of finite-state Markov chains and detailed proof is presented in appendix. □

6 Simulations

In this section, we carry out numerical experiments using our proposed scheduling algorithms to illustrate the value of delayed network state information for throughput performance, and the efficacy of the Markov chain mixing bounds with homogeneously delayed NSI shown in Section 5.

6.1 Methodology

For our simulations, we consider a wireless network with $L = 10$ links. Complete interference is assumed with perfect collisions, i.e., $I_l = \mathcal{L} \setminus \{l\}$ and $\gamma_l = 0$ $\forall l$. Thus, for a transmission to be successful on a link l, we need all the other links in the network to be "silent", otherwise no packet is transmitted. The channel state process for each link l is assumed to be a two-state Markov chain on the state space $\{0, 1\}$, with uniform crossover probabilities p. Throughout this section, we assume symmetric traffic at all links, i.e., $A_l[t] \sim \text{Bernoulli}(\lambda)$ $\forall l$. Thus all the flows are single hop. The proposed algorithm in Section 4.3 is implemented in each time slot by solving the optimization (1) as a brute-force search over all possible thresholds $\mathbf{T}$.

6.2 Throughput Performance with Delayed NSI

We simulate in Matlab, the proposed algorithm (Section 4.3) on the 10-links wireless network described above for various values of the channel crossover probability p and NSI delays τ. For each value of p, Figure 4 depicts the maximum sum-throughput, i.e., $10 \times \lambda$, that the proposed algorithm achieves as a function of increasing homogeneous NSI delay $\tau = 0, 1, \ldots, 10$.

The maximum sum-rate when all nodes have instantaneous NSI (i.e., $\tau = 0$) is 1. In this case, our algorithm reduces to performing standard Max-Weight scheduling, and results in each of the the 10 nodes exclusively transmitting $\frac{1}{10}$-th of the time. Thereafter, as the information delay τ increases from 0 to 10, the sum-capacity decreases owing to more degradation in the nodes' NSI

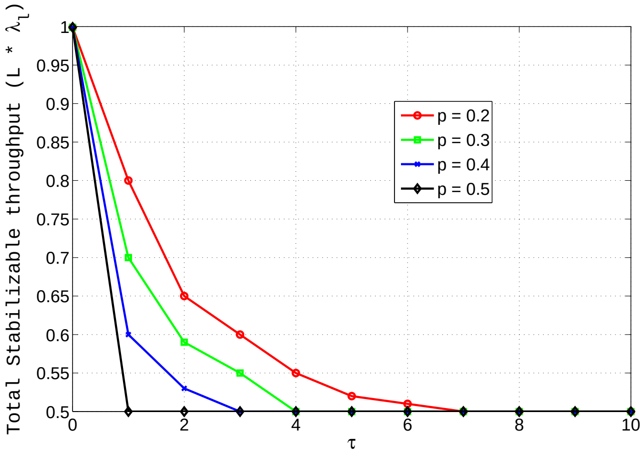

Fig. 4 Sum-throughput performance of the proposed algorithm for a 10-node wireless network with full collision interference. Traffic is symmetric with rate $\lambda_l = \lambda$, and channels are 2-state Markov with rates $\{0, 1\}$ packets/slot. Each curve depicts optimal sum-rate achievable for various homogeneous NSI delays $\tau = 0, 1, \ldots, 10$, for a different value of channel state crossover probability p.

structure. This sum-throughput degradation with delay occurs faster when p is closer to 0.5. Note that $p = 0.5$ represents channel states that are i.i.d. across time slots, so there is nothing to be gained from using delayed channel state information. Hence, the more rapid degradation of sum-rate closer to the i.i.d. channel regime is consistent with the fact that the dependence of current channel state decreases with p increasing to 0.5.

6.3 Throughput Region Mixing-based Bounds

We next turn to evaluating the efficacy of our bound $\alpha(\cdot, \cdot)$ from Theorem 5.1. Note that, from Section 5, the quantity $(1 - \alpha(\tau, \infty))$ *lower bounds* the factor by which the throughput region with "infinitely delayed NSI" (i.e., NSI with a very large delay) Λ_∞ is smaller relative to the throughput region Λ_τ with a homogeneous NSI delay τ. Thus, $(1 - \alpha(\tau, \infty)) = 1$ denotes that $\Lambda_\tau = \Lambda_\infty$, i.e., there is no further throughput degradation beyond a NSI delay of τ.

Figure 5 plots the calculated values of $(1 - \alpha(\tau, \infty))$ versus τ for various values of channel state crossover probabilities p. Observe that for $p = 0.5$, i.e., channel states independent across time slots, this quantity is always 1, which agrees with the fact that throughput with delayed NSI over independent channel states does not depend on the amount of delay. Also, note that the closer p is to 0.5, the faster $(1 - \alpha(\tau, \infty))$ approaches 1, i.e., the more rapidly the throughput region shrinks to Λ_∞ as noted in the previous section.

Figure 5 shows that the bounds of Theorem 5.1 are indicative of the level of NSI delay beyond which there is effectively little degradation of the system throughput. From Figure 5, when $p = 0.4$ observe that the bound is 1 for all $\tau > 5$. At the same time, from the simulation results of Figure 4, we notice that for $\tau \geq 3$ there is no further degradation in throughput. Thus, the bound derived in Theorem 5.1 provides an estimate of the NSI delay beyond which there is no further degradation in throughput.

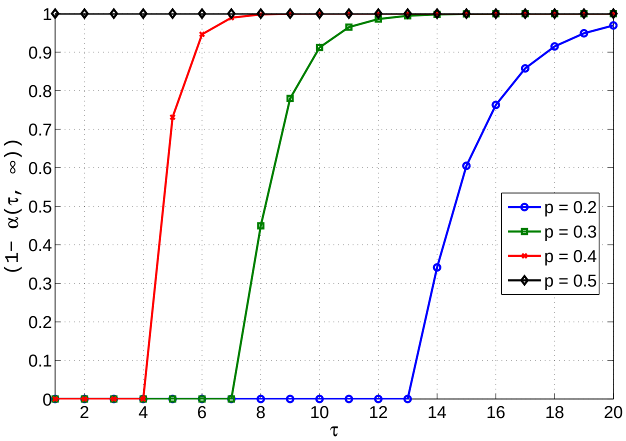

Fig. 5 Plot of $(1 - \alpha(\tau, \infty))$ versus homogeneous NSI delay τ. Each curve represents a different value of channel state crossover probability p. The quantity $(1 - \alpha(\tau, \infty))$ lower-bounds the factor by which the throughput region with "infinitely delayed NSI" (i.e., NSI with a very large delay) Λ_∞ is smaller relative to the throughput region Λ_τ with a homogeneous NSI delay τ. Thus, $(1 - \alpha(\tau, \infty)) = 1$ denotes that $\Lambda_\tau = \Lambda_\infty$, i.e., there is no further throughput degradation beyond a NSI delay of τ.

7 Discussion: Implementation Complexity

We remark that the throughput-optimal algorithm developed in Section 4.3 is computationally complex. The solution which we provide is in terms of an integer program with a high complexity if solved in a brute-force manner. We have numerically evaluated the run times of our algorithm using Matlab simulations (but without any approximations to reduce compelxity). The run time of algorithm for network sizes with 5, 10, 15 and 20 links are 4, 110, 3900 and 81400 ms respectively. Note that the time taken roughly grows exponentially with the number of links in the network. A simple further approximation is to ignore the far-away links delayed channel state information and just use the expected values instead. With this approximation, as the network scales the complexity at an individual node will not scale after a point in network size but will incur a loss in throughput. However, the above calculations do not use this approximation and are computed using the "brute-force" exact solution. It is possible that there could be sophisticated methods that reduce this complexity; instead, we have studied complexity reductions via structural properties of the solution. In particular, our approach towards complexity reduction in this work is the following:

1. We characterize the minimal/critical information necessary (and sufficient) for throughput-optimality (Section 4.2). This is significant as the complexity is exponential in the size of the information set.
2. We show that *threshold-type policies* are sufficient for throughput-optimality (Section 4.3). Note that in general, the throughput-optimal policy in each time slot at each node is a mapping from observed delayed channel and queue state to a scheduling decision (i.e., transmit/no-transmit). However, we show that threshold-type mappings, i.e., transmit only if the current channel state exceeds a threshold, are sufficient for achieving throughput-optimality. This reduces the complexity of the algorithm from exponential to linear in the number of channel states, though we note that the complexity remains exponential in the network size.
3. To obtain further complexity reductions, we consider alternative (sub-optimal) schemes based on the use of "degraded common information" (Section 5). The technical challenge here is in characterizing the loss in throughput, and we develop novel Markov chain mixing-based techniques to do so.

8 Conclusion

In this paper, we have addressed the problem of distributed scheduling in wireless networks with Markovian channels and heterogeneously delayed NSI. We have proposed a threshold-type distributed scheduling algorithm that is provably throughput-optimal. We have shown that thresholds depend only up on the critical set of NSI. We have also characterized the effect of delayed NSI on the network throughput region.

Acknowledgements

This work was partially supported by NSF grants CNS-0721380, CNS-0831756, the DARPA ITMANET program and DTRA grant HDTRA1-09-1-0055. We thank the anonymous reviewers for their valuable comments and suggestions.

References

1. A. Reddy, S. Banerjee, A. Gopalan, S. Shakkottai, and L. Ying, "Wireless scheduling with heterogeneously delayed network-state information," in *Proc. Ann. Allerton Conf. Communication, Control and Computing*, 2010.
2. L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," *IEEE Trans. Automat. Contr.* , 1992, pp. 1936–48.

3. M. Andrews, K. Kumaran, K. Ramanan and A.L. Stolyar and R. Vijayakumar and P. Whiting, “CDMA data QoS scheduling on the forward link with variable channel conditions,” *Bell Labs Tech. Memo*, April 2000.
4. L. Tassiulas and A. Ephremides, “Dynamic server allocation to parallel queues with randomly varying connectivity,” *IEEE Trans. Inform. Theory*, vol. 39, pp. 466–478, 1993.
5. S. Shakkottai and A. Stolyar, “Scheduling for multiple flows sharing a time-varying channel: The exponential rule,” *Ann. Math. Statist.*, vol. 207, pp. 185–202, 2001.
6. A. Stolyar, “Maximizing queueing network utility subject to stability: Greedy primal-dual algorithm,” *Queueing Systems*, vol. 50, pp. 401–457, August 2005.
7. A. Eryilmaz, R. Srikant and J. Perkins, “Stable scheduling policies for fading wireless channels,” *IEEE/ACM Trans. Network.*, vol. 13, pp. 411–424, April 2005.
8. M. J. Neely, E. Modiano and C. E. Rohrs, “Power and server allocation in a multi-beam satellite with time varying channels,” *Proc. IEEE Infocom*, vol. 3, pp. 1451–1460, 2002.
9. M. J. Neely, E. Modiano and C. Li, “Fairness and optimal stochastic control for heterogeneous networks,” *Proc. IEEE Infocom*, vol. 3, pp. 1723–1734, March 2005.
10. X. Lin and S. Rasool, “Constant-Time Distributed Scheduling Policies for Ad HocWireless Networks,” *Proc. Conf. on Decision and Control*, 2006.
11. C. Joo and N. Shroff, “Performance of Random Access Scheduling Schemes in Multihop Wireless Networks,” *Proc. IEEE Infocom*, 2007.
12. S. Rajagopalan, J. Shin, D. Shah, “Network Adiabatic Theorem: An Efficient Randomized Protocol for Contention Resolution,” *Proc. Ann. ACM SIGMETRICS Conf.*, 2009.
13. L. Jiang and J. Walrand, “A Distributed CSMA Algorithm for Throughput and Utility Maximization in Wireless Networks,” *Proc. Ann. Allerton Conf. Communication, Control and Computing*, 2008.
14. J. Liu, Y. Yi, A. Proutiere, M. Chiang and H. V. Poor, “Maximizing utility via random access without message passing,” *Microsoft Research Technical Report*, 2008.
15. S. Rajagopalan, D. Shah and J. Shin, “Network adiabatic theorem: An efficient randomized protocol for contention resolution,” *Proc. Ann. ACM SIGMETRICS Conf.*, pp. 133–144, 2009.
16. J. Ni, B. Tan and R. Srikant, “Q-CSMA: Queue-length based CSMA/CA algorithms for achieving maximum throughput and low delay in wireless networks,” *Proceedings of IEEE INFOCOM Mini-Conference*, 2010.
17. S. Ahmad, L. Mingyan, T. Javidi, Q. Zhao and B. Krishnamachari, “Optimality of Myopic Sensing in Multichannel Opportunistic Access,” *Information Theory, IEEE Transactions on*, vol. 55, pp. 4040–4050, 2009.
18. S. Guha, K. Munagala and S. Sarkar, “Performance Guarantees Through Partial Information Based Control in Multichannel Wireless Networks,” *http://www.seas.upenn.edu/∼swati/report.pdf*, 2006.
19. N. Chang and M. Liu, “Optimal channel probing and transmission scheduling for opportunistic spectrum access,” *ACM Int. Conf. on Mobile Computing and Networking (MobiCom)*, 2007.
20. A. Gopalan, C. Caramanis and S. Shakkottai, “On Wireless Scheduling with Partial Channel State Information,” *IEEE Trans. Inform. Theory*, 2011.
21. P. Chaporkar, A. Proutiere, H. Asnani, and A. Karandikar, “Scheduling with limited information in wireless systems,” in *ACM Mobihoc*, 2009, pp. 75–84.
22. A. Pantelidou, A. Ephremides, and A. Tits, “Joint scheduling and routing for ad-hoc networks under channel state uncertainty,” in *Intl. Symposium on Modeling and Optimization in Mobile, Ad-Hoc and Wireless Networks (WiOpt)*, April 2007, pp. 1–8.
23. K. Kar, X. Luo and S. Sarkar, “Throughput-optimal Scheduling in Multichannel Access Point Networks under Infrequent Channel Measurements,” *Proceedings of IEEE Infocom*, 2007.
24. J. Chen, R. A. Berry, and M. L. Honig, “Limited feedback schemes for downlink OFDMA based on sub-channel groups,” *IEEE J. Sel. Areas Commun.*, vol. 26, pp. 1451–1461, 2008.
25. M. Ouyang and L. Ying, “On scheduling in multi-channel wireless downlink networks with limited feedback,” *Proc. Ann. Allerton Conf. Communication, Control and Computing*, pp. 455- 469, 2009.

26. M. Ouyang and L. Ying, "On Optimal Feedback Allocation in Multichannel Wireless Downlinks," *ACM Mobihoc*, 2010.
27. L. Ying and S. Shakkottai, "On Throughput Optimality with Delayed Network-State Information," *Technical Report*, 2008.
28. L. Ying and S. Shakkottai, "Scheduling in Mobile Ad Hoc Networks with Topology and Channel-State Uncertainty," *Proc. IEEE Infocom*, pp. 2347–2355, 2009.
29. P. Billingsley, "Probability and Measure," *Wiley Interscience Publication*, 1994.
30. S. Asmussen, "Applied Probability and Queues," *Springer-Verlag*, New York, 2003.

Appendix

Proof. (Lemma 4.1) First, assume that the arrival rates $E[\mathbf{A}[t]]$ are such that $\tilde{\lambda} := (1+\epsilon)E[\mathbf{A}[t]] \in \Lambda$ for some $\epsilon > 0$. Then, by the definition of the region Λ, it follows that we can construct a set of channel state dependent policies (i.e., f_l's) and "time-share" over those policies to get a long-term service rate of $\tilde{\lambda}$ (analogous to the proof of Theorem 1 in [3]). This, in turn, ensures that the network is stochastically stable.

Now for the other direction, given $\mathbf{A}[t]$ is supportable, by definition, there exists a scheduling algorithm $\mathcal{F}$ which makes the network stable. Since the system state Markov chain $\mathbf{Y}^{\mathcal{F}}[t]$ is positive recurrent, it exhibits a stationary distribution. Let us denote the scheduling decision under policy $\mathcal{F}$ as $S^{\mathcal{F}}(\mathbf{Y}[t])$. We will now construct a time-sharing scheduling policy $\mathcal{F}_s$ that depends on the steady state distribution of queue lengths and channel states (denoted as $\pi(\mathbf{y}), y = \{\mathbf{q}(0:\tau_{max}), \mathbf{c}(0:\tau_{max})\}$) under policy $\mathcal{F}$. Let $r(\mathbf{y}) = \Pr(\mathbf{q}|\mathbf{c})$, computed using $\pi(\mathbf{y})$.

At each time, when delayed channel state information $\mathbf{C}[t](0:\tau_{max}) = \mathbf{c}$, the policy $\mathcal{F}_s$ probabilistically selects the scheduling decision $S^{\mathcal{F}}(\mathbf{q},\mathbf{c})$ with probability $r(\mathbf{y} = (\mathbf{q},\mathbf{c}))$. We observe that the time-sharing policy $\mathcal{F}_s$ allocates the same amount of service to each link as $\mathcal{F}$. Since $\mathbf{A}[t]$ can be supported by the time sharing policy, we have that $E[\mathbf{A}[t]] \in \Lambda$.

□

Proof. (Theorem 4.2) The proof is split into two parts. Part one proves the threshold property of optimal solution and part two shows that optimal solution depends only up on the critical set of NSI. In other words, part two shows that the optimizing solution is independent of extra channel state NSI available at each node other than the critical NSI. (Proof : Part 1) We first show the following threshold property for the optimal solution to the optimization problem defined in equation (3),

$$F_l^*(\mathcal{P}_l(\mathbf{C}[t](0:\tau_{max}))) = 1_{C_l[t] \geq T_l^*(\mathcal{P}_l(\mathbf{C}[t](0:\tau_{max})))},$$

Let us assume that we partly know the optimal solution. In particular, we assume that we are given the entire $\{F_l^*(\mathcal{P}_l(\mathbf{C}[t](0:\tau_{max})))\}_{l\in\mathcal{L}}$ except $F_k^*(\mathcal{P}_k(\mathbf{C}[t](0:\tau_{max})))$ at two different values of NSI $(\mathcal{P}_k(\mathbf{C}[t](0:\tau_{max})) = \{(C_k[t]=c_i,\mathbf{r}),(C_k[t]=c_j,\mathbf{r})\}$)available at transmitter k.

To find $F_k^*(C_k[t]=c_i,\mathbf{r}), F_k^*(C_k[t]=c_j,\mathbf{r})$, we can solve the optimization (3) with other variables being fixed to the optimal solution. Consider the function that needs to be optimized:

$$\sum_l Q_l E\big[C_l[t]F_l(.)\big(\gamma_l+(1-\gamma_l)\prod_{m\in I_l}(1-F_m(.))\big)|\mathbf{c}[t-\tau_{max}]\big].$$

Expanding this out, we can write this as

$$\sum_l Q_l \sum_{\mathbf{z}\in\mathcal{C}^{L\tau_{max}}} Pr(\mathbf{z}|\mathbf{c}[t-\tau_{max}])C_l(\mathbf{z})F_l(\mathbf{z})\big(\ \gamma_l+(1-\gamma_l)\prod_{m\in I_l}(1-F_m(\mathbf{z}))\big).$$

Note that $\mathbf{z}\in\mathcal{C}^{L\tau_{max}}$ corresponds to one particular realization of channel states of the network for the past τ_{max} slots. Since the variables in the above optimization are only $F_k(C_k[t]=c_i,\mathbf{r})$ and $F_k(C_k[t]=c_j,\mathbf{r})$, we ignore the terms in the summation that do not involve these variables (as they are constant and do not affect the arg max). Let A_i denote the set $\{\mathbf{z}:\mathbf{z}\in\mathcal{C}^{L\tau_{max}},\mathcal{P}_k(\mathbf{z})=(c_i,\mathbf{r})\}$. The new function we now have is:

$$\begin{aligned}&Q_k\sum_{\mathbf{z}\in A_i\cup A_j} Pr(\mathbf{z}|\mathbf{c}[t-\tau_{max}])C_k(\mathbf{z})F_k(\mathbf{z})\big(\gamma_k+(1-\gamma_k)\prod_{m\in I_k}(1-F_m(\mathbf{z}))\big)\\ &+\sum_{l:l\in I_k} Q_l\sum_{\mathbf{z}\in A_i\cup A_j} Pr(\mathbf{z}|\mathbf{c}[t-\tau_{max}])C_l(\mathbf{z})F_l(\mathbf{z})\big(\ \gamma_l+(1-\gamma_l)\prod_{m\in I_l}(1-F_m(\mathbf{z}))\big).\end{aligned}$$

From the above expression, we observe that the above optimization for finding two variables $F_k(c_i,\mathbf{r}), F_k(c_j,\mathbf{r})$ splits into two independent optimization problems. First, let us consider the function that needs to be optimized to get $F_k(c_i,\mathbf{r})$:

$$\begin{aligned}&Q_kF_k(c_i,\mathbf{r})c_i\sum_{\mathbf{z}\in A_i}\Big(Pr(\mathbf{z}|\mathbf{c}[t-\tau_{max}])\big(\gamma_k+(1-\gamma_k)\prod_{m\in I_k}(1-F_m(\mathbf{z}))\big)\Big)+\\ &(1-F_k(c_i,\mathbf{r}))\sum_{l:l\in I_k}Q_l\sum_{\mathbf{z}\in A_i}\Big(Pr(\mathbf{z}|\mathbf{c}[t-\tau_{max}])\times\\ &\qquad\qquad C_l(\mathbf{z})F_l(\mathbf{z})\big(\gamma_l+(1-\gamma_l)\prod_{m\in I_l,m\neq k}(1-F_m(\mathbf{z}))\big)\Big).\end{aligned}$$

From the above equation, we observe that the optimization function is *linear* in the variable $F_k(c_i,\mathbf{r})$. Using the fact that channels are independent across links, we have the above function of the form $Pr(C[t]=c_i|\mathbf{r})(ac_iF_k(c_i,\mathbf{r})+b(1-F_k(c_i,\mathbf{r})))$, where parameters a and b are independent of value of c_i. Similarly, we can show that the function that needs to be optimized for variable $F_k(c_j,\mathbf{r})$ is of form $ac_jF_k(c_j,\mathbf{r})+b(1-F_k(c_i,\mathbf{r}))$. Thus the optimal solution is of the form

$$F_k^*(c_i,\mathbf{r})=\begin{cases}1 & \text{if } ac_i\geq b,\\ 0 & \text{if } ac_i<b.\end{cases}$$

The above solution implies that if $c_j \geq c_i$ and $F_k^*(c_i, \mathbf{r}) = 1$, then $F_k^*(c_j, \mathbf{r}) = 1$. This proves the threshold nature of optimal solution.

(Proof: Part 2) Let us consider the original function that needs to be optimized (3)

$$\sum_l Q_l E\big[C_l[t] F_l(.)\big(\gamma_l + (1-\gamma_l) \prod_{m \in I_l} (1 - F_m(.))\big) | \mathbf{c}[t - \tau_{max}]\big].$$

Expanding the above expression, we have

$$\sum_l Q_l \sum_{\mathbf{z} \in \mathcal{C}^{L\tau_{max}}} Pr(\mathbf{z}|\mathbf{c}[t-\tau_{max}]) C_l(\mathbf{z}) F_l(\mathbf{z}) \big(\gamma_l + (1-\gamma_l) \prod_{m \in I_l} (1 - F_m(\mathbf{z}))\big).$$

First, observe that each variable in the above expression has a unique notation. In particular, a variable that is associated with link l and a particular value of channel state $\mathbf{z} \in \mathcal{C}^{L\tau_{max}}$ is denoted by $F_l(\mathbf{z})$ and more specifically $F_l(\mathcal{P}_k(\mathbf{z}))$. Consider a $\tau (\neq \tau_1(l) \forall l)$ and let the set $B(\tau) = \{\mathbf{z} \in \mathcal{C}^{L\tau_{max}} : C_1[\tau] = c1 \,\text{or}\, C_1[\tau] = c2\}$ denote the set of variables whose optimal values are not known. In other words, assume that the optimal values of all the variables are known to us except those in set B.

We define the sets $B1 = \{\mathbf{z} \in \mathcal{C}^{L\tau_{max}} : C_1[\tau] = c1\}$ and $B2 = \{\mathbf{z} \in \mathcal{C}^{L\tau_{max}} : C_1[\tau] = c2\}$. The sets $B1$ and $B2$ satisfy $B = B1 \cup B2$. We now observe that the optimization functions that depend on variables in sets $B1$ and $B2$ are exactly identical up to a scaling factor. Therefore the optimal solutions are also equal and thus we have that optimal solution is independent of channel state information that is not critical NSI. □

Proof. (Lemma 4.3) Consider the following Lyapunov function $V[t]$, of the system state $\mathbf{Y}^{\mathcal{F}}[t]$, as follows,

$$V[t] := \sum_{l \in \mathcal{L}} Q_l^2[t].$$

We thus have,

$$E[V[t+1] - V[t] | \mathbf{Q}[t-\tau_{max}], \mathbf{C}[t-\tau_{max}]] = \\ E[\sum_{l \in \mathcal{L}} (\Delta Q_l[t])(Q_l[t+1] + Q_l[t]) | \mathbf{Q}[t-\tau_{max}], \mathbf{C}[t-\tau_{max}]]$$

where $\Delta Q_l[t]$ is the difference $Q_l[t+1] - Q_l[t]$. Using the fact that arrivals and services are bounded in each time slot, we have

$$E\big[V[t+1] - V[t] | (\mathbf{Q}[t-\tau_{max}], \mathbf{C}[t-\tau_{max}])\big] \leq K + \\ E\big[\sum_{l \in \mathcal{L}} (\Delta Q_l[t])(2Q_l[t-\tau_{max}]) | (\mathbf{Q}[t-\tau_{max}], \mathbf{C}[t-\tau_{max}])\big].$$

Using the queue update equation, we have

$$E\big[V[t+1]-V[t]|(\mathbf{Q}[t-\tau_{max}],\mathbf{C}[t-\tau_{max}])\big] \leq K+ \\ E\big[\sum_{l\in\mathcal{L}}(\mathbf{R}_{l,\tau_{max}}(\mathbf{F}^*(.)))(2Q_l[t-\tau_{max}])|(\mathbf{Q}[t-\tau_{max}],\mathbf{C}[t-\tau_{max}])\big]. \tag{6}$$

Since $(1+\epsilon)\boldsymbol{\lambda}\in\Lambda$, there exists $\{\bar{\eta}(\mathbf{c})\}_{\mathbf{c}}$ such that

$$\sum_{\mathbf{c}\in\mathcal{C}^L}\pi(\mathbf{c})\big((1+\epsilon)\lambda_l-\bar{\eta}_l(\mathbf{c})\big)\leq 0.$$

From the scheduling algorithm optimization, we also have that

$$E\big[(\sum_{l\in\mathcal{L}}(\mathbf{R}_{l,\tau_{max}}(\mathbf{F}^*(.)))|\mathbf{C}[t-\tau_{max}]- \\ \bar{\eta}_l(\mathbf{C}[t-\tau_{max}]))Q_l[t-\tau_{max}]\big]\leq 0.$$

Taking the expectation on both sides of inequality (6) over $\mathbf{C}[t-\tau_{max}]$, we have that

$$E\big[V[t+1]-V[t]|\mathbf{Q}[t-\tau_{max}]\big]\leq K_1-2\epsilon\sum_l Q_l[t-\tau_{max}]\lambda_l.$$

It now follows from the standard Foster-Lyapunov drift criterion [30] that the network is stochastically stable.

□

Proof. (Lemma 5.3) From the equation (5), we have

$$|R_{l,\tau_1}(\mathbf{T_2^*})-R_{l,\tau_2}(\mathbf{T_2^*})| = \\ |(\sum_{i=1}^{M}c_iP_{.i}^{\tau_1}1_{c_i\geq T_{2,l}^*})\Big(\gamma_l+(1-\gamma_l)\prod_{m\in I_l}(\sum_{i=1}^{M}P_{.i}^{\tau_1}1_{c_m\geq T_{2,l}^*})\Big)- \\ (\sum_{i=1}^{M}c_iP_{.i}^{\tau_2}1_{c_i\geq T_{2,l}^*})\Big(\gamma_l+(1-\gamma_l)\prod_{m\in I_l}(\sum_{i=1}^{M}P_{.i}^{\tau_2}1_{c_m\geq T_{2,l}^*})\Big)|.$$

Let us denote the summation $\sum_{i=1}^{M}c_iP_{.i}^{\tau_1}1_{c_i\geq T_{2,l}^*}$ by $f_l(\tau_1)$ and the summation $\sum_{i=1}^{M}P_{.i}^{\tau_1}1_{c_m\geq T_{2,l}^*}$ by $g_m(\tau_1)$. Thus, we have

$$|R_{l,\tau_1}(\mathbf{T_2^*})-R_{l,\tau_2}(\mathbf{T_2^*})| = \\ |f_l(\tau_1)\big(\gamma_l+(1-\gamma_l)\prod_{m\in I_l}g_m(\tau_1)\big)-f_l(\tau_2)\big(\gamma_l+(1-\gamma_l)\prod_{m\in I_l}g_m(\tau_2)\big)|.$$

Expanding out the terms with γ_l and $(1-\gamma_l)$, we have

$$|R_{l,\tau_1}(\mathbf{T_2^*})-R_{l,\tau_2}(\mathbf{T_2^*})| = \\ |\gamma_l(f_l(\tau_1)-f_l(\tau_2))+(1-\gamma_l)\big(f_l(\tau_1)\prod_{m\in I_l}g_m(\tau_1)\big)-f_l(\tau_2)\prod_{m\in I_l}g_m(\tau_2)\big)|.$$

Using the triangle inequality, we have the following inequality,

$$
\begin{aligned}
&|R_{l,\tau_1}(\mathbf{T_2^*}) - R_{l,\tau_2}(\mathbf{T_2^*})| \leq \\
&\qquad |\gamma_l(f_l(\tau_1) - f_l(\tau_2))| + (1-\gamma_l)|\big(f_l(\tau_1)\prod_{m\in I_l} g_m(\tau_1)\big) - f_l(\tau_2)\prod_{m\in I_l} g_m(\tau_2)\big)|.
\end{aligned}
$$

By adding and subtracting the term $f_l(\tau_2)\prod_{m\in I_l} g_m(\tau_1)$, we have

$$
\begin{aligned}
|R_{l,\tau_1}(\mathbf{T_2^*}) - R_{l,\tau_2}(\mathbf{T_2^*})| \leq \\
&\gamma_l|(f_l(\tau_1) - f_l(\tau_2))| + (1-\gamma_l)|f_l(\tau_1)\prod_{m\in I_l} g_m(\tau_1) - f_l(\tau_2)\prod_{m\in I_l} g_m(\tau_1) + \\
&f_l(\tau_2)\prod_{m\in I_l} g_m(\tau_1) - f_l(\tau_2)\prod_{m\in I_l} g_m(\tau_2)|.
\end{aligned}
$$

Using the triangle inequality results in

$$
\begin{aligned}
|R_{l,\tau_1}(\mathbf{T_2^*}) - R_{l,\tau_2}(\mathbf{T_2^*})| \leq \\
&\gamma_l|(f_l(\tau_1) - f_l(\tau_2))| + (1-\gamma_l)|\big(f_l(\tau_1) - f_l(\tau_2)\big)\prod_{m\in I_l} g_m(\tau_1)| + \\
&(1-\gamma_l)|f_l(\tau_2)\prod_{m\in I_l} g_m(\tau_1) - f_l(\tau_2)\prod_{m\in I_l} g_m(\tau_2)|.
\end{aligned}
$$

Let the set I_l be expressed as $\{m_1, m_2, m_3, ..., m_l\}$. By iterating the above idea of adding and subtracting terms on the second component of the above expression and using the triangle inequality, we have

$$
\begin{aligned}
|R_{l,\tau_1}(\mathbf{T_2^*}) - R_{l,\tau_2}(\mathbf{T_2^*})| \leq \\
&\gamma_l|(f_l(\tau_1) - f_l(\tau_2))| + (1-\gamma_l)|\big(f_l(\tau_1) - f_l(\tau_2)\big)\prod_{m\in I_l} g_m(\tau_1)| + ...+ \\
&|f_l(\tau_2)(g_{m_l}(\tau_1) - g_{m_l}(\tau_2))\prod_{k:m_k\in I_l, k\neq l} g_{m_k}(\tau_2)|.
\end{aligned}
$$

Using the following upper bounds, $|f_l(\tau_1) - f_l(\tau_2)| \leq \sum c_i \beta(\tau_1, \tau_2)$, $|g_l(\tau_1) - g_l(\tau_2)| \leq M\beta(\tau_1, \tau_2)$ and $|f_l(\tau_1)| \leq \sum c_i$, we have

$$
\begin{aligned}
|R_{l,\tau_1}(\mathbf{T_2^*}) - R_{l,\tau_2}(\mathbf{T_2^*})| \leq \\
&(\sum c_i)\beta(\tau_1, \tau_2) + (1-\gamma_l)(\sum c_i)|I|M\beta(\tau_1, \tau_2) \\
&= (1 + M|I|(1-\gamma_l))(\sum c_i)\beta(\tau_1, \tau_2). \\
&\leq (1 + M|I|(1-\alpha))(\sum c_i)\beta(\tau_1, \tau_2).
\end{aligned}
$$

where the last inequality follows from definition of $\gamma = \min \gamma_l$. $\square$

Proof. (Corollary 5.4) From equation (4), we have

$$\alpha(\tau_1,\tau_2) := \frac{2Lk_o\beta(\tau_1,\tau_2)}{\sum_j c_j \min_i P_{ij}^{\tau_1}}.$$

It is sufficient to prove that $\beta(\tau_1,\infty) \leq (1-M\delta)^{\tau_1}$ and $\beta(\tau_1,\tau_2) \leq 2(1-M\delta)^{\tau_1}\ \forall\tau_2 \geq \tau_1$. Consider the following difference:

$$\begin{aligned} P_{ij}^{\tau} - P_{kj}^{\tau} &= \sum_u (P_{iu}-P_{ku})P_{uj}^{\tau-1} \\ &= \sum_{u:P_{iu}\geq P_{ku}} (P_{iu}-P_{ku})P_{uj}^{\tau-1} + \sum_{u:P_{iu}<P_{ku}} (P_{iu}-P_{ku})P_{uj}^{\tau-1}. \end{aligned}$$

Let us denote $\min_u P_{uj}^{\tau}$ by m_j^{τ} and $\max_u P_{uj}^{\tau}$ by M_j^{τ}. We now bound the above difference using m_j^{τ} and M_j^{τ}, we have

$$P_{ij}^{\tau} - P_{kj}^{\tau} \leq \sum_{u:P_{iu}\geq P_{ku}} (P_{iu}-P_{ku})M_j^{\tau-1} + \sum_{u:P_{iu}<P_{ku}} (P_{iu}-P_{ku})m_j^{\tau-1}.$$

By noticing that $\sum_{u:P_{iu}<P_{ku}}(P_{iu}-P_{ku})+\sum_{u:P_{iu}\geq P_{ku}}(P_{iu}-P_{ku})=0$, we have

$$\begin{aligned} P_{ij}^{\tau} - P_{kj}^{\tau} &\leq \sum_{u:P_{iu}\geq P_{ku}} (P_{iu}-P_{ku})\big(M_j^{\tau-1}-m_j^{\tau-1}\big) \\ &= \big(M_j^{\tau-1}-m_j^{\tau-1}\big)\Big(\sum_{u:P_{iu}\geq P_{ku}} P_{iu} - \sum_{u:P_{iu}\geq P_{ku}} P_{ku}\Big) \\ &= \big(M_j^{\tau-1}-m_j^{\tau-1}\big)\Big(1-\sum_{u:P_{iu}<P_{ku}} P_{iu} - \sum_{u:P_{iu}\geq P_{ku}} P_{ku}\Big) \\ &\leq (1-M\delta)\big(M_j^{\tau-1}-m_j^{\tau-1}\big), \end{aligned}$$

where the last inequality follows from the definition of δ.
Using the definition of M_j^{τ} and m_j^{τ}, we have that

$$\begin{aligned} M_j^{\tau} - m_j^{\tau} &\leq (1-M\delta)\big(M_j^{\tau-1}-m_j^{\tau-1}\big) \\ &\leq (1-M\delta)^{\tau}. \end{aligned}$$

Using the fact that m_j^{τ} monotonically increases with τ, M_j^{τ} monotonically decreases with τ, and both have a common limit π_j, we have

$$|P_{ij}^{\tau} - \pi_j| \leq (1-M\delta)^{\tau}. \tag{7}$$

Consider the following difference:

$$\begin{aligned} |P_{ij}^{\tau_2} - P_{kj}^{\tau_1}| &= |P_{ij}^{\tau_2} - \pi_j + \pi_j - P_{kj}^{\tau_1}| \\ &\leq |P_{ij}^{\tau_2} - \pi_j| + |\pi_j - P_{kj}^{\tau_1}|. \end{aligned}$$

Using (7) in the above inequality, we have the desired corollary. □