The Dissertation Committee for Akula Aneesh Reddy
certifies that this is the approved version of the following dissertation:

On Distributed Scheduling for Wireless Networks with Time-varying Channels

Committee:

Sanjay Shakkottai, Supervisor

Constantine Caramanis

Gustavo de Veciana

John Hasenbein

Sujay Sanghavi

On Distributed Scheduling for Wireless Networks with Time-varying Channels

by

Akula Aneesh Reddy, B. Tech.; M. S. E.

DISSERTATION

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT AUSTIN

May 2013

Dedicated to my mother Pandurangamma.

Acknowledgments

I have been extremely fortunate to have Prof. Sanjay Shakkottai as my mentor. This thesis would not have been possible with out his guidance at every stage. I would like to thank Sanjay especially for his patience in helping me formulate new research problems, for providing new insights, for discussing proofs and even helping me out with writing better papers. His enthusiasm for research and activeness have always been a great source of my inspiration. I am very grateful to Sanjay for taking time to help me out with some of my personal problems.

I would like to thank Prof. Sujay Sanghavi for asking good questions on my research problems, that made me go think twice about the problem at hand. It has been a great privilege to have worked with Prof. Sujay on a couple of problems that formed a part of my thesis work. I am thankful to Prof. Constantine Caramanis for both being a reader for my Master's thesis in the past and for currently being on my PhD thesis committee. I have had taken two courses with Constantine Caramanis and they have been extremely useful for the completion of this thesis. I would like to thank Prof. Gustavo De Veciana for being on my committee and for being a great teacher I have had the opportunity to take a class with. I would also like to thank Prof. John Hasenbein for being on my committee.

Special token of thanks to Prof. Lei Ying, with whom I have had a very fruitful collaboration and for providing me with valuable feedback at the initial stages of this thesis work. I would like to express my gratitude to my lab mates at WNCG - Jubin Jose, Rajiv Soundararajan, Shreeshankar Bodas, Siddhartha Banerjee, Aditya Gopalan, Rahul Vaze, Sriram Sridharan, Zrinka Puljiz, Abhik Das, Kumar Appaiah, Sharayu Moharir, Marcel Nassar, Salam Akoum and many others, with whom I have had countless discussions about research and other topics of interest. I have enjoyed all the time that we have spent solving mathematical puzzles and cross words during coffee breaks at JP's Java.

I would like to acknowledge my friends - Jubin Jose, Praveen Ramanathan, Hari Nair, Sri Ramya Duddukuri, Dharmendar Reddy, Rajesh Paleti, Aji, Reeja Jayan, Ashiwini Gopal, Vamsi Nadimpalli, Adarsh Mothukuri and many others who have made my stay at Austin a memorable place to remember. Thanks to Praneeth Netrapalli for pushing me to hit the gym couple of times a week, and thanks to Srinadh for having introduced me to the sport of bouldering and rock climbing. I am grateful to my undergraduate friends - Naveen, Ganesh, Gowtham, Deepti, Raghava, Arvind Ajoy, Dilip, Kalyan, Anirudh and many others, for being a part of my life and for always being there when required.

Last but not the least, I would like express my deepest gratitude to my family members for the support and encouragement they have provided me throughout my life. I will forever be indebted to them.

On Distributed Scheduling for Wireless Networks with Time-varying Channels

Publication No. ____________________

Akula Aneesh Reddy, Ph.D.
The University of Texas at Austin, 2013

Supervisor: Sanjay Shakkottai

Wireless scheduling is a fundamental problem in wireless networks that involves scheduling transmissions of multiple users in order to support data flows with as high rates as possible. This problem was first addressed by Tassuilas and Ephremides, resulting in the celebrated Back-Pressure network scheduling algorithm. 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.

However, the Back-Pressure (BP) algorithm suffers from various drawbacks - (a) 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, (b) the algorithm requires solving a global optimization

problem at each time to determine the schedule, making it highly centralized. Further, Back-pressure algorithm was originally designed for wireless networks where interference is modeled using protocol interference model. As recent break-throughs in full-duplex communications and interference cancelation techniques provide greatly increased capacity and scheduling flexibility, it is not clear how BP algorithm can be modified to improve the data rates and reduce the delay.

In this thesis, we address the drawbacks of Back-Pressure algorithm to some extent. In particular, our first work provides a new scheduling algorithm (similar to BP) that allows users to make individual decisions (distributed) based on heterogeneously delayed network state information (NSI). Regarding the complexity issue, in our second work, we analyze the performance of the greedy version of BP algorithm, known as Greedy Maximal Scheduling (GMS) and understand the effect of channel variations on the performance of GMS. In particular, we characterize the efficiency ratio of GMS in wireless networks with fading. In our third and fourth work, we propose and analyze new scheduling algorithms that can benefit from new advancements in interference cancelation techniques.

Table of Contents

List of Tables

List of Figures

Chapter 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 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 first by Tassiulas and Ephremides, and they a propose a online scheduling algorithm, Back-Pressure, that makes scheduling decisions based on instantaneous network state information (NSI), i.e., queue and channel state knowledge across the entire network. The authors show that the proposed algorithm can stabilize the queues as long as the arrivals are inside the throughput region.

Though Back-Pressure scheduling guarantees the best possible throughput performance for flows in networks, it suffers from few major 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 and (c) it is not delay optimal and (d) is only limited to protocol interference models.

Towards addressing these limitations, researchers have developed distributed implementations of the Max-weight algorithm [14–16, 36, 40, 50, 56], which use local NSI to achieve optimal/near-optimal 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]. There have been studies on scheduling with hop-delay optimality [61, 70] and the authors propose modified versions of BP to ensure hop-delay optimality along with throughput optimality.

In this thesis, we mainly focus on the throughput metric and are interested in developing new scheduling algorithms and analyzing existing algorithms that have low-complexity (so that they can be implementable in real time) and use either delayed or local NSI to make scheduling decisions. Further, we would investigate the problem of scheduling when wireless networks can leverage newly developed interference cancelation techniques in the literature.

1.1 Main Contributions

The main contributions of this thesis are as follows,

1. We address the problem of distributed scheduling in wireless networks with Markovian channels and heterogeneously delayed NSI. We propose a threshold-type distributed scheduling algorithm that is provably throughput-optimal. We further characterize the effect of delayed NSI on the network throughput region.

2. We analyze the performance of a well-known low-complexity algorithm, Greedy-Maximal Scheduling (GMS), to the case of general wireless networks with fading structure. We define Fading-Local pooling factor for graphs with fading and showed that it characterizes the fraction of throughput that can be achieved by GMS. We further illustrate using examples that fading can either help or hurt the performance of GMS.

3. We analyze the performance of greedy version of Shortest-Path aided BP algorithm (SPBP) [61] both in terms of achievable throughput and average hop-delay. We further show that the greedy SPBP can achieve hop-delay optimality in few wireless networks where cut-through switching (CTS) is feasible.

4. We address the problem of scheduling in wireless networks with nodes that can implement advanced interference mitigation techniques. In particular, we propose new queue-structures and algorithms that can extract

the benefits of ergodic interference alignment (IA). We further provide low-complexity algorithms and characterize the loss in throughput.

The below table summarizes our contributions to the current literature.

Algorithm	*Drawbacks*	*Contribution*
Back-Pressure (BP)	Instantaneous Global NSI	Algorithmic : Heterogeneous delayed NSI Analytical : Thruput loss with delayed NSI
	Computational Complexity	Analytical : Performance of GMS in fading Numerical : Fading can help or hurt GMS
	Explores all the paths	Analytical : Performance of Greedy SPBP Applications : Cut-through switching
	Restricted to few interference models	Algorithmic : Modified BP to use Ergodic IA Analytical : Loss with sub optimal algorithms

Table 1.1: Our contributions to the current literature

1.2 Organization of the thesis

In chapter 2, we present our results on distributed scheduling in wireless networks, where only heterogeneously delayed NSI is available. In chapter 3, we present our results on the performance of the popular low-complexity distributed GMS algorithm in the presence of fading. We analyze the performance of greedy version of Shortest-Path aided BP (SPBP) algorithm for multi-hop networks and identify networks where it performs optimally in chapter 4. In chapter 5, we extend the BP algorithm for wireless networks with advanced interference cancelation techniques (in particular, ergodic interference alignment technique). We provide future directions and conclusions in chapter 6.

Chapter 2

Distributed scheduling in wireless networks with heterogeneous delayed NSI

2.1 Introduction

An important problem which arises while scheduling in the presence of channel fading and network interference, and which remains unexplored in the literature, 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 work 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.

2.1.1 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 re-

gion 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.

2.2 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.

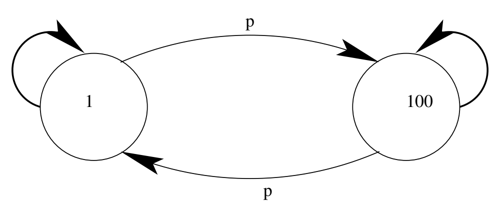

Figure 2.1: User C channel Markov chain

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. 2.1). 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. 2.2 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.

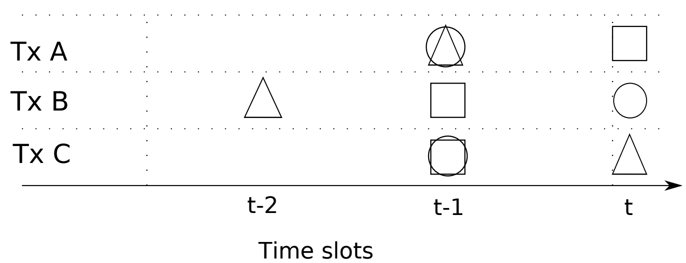

Figure 2.2: Heterogeneous NSI for the 3-user network: Squares, circles and triangles represent the *most recent* channel state information available to user A, B and C respectively.

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 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?

2.3 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 work.

- **Network Model:** We consider a wireless network consisting of L transmitter-receiver pairs denoted by 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 $C = \{c_1, c_2, c_3, ..., c_M\}$, 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.

[1] This assumption is to ensure that the system state Markov chain (defined in Section 2.3.2) is irreducible and aperiodic, by suitably augmenting the state space.

- **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 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 work 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]

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 $\vec{\tau}_l$ that characterizes the available delayed

[2] These assumptions are to ensure that the system state Markov chain (defined in Section 2.3.2) is irreducible and aperiodic, by suitably augmenting the state space.

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.,

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

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 $\{C_l[t]\}_{l \in L}$ by $C[t]$. We denote the information available at transmitter l by $\{P_l(C[t](0:\tau_{max})), P_l(Q[t](0:\tau_{max}))\}$, where

$$P_l(C[t](0:\tau_{max})) := \{\vec{P}_{lm}(C[t](0:\tau_{max}))\}_{m \in L}, \text{ with}$$

$$\vec{P}_{lm}(C[t](0:\tau_{max})) := \{C_m[t-\tau]\}_{\tau=\tau_{max}}^{\tau_l(m)},$$

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

2.3.2 Performance Objective: Throughput/Stability

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

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

2.4 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, 20, 27]. 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 2.3, 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 2.3 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.

Given this distinguishing feature of static-split scheduling in our setting, we show in Section 2.4.1 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 2.4.2, we further simplify the structure of the static scheduling policies, by identifying

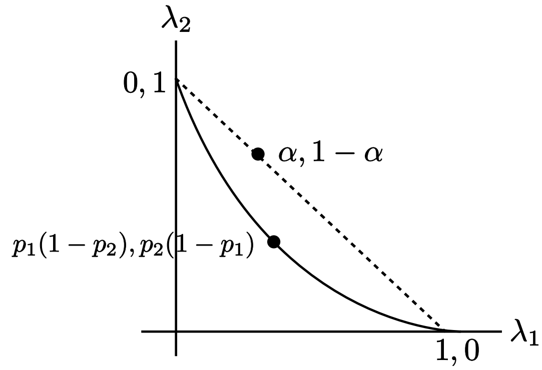

Figure 2.3: Static-split scheduling rules, in our setting, are not necessarily throughput-optimal in the conventional sense. For two transmitters sharing a unit-rate collision channel, the corner points $(1, 0)$ and $(0, 1)$ are achieved by individual static-split scheduling at each transmitter. However, 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.

the "critical set" of available delayed NSI sufficient for each transmitter to achieve any valid rate point.

Finally, in Section 2.4.3, 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.

2.4.1 Throughput Characterization

Towards describing the throughput region, i.e., the set of all supportable arrival rate vectors $\{\lambda_l\}_{l\in L}$, consider a collection of functions $\{f_l\}$, one for each link/transmitter $l \in L$, where each $f_l : P_l(C[t](0:\tau_{max})) \to \{0,1\}$. These maps $\{f_l\}_{l\in L}$ 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 $f_l(P_l(C[t](0:\tau_{max})))$ and attempts to transmit (i.e., schedule itself) whenever this binary value is 1.

If the delayed channel state information at time t is $C[t-\tau_{max}] = c$, 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 $S(c,f) = \{S_l(c,f)\}_{l\in L}$, as follows:

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

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

$$\eta(c) = CH_f\big(S(c, f)\big).$$

Thus, $\eta(c) \subset \mathbf{R}^L$ 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

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

In other words, Λ is the Minkowski sum of the sets $\{\eta(c)\}_{c \in C^L}$ weighted by the respective probabilities $\pi(c)$. 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 2.4.1. *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 2.4.1 (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 2.4.3 to design a throughput-optimal scheduling policy. We refer the reader to the appendix for the detailed proof of the lemma.

2.4.2 Critical NSI

As defined in the system model (Section 2.3), $\tau_l(h)$ 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., $\{C_k[t-\tau_k(l)] : k \in L, k \neq l\}$) 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 $C[t](0 : \tau_{max})$, the critical set of information related to link l is defined as the the channel state information at times $\{t-\tau_k(l)\}_{k\in L:k\neq l}$. Let us denote the critical NSI of the network at time t as $CS(.)$, which can be expressed mathematically as follows

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

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

$$CS_l(C[t](0:\tau_{max})) := CS(C[t](0:\tau_{max})) \bigcap P_l(C[t](0:\tau_{max})).$$

Recalling the example in Section 4.2, 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 work, 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.

2.4.3 A Threshold-based Throughput-optimal Scheduling Algorithm:

The two ideas discussed so far – (a) that global, common randomness helps span the stability region (Section 2.4.1), and (b) that critical delayed NSI at each transmitter is as good as all available delayed NSI (Section 2.4.2), 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:

$$\arg\max_{T} \sum_{l \in L} Q_l(t - \tau_{max}) R_{l,\tau_{max}}(T), \tag{2.1}$$

  where

$$R_{l,\tau}(T) := E\big[C_l[t] 1_{C_l[t] \geq T_l(.)} \big(\gamma_l + (1-\gamma_l) \prod_{m \in I_l} 1_{C_m[t] < T_m(.)}\big) | C[t-\tau]\big], \tag{2.2}$$

  and $T_l(.) := T_l(CS_l(C[t](0 : \tau_{max})))$.

- **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

$$C_l[t] \geq T_l(CS_l(C[t](0 : \tau_{max}))).$$

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 2.4.2. *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 2.4.3. *Consider the optimization problem*

$$\arg\max_{F(.)} \sum_{l \in L} Q_l(t - \tau_{max}) R_{l,\tau_{max}}(F(.)), \tag{2.3}$$

where

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

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

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

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

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

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

$$T_l^*(P_l(C[t](0:\tau_{max}))) = T_l^*(CS_l(C[t](0:\tau_{max}))).$$

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. □

2.5 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 $\{\vec{\tau}_l\}_{l \in L}$ (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 2.5.1. *For integers* $\tau_1, \tau_2 \geq 0$*, let*

$$\alpha(\tau_1, \tau_2) := \frac{2Lk_o\beta(\tau_1, \tau_2)}{\sum_j c_j \min_i P_{ij}^{\tau_1}}, \tag{2.4}$$

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 2.5.1 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 2.4.2) that is throughput-optimal. We will need the following useful lemma [7].

Lemma 2.5.2. *(Adapted from [7]) At any time t, given the common NSI $(Q[t](\tau_1 : t), C[t](\tau_1 : t))$, let T_1^* be the optimal set of thresholds calculated using the proposed algorithm and T_2 be set of thresholds computed using a scheduling policy S_ρ such that the following condition holds (for some $\rho \in [0,1]$):*

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

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

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

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

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

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

Let P_{ij}^{τ} 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_{ij}^{τ}, we have

$$\begin{aligned} R_{l,\tau_1}(T_2^*) =& \gamma_l \left(\sum_{i=1}^{M} c_i P_{.i}^{\tau_1} 1_{c_i \geq T_{2,l}^*}\right) + \\ & (1-\gamma_l)\left(\sum_{i=1}^{M} c_i P_{.i}^{\tau_1} 1_{c_i\geq T_{2,l}^*}\right) \prod_{m\in I_l} \left(\sum_{i=1}^{M} P_{.i}^{\tau_1} 1_{c_m \geq T_{2,l}^*}\right). \end{aligned} \tag{2.5}$$

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

Lemma 2.5.3. $|R_{l,\tau_1}(T_2^*) - R_{l,\tau_2}(T_2^*)| < k_o\beta(\tau_1,\tau_2)$.

Using Lemma 2.5.3, we have that

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

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

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

Employing Lemma 2.5.3 once again, we have

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

Note that

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

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

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

Using Lemma 2.5.2 now yields $\Lambda_{\tau_2} \supseteq (1-\alpha(\tau_1,\tau_2))\Lambda_{\tau_1}$ as desired. □

Finally, as a corollary of Theorem 2.5.1, 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 2.5.4.

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

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

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. □

2.6 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 2.5.

2.6.1 Methodology

For our simulations, we consider a wireless network with $L = 10$ links. Complete interference is assumed with perfect collisions, i.e., $I_l = \mathcal{L} \ \{l\}$ 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 2.4.3 is implemented in each time slot by solving the optimization (2.1) as a brute-force search over all possible thresholds $\mathbf{T}$.

2.6.2 Throughput Performance with Delayed NSI

We simulate in Matlab, the proposed algorithm (Section 2.4.3) 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 2.4 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 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

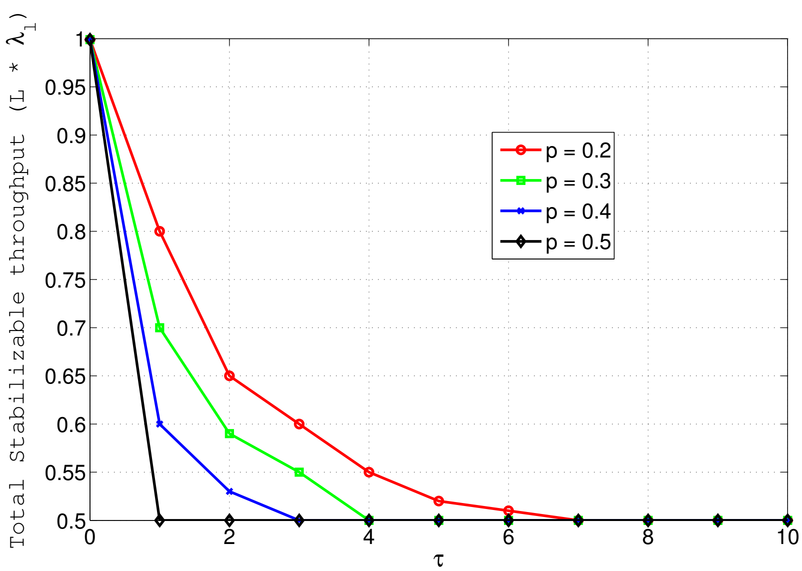

Figure 2.4: Sum-throughput performance of the proposed algorithm for a 10-node wireless network with full collision interference. Traffic is symmetric with rate $\lambda_l = \lambda$, and channels are 2-state Markov with rates $\{0, 1\}$ packets/slot. Each curve depicts optimal sum-rate achievable for various homogeneous NSI delays $\tau = 0, 1, \dots, 10$, for a different value of channel state crossover probability p.

current channel state decreases with p increasing to 0.5.

2.6.3 Throughput Region Mixing-based Bounds

We next turn to evaluating the efficacy of our bound $\alpha(\cdot,\cdot)$ from Theorem 2.5.1. Note that, from Section 2.5, 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 2.5 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 2.5 shows that the bounds of Theorem 2.5.1 are indicative of the level of NSI delay beyond which there is effectively little degradation of the system throughput. From Figure 2.5, when $p = 0.4$ observe that the bound is 1 for all $\tau > 5$. At the same time, from the simulation results of Figure 2.4, we notice that for $\tau \geq 3$ there is no further degradation in throughput. Thus, the bound derived in Theorem 2.5.1 provides an estimate of the NSI delay beyond which there is no further degradation in throughput.

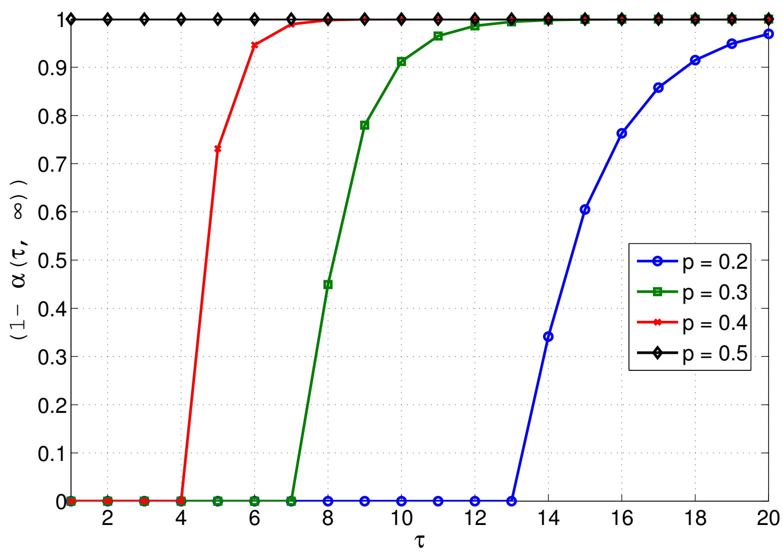

Figure 2.5: Plot of $(1 - \alpha(\tau, \infty))$ versus homogeneous NSI delay τ. Each curve represents a different value of channel state crossover probability p. The quantity $(1-\alpha(\tau, \infty))$ lower-bounds the factor by which the throughput region with "infinitely delayed NSI" (i.e., NSI with 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.

2.7 Discussion: Implementation Complexity

We remark that the throughput-optimal algorithm developed in Section 2.4.3 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 2.4.2). 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 2.4.3). 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 2.5). The technical challenge here is in characterizing the loss in throughput, and we develop novel Markov chain mixing-based techniques to do so.

2.8 Conclusion

In this chapter, 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.

Chapter 3

On the effect of channel fading on greedy scheduling

3.1 Introduction

In this work, we analytically investigate the effect of fading on the throughput performance of a natural and popular scheduling algorithm: Greedy Maximal Scheduling (GMS) [33, 51, 54, 67]. As with any scheduling algorithm, GMS is a way to determine which wireless links can transmit at any given time, based on their mutual interference characteristics and their current level of fading. In particular, GMS involves first associating a weight with each link – which depends on the load of the link and its channel condition. Then, GMS involves iteratively turning on the heaviest link that does not interfere with links already turned on. This is repeated every time slot.

GMS has empirically shown to have very good throughput and delay performance; recent theoretical advances [31, 37, 39, 51, 52, 62] characterize its throughput. All of these works assume that there is no fading; ie that the rate a link can support is invariant as long as all the links that interfere with it are not simultaneously on. Our work investigates what happens to this performance in the more realistic setting with intrinsic channel fading as well.

In particular, we compare the relative throughput of GMS as compared to that of an optimal scheduler.

Our results demonstrate that the effect of fading is quite subtle; in particular, in some instances fading can degrade the relative performance of GMS, while in other cases it can improve it. The former reflects the fact that fading provides an extra degree of freedom and complexity in the system, which GMS may not be able to handle as well as in a system without this fading. The latter reflects the, perhaps more subtle, fact that the sub-optimality of GMS (even without fading) is tied to the existence of special global system configurations that result in poor performance. The presence of fading "breaks up" these global configurations – not allowing them to occur too often – allowing GMS to perform relatively better.

Specifically, our contributions are as follows: For a given wireless network with fading channels,

1. We define a new quantity, called Fading-Local Pooling Factor (F-LPF), analogous to LPF defined in [51] that characterizes the performance of Greedy Maximal Scheduling (GMS) in wireless networks with fading channels. Furthermore, we show that Fading-LPF is a lower bound on the fraction of throughput that can be stabilizable by the GMS when the arrivals and channels are independent and identically distributed over time.

2. With arbitrary arrival and channel state process, we show that Fading-

LPF is an upper bound on the fraction of throughput that can be stabilizable by the greedy schedule. More specifically, we construct an adversarial arrival and channel process with long term averages that lie outside the scaled throughput region and show that GMS policy cannot stabilize the queues.

3. We further provide lower and upper bounds on Fading-LPF that are easy to evaluate. We provide two example networks with specific fading structure and use the derived bounds to demonstrate that fading can either enhance or degrade the relative performance of GMS as compared to the non-fading scenario.

4. With fading, we can represent the channel model as a collection of global channel-states, where each state is associated with an independent set and an occurance probability. A natural question that arises is the following: Is the acheivable rate-region with fading simply the (channel-probability weighted) average of the per-state *scaled* rate regions, with the scaling parameter simply being the conventional LPF for each state? We show that this is in general not true. However, we derive a region that *can* be stabilized by the GMS in wireless networks with fading channels. This region is characterized based on the interference degree of the subgraphs (generated from original network) and the fading distribution.

3.1.1 Related Work:

Transmission scheduling has been a key challenge in modern wireless systems. The MaxWeight algorithm, proposed in [42], has been the inspiration for many approaches to address this in various wireless systems (see [67] for several variants). However, this algorithm suffers from centralization as well as computational complexity.

Thus, there has been significant research in finding sub-optimal (i.e., achieving a subset of the throughput region) distributed scheduling algorithms with low complexity. The authors in [54] propose one such policy called Greedy Maximal Scheduling, whose time complexity is linear in the number of links, and has a distributed implementation [52]. There are other sub-optimal, randomized algorithms that have been proposed with similar performance as GMS [36, 40].

The authors in [33] have been the first to study the performance of GMS under a general interference model. They have identified conditions (so called 'Local Pooling') under which there is no loss in the network throughput region with GMS. The notion of Local Pooling has been extended to a multi-hop regime by [62].

This condition being identified as too restrictive, the authors in [51] have defined a new quantity called Local Pooling Factor (LPF) that exactly characterizes the fraction of throughput region achieved by GMS, and show that over tree networks with a $K-$hop model for interference, GMS achieves the entire throughput region. Additional characterizations, including a per-link LPF [38] and bounds to characterize the stability region [39], have been proposed in literature.

The authors in [31] exactly characterize, using graph theoretic methods, the set of network graphs (with only the primary interference constraints) where GMS is optimal (LPF = 1). Finally, the authors in [37] have studied the performance of GMS with the SINR interference model, and have shown that GMS exhibits zero LPF in the worst case.

All the above results assume that there are no channel variations (fading). In this work, we study the effect of channel variation on the performance of GMS.

3.2 System Model and Back Ground

We consider a wireless network consisting of K links labeled as $\{1, 2, 3, ..., K\}$. Let K denote the set of links in the network. Each link l consists of a transmitter and receiver. We assume time to be slotted. Each time slot is composed of two parts. The first (control) part is reserved for making the transmission decision and second part for transmitting the packet. At time slot t, we denote the channel capacity of link by $C_l[t]$. We assume that the capacity varies from slot to slot, and is constant during a time slot. We consider collision interference/protocol model and denote the set of links that interfere with link l by I_l. We say that the transmission on link l at time t is successful, if no link in the I_l transmits during the same time t. The maximum number of packets that can be successfully transmitted in time slot t on link l is bounded by $C_l[t]$.

We assume single hop flows in the network. Let $A_l[t]$ denote the number of packets that arrive at transmitter of link l at time slot t. We assume that arrival processes is bounded and average rate of arrivals for link l is denoted by λ_l.

For simplicity we first consider ON/OFF channels (i.e $C_l[t] = 0 \,\text{or}\, 1$) and later show that our results can be extended to channels with finite number of channel states. For the ON/OFF setting, global state (GS) refers to specifying the set of links that are in 'ON' state. Let $GS(t)$ denote the set of links that are in 'ON' state in time slot t. Let $\pi(J)$ denote the fraction of time the network is in global channel state J, where links in set J are 'ON' and links in the set $K \backslash J$ are in 'OFF' state. Let $\pi := \{\pi(J), J \subset K\}$ denote the *fading structure*.

Assumptions. :

A1 (Long-term Averages): We assume that the long-term time averages of arrivals and channel states satisfy the following:

$$\frac{1}{T}\sum_{t=0}^{T} A_l[t] \to \lambda_l \quad \text{as} \quad T \to \infty. \tag{3.1}$$

and

$$\frac{1}{T}\sum_{t=0}^{T} 1_{GS(t)=J} \to \pi(J) \quad \text{as} \quad T \to \infty. \tag{3.2}$$

A2 (Randomness): We assume that arrivals are mutually independent i.i.d processes with $\lambda_l = E[A_l[t]]$. Similarly the channels are independent across time and form a stationary process with $\pi(J) = E[1_{GS(t)=J}]$.

While both assumptions A1 and A2 specify the same long-term averages, we note that assumptions in A1 allow for arrival and channel state processes to be *dependent across time and across links* in a deterministic, and possibly *adversarial manner*. The necessity for the above sets of assumptions will be clear as we state our main results in Section 3.3.

3.2.1 Preliminaries

As discussed earlier, there is a rich history of analysis of GMS algorithms for the non-fading case [31, 33, 37–39, 51]. In this section we build on this notation in literature to allow for time-varying (fading) channels.

We define Interference graph IG for a set of links as follows: Each link is represented by a node and an edge is drawn between two nodes if transmissions on the corresponding links in the original graph interfere with each other. This model captures many existing wireless models and is quite general. We define the Independent set on this graph as set of nodes with no edges between them. Let $Q_l[t]$ denote the number of packets present at the transmitter at time t waiting to get scheduled on link l. Let $S_l[t] \in \{0, 1\}$ denote the schedule decision for link l at time t. At each time t, a schedule $\vec{S}[t]$ is determined based on the global queue state and channel state information at time t, that is $(\vec{Q}[t]), \vec{C}[t])$. We also assume that arrivals occur at the end of time slot, thus we have the following queue dynamics:

$$Q_l[t+1] = (Q_l[t] - C_l[t]S_l[t])^+ + A_l[t], \qquad (3.3)$$

where $a^+ = \max(0, a)$.

Given the arrival traffic rate $\{\lambda_l\}_{l \in L}$ and a scheduling policy, we say that the network is *stable* under scheduling policy if the mean of the sum of queue lengths is bounded. We say that an arrival rate vector $\{\lambda_l\}_{l \in L}$ is *supportable* if there exists any scheduling policy that can make the network stable. We call the set of all arrival vectors that are supportable by *throughput region* and denote it as Λ_f, where f denotes that the channels are fading.

We say that a scheduling policy is throughput optimal if it can stabilize the network for all arrival rates inside the throughput region.

Definition 1: ([51]) The interference degree $d_I(l)$ of link l is the maximum number of links in the set $\{l \cup I_l\}$that can be active at the same time with out interfering with each other. The interference degree $d_I(G)$ of a graph $G = \{V, E\}$ is the maximum interference degree across all its links in E

Consider a wireless system with 4 links. Let $I_1 = \{2\}$, $I_2 = \{1, 3, 4\}$, $I_3 = \{2, 4\}$ and $I_4 = \{2, 3\}$. The interference graph is shown in the Figure 3.1 with the corresponding $d_I(l)$. The interference degree of this example graph is 2.

Definition 2: Given an interference graph, an independent set corresponds to set of nodes (links in the original graph) such that there is no edge between any two nodes in the set (no two links interfere in the original graph). Further, it is maximal if it is not a subset of any other independent set. For a set of links L, define a matrix M_L whose columns represent the maximal inde-

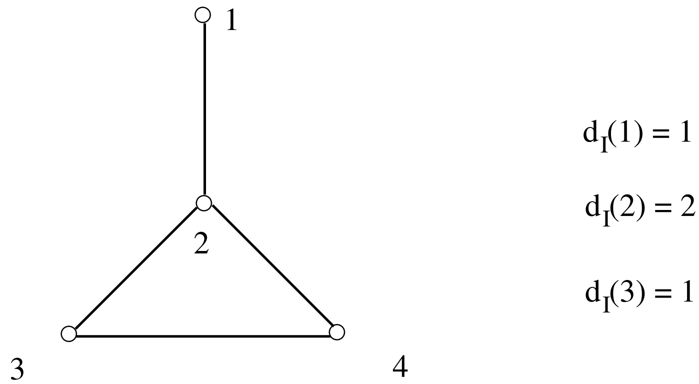

Figure 3.1: Interference Graph where nodes denote the links and edges denote the interference constraints.

pendent sets on the set L, with $|L|$ rows one for each link. We assume links are naturally ordered and rows in M_L are assigned according to the defined order. For $J \subset L$, let $M_{J,L}$ denote the matrix with $|L|$ rows and is constructed from M_J as follows: columns from M_J are used and zero row vectors are added for links which do not belong to set J. Let $CH(M_{J,L})$ denote the convex hull of all column vectors of matrix $M_{J,L}$.

For the above example with 4 links, let $J = \{1, 2, 3\}$ and $L = \{1, 2, 3, 4\}$, we have

$$M_J = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{pmatrix}$$

and

$$M_{J,L} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 0 \end{pmatrix}$$

Note that the set $\Lambda_L := \{\vec{\lambda} : \vec{\lambda} < \vec{\mu}; \vec{\mu} \in CH(M_L)\}$ characterizes the throughput region of set of L links if no fading were present. We now define the throughput region with the *fading structure*,

Definition 3: The throughput region Λ_f for a given network with fading pattern $\pi(J)$ is described as follows,

$$\Lambda_f = \Big\{\vec{\lambda} : \vec{\lambda} > 0, \vec{\lambda} \le \sum_J \pi(J)\vec{\eta}_J \, \text{where} \, \vec{\eta}_J \in CH(M_{J,K})\Big\}.$$

Definition 4: ([51]) The efficiency ratio γ^*_{pol} under a given scheduling policy is defined as follows,

$$\gamma^*_{pol} = \sup\Big\{\gamma : \text{the policy can stabilize for allthe arrival rate vectors} \, \lambda \in \gamma\Lambda_f\Big\}.$$

Definition 5: Given $x(J) \in [0, 1]$, we define a new region $\Lambda_f(\vec{x})$ as follows,

$$\Lambda_f(\vec{x}) = \Big\{\vec{\lambda} : \vec{\lambda} > 0, \vec{\lambda} \le \sum_J x(J)\pi(J)\vec{\eta}_J \, \text{where} \, \vec{\eta}_J \in CH(M_{J,K})\Big\}.$$

Note that throughput region is same as $\Lambda_f(1)$.

3.2.2 GMS Algorithm [54]

We now describe the Greedy Maximal Scheduling(GMS) Algorithm. GMS essentially finds a maximal schedule in a greedy fashion. Each node in the interference graph is assigned weight equal to $f(Q_l(t)C_l(t))$, where $f(.)$ is a strictly increasing function that is zero at 0 and tends to infinity as $Q_l(t)C_l(t) \to \infty$. It then proceeds as follows: it finds the node with maximum weight in the whole network and adds it to GMS schedule (ties are broken arbitrarily), it further discards all the neighboring nodes along with the selected node and repeats the above procedure on the reduced graph, till there are no more nodes left in the interference graph.

3.3 Main Results

In this work, we characterize the performance of GMS algorithm for wireless networks with time-varying channels. We define the fading local pooling factor, $\sigma_L^*(\pi)$, for a set of links $L(\subseteq K)$ with fading structure π as follows:

$$\sigma_L^*(\pi) = \inf\{\sigma : \exists\, \vec{\phi}_1, \vec{\phi}_2 \in \Phi(L) \text{ such that } \sigma\vec{\phi}_1 \geq \vec{\phi}_2\}, \tag{3.4}$$

where,

$$\Phi(L) = \{\vec{\phi} : \vec{\phi} = \sum_{J:J\subseteq K} \pi(J)\vec{\eta}_J \text{ where } \vec{\eta}_J \in CH(M_{J\cap L,L})\}, \tag{3.5}$$

and *Fading-Local Pooling Factor (F-LPF)* for a network G, $\sigma_G^*(\pi)$, with fading structure π as follows:

$$\sigma_G^*(\pi) = \min_{L:L\subseteq K}\sigma_L^*(\pi), \tag{3.6}$$

Note that the above definition reduces to the known definition of LPF for a graph [51] when there is no fading, i.e, when $\pi(K) = 1$.

The F-LPF can be understood as follows: Consider arrivals only to links of set L (assume arrivals to other links are 0); when the links in set J are 'ON' (others are 'OFF'), GMS will pick a maximal schedule among the 'ON' links, i.e. a column of $M_{J\cap L,L}$. Thus vector $\vec{\eta}_J$ is the long run average of these maximal schedules when system is in state J; so $\vec{\eta}_J \in CH(M_{J\cap L,L})$. Thus $\Phi(L)$ is the set of all long-run average service vectors that could appear due to GMS when the arrivals are restricted only to set of links in L. For any two vectors $\vec{\phi}_1, \vec{\phi}_2 \in \Phi(L)$, it may thus happen that GMS results in $\vec{\phi}_2$ service vector, when it should have been $\vec{\phi}_1$ (for the optimal case). Thus $\sigma^*_L(\pi)$ is the worst possible ratio difference among all the possible service vectors of $\Phi(L)$.

Dual Characterization and Implications: In the same spirit as [33, 38], the *Fading*- Local Pooling Factor has a dual characterization, as noted in Lemma B.3.1, and displayed below. The F-LPF, $\sigma^*_L(\pi)$, is given by the solution to the following optimization problem:

$$\sigma^*_L(\pi) = \max_{x,a(J),b(J)} \sum_{J:J\subseteq L} \pi_L(J)a(J) \tag{3.7}$$

$$\text{s.t} : x'M_{J,L} \geq a(J)e' \quad \forall J \subseteq L$$

$$x'M_{J,L} \leq b(J)e' \quad \forall J \subseteq L$$

$$\sum_{J:J\subseteq L} \pi_L(J)b(J) = 1, \tag{3.8}$$

where e is a column vector of all ones, $(\cdot)'$ is the vector transposition operation and π_L denotes the marginal distribution on set of links L induced by π.

Observe that each fading state J *induces* a network defined by ON edges (i.e., all OFF links are removed from the network). Thus, one could ask if with fading channels, the F-LPF can be determined simply by computing the "standard" LPF (denoted by $\sigma^*(J)$) for each of these induced networks, and then averaging these quantities (weighted by the steady-state fractions of times for each of the fading states) over all possible fading states? In other words, *is the following true*?

$$\sigma_L^*(\pi) \stackrel{?}{=} \sum_{J:J\subseteq L} \pi_L(J)\sigma^*(J)$$

where $\sigma^*(J)$ is the standard LPF [51] for the network that is induced by state J.

An important insight that emerges from the dual characterization is that such **averaging does necessarily not hold**, in particular because the possibly adversarial nature of the fading channel does not permit averaging. Note that the adversary *cannot* change the long-term fractions of the global states – it can merely change the temporal correlations. Inspite of this, averaging does not hold, as clearly shown in Example B in Section 3.3.2).

In a tree network with fading as in Example B (see Section 3.3.2), while the LPF for each state is '1', the F-LPF is less than 4/5 which is lower than *any* convex averaging of the states! This discussion implies that the regular LPF does not immediately extend to the case with fading. This motivates us to explicitly develop the local pooling factor in the presence fading, and understand its implications.

Contributions:

3.3.1 Characterization in terms of F-LPF:

Our first contribution, Theorem 3.3.1, characterizes the efficiency ratio of GMS algorithm in the presence of fading.

Theorem 3.3.1. *a) (Upper Bound) Under a given network topology and channel state distribution with Assumption A1 on the arrivals and fading channels, the efficiency ratio of GMS (γ^*) is less than or equal to $\sigma^*_G(\pi)$.*

b) (Achievability) Under a given network topology and channel state distribution π with Assumption A2 on the arrivals and fading channels, the efficiency ratio of GMS (γ^*) is greater than or equal to $\sigma^*_G(\pi)$.

Implications: The above result enables us to understand the performance of GMS compared to the optimal scheduler in the presence of fading. In particular, computing bounds on $\sigma^*_G(\pi)$ leads to insights on the positive and negative aspects of fading (discussed further in Theorems 3.3.2 and 3.3.3). Observe first that as long as the long-term averages on the arrivals and channels are satisfied (Assumption A1), we can construct an arrival and channel process that ensures that the efficiency *cannot* exceed the F-LPF $\sigma^*_G(\pi)$. Further, for *typical* arrival and channel processes with sufficient randomness (in this work i.i.d. assumptions have been imposed, however this can be weakened), the converse holds wherein $\sigma^*_G(\pi)$ is achievable.

Proof Discussion: For the first part, we extend the ideas in [51], to construct an adversarial arrival and *fading* process pattern when arrival rates are outside the $(\sigma^*_G(\pi)+\epsilon)\Lambda_f$ and show that a set of queues are unstable under GMS policy. For the second part, we use the approach in [33, 51] as follows: we show that if $\vec{\lambda}$ is inside $(\sigma^*_G(\pi)-\epsilon)\Lambda_f$ then GMS policy can stabilize all the queues in the network. We look at the deterministic fluid limit of the system and exhibit a Lyapunov function whose drift is negative under the GMS policy. We have that fluid model is stable and therefore that the original system is stable. Please refere to appendix for full details.

Theorem 3.3.2 (Upper Bound)**.** *For every $J \subseteq K$ and any $(\vec{\mu}_J, \vec{\nu}_J, H_J)$ such that $\vec{\mu}_J, \vec{\nu}_J \in CH(M_J)$, $\vec{\nu}_J \leq H_J \vec{\mu}_J$, we have that*

$$\sigma^*_G(\pi) \leq max_l \frac{\sum_{J \subseteq K} \pi(J) H_J \mu_J(l)}{\sum_{J \subseteq K} \pi(J) \mu_J(l)},$$

where $\mu_J(l) = 0$ if $l \;\; J$.

Implications: While $\sigma^*_G(\pi)$ is defined only though an optimization problem, the upper bound permits an explicit solution. This bound is useful, as evidenced in Example B provided in Section 3.3.2. In particular this upper bound is useful to illustrate that the F-LPF is not a simple convex combination of the standard LPF averaged over the fading states, and that adversarial fading can indeed worsen the performance of GMS.

Proof Discussion: Though the proof follows from straightforward algebraic computations, the value of the theorem lies in the smart selection of $(\vec{\mu}_J, \vec{\nu}_J, H_J)$ vectors that satisfy the inequality stated in the above theorem. In the worst case the bound yields 1; however we can use the existing results in literature [31] to get good bounds. Thus, the tightness of the upper bound depend up on the ability to identify good vectors that satisfy the above constraints.

Theorem 3.3.3 (Lower Bound)**.**

$$\sigma_L^*(\pi) \geq \frac{\sum_{J\subseteq L} \pi_L(J) n(M_J)}{\sum_{J\subseteq L} \pi_L(J) N(M_J)}, \tag{3.9}$$

where $n(M) = \min_j \sum_i M_{ij}, N(M) = \max_j \sum_i M_{ij}$. π_L *denotes the marginal distribution on set of links* L *induced by* π *and can be computed as follows,*

$$\pi_L(J) = \sum_{I:I\subseteq K, I\cap L=J} \pi(I)$$

.

Implications: The ability to compute a lower bound leads to the interesting observation that fading can help *improve* efficiency. This is because, by turning links 'OFF', fading "breaks up" some of the bad global states that can lead to poor GMS performance. This is explicitly brought out in Example A in the context of a six-link network.

Proof Discussion: The lower bound is derived using the dual formulation of the F-LPF, see (3.7). We find a point in the dual search space that satisfies all the constraints in the dual characterization, thus yielding a lower bound on the primal problem. Observe that $n(M_J)$ corresponds to the minimum number of links that needs to be 'ON' in any maximal schedule on set of J links and $N(M_J)$ denotes the maximum number of links that could be 'ON' among all the maximal schedules on set of J links. Thus, the lower bound can be computed easily and can be shown to be tight for some wireless networks. As an interesting aside, note that the lower bound provided is always better than the inverse of the interference degree of graph G (see Corollary 1).

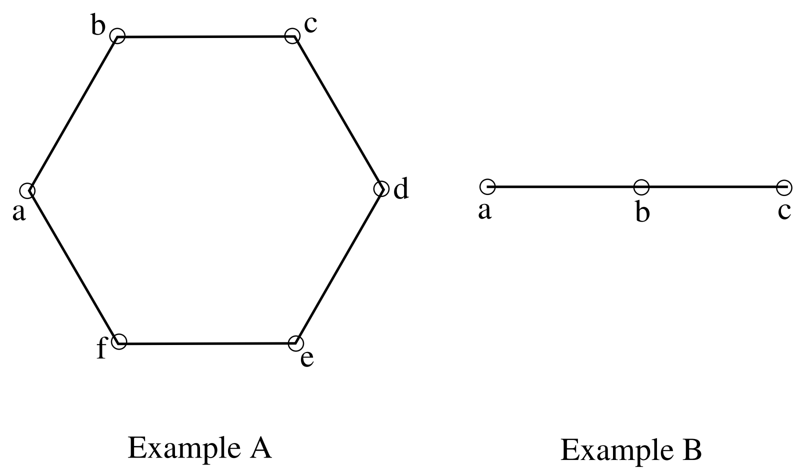

Figure 3.2: Interference graphs for the two examples: Hexagon network and Line network

We now present two examples: A and B, one in which fading reduces the relative performance of GMS and the other in which fading enhances the relative performance of GMS respectively to illustrate the value of the above results.

3.3.2 Examples: Benefit and Detriment with Fading

Example A: A network where fading structure improves the relative performance of GMS: Consider a graph with six links $K = \{a, b, c, d, e, f\}$. The interference graph for the six links is shown in the Figure 3.2. Each link is either is state 'ON' or 'OFF'. We consider the following fading structure, π, for $J \subseteq K$

$$\pi(J) = p^{|J|}(1-p)^{6-|J|},$$

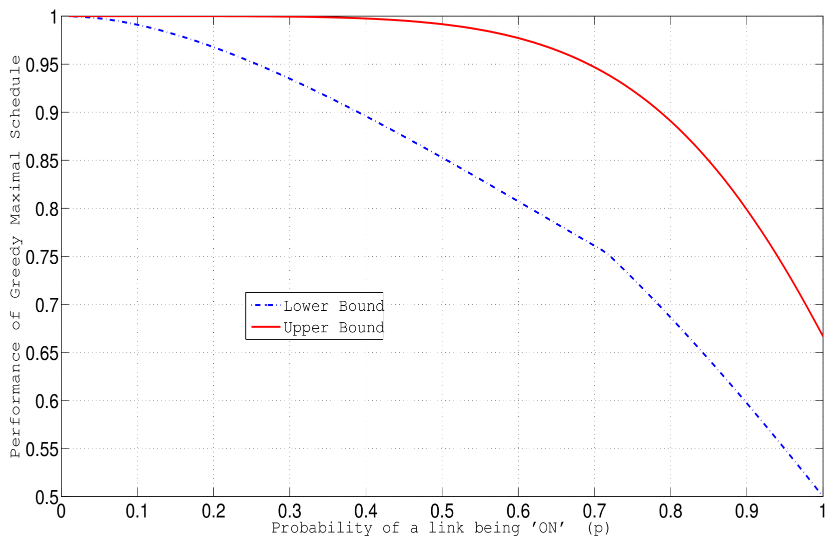

Figure 3.3: Bounds on the fading local pooling factor for the Hexagon network

where $|J|$ denotes the size of set J. Note that $p = 1$ corresponds to the no-fading case.

Using our results, we compute the lower bound and upper bounds on local pooling factor $\sigma_G^*(\pi)$ and is plotted in Figure 3.3.

It is known [51] that the non-fading LPF for the above example is equal to 2/3. From the graph, we observe that for smaller values of p, F-LPF for above hexagon network with fading is greater than LPF with out fading structure. As p tends to zero, the fraction of time network remains a cycle also tends to be small and it is known that GMS is optimal for tree networks. Therefore, it fits well with intuition to see that fading enhances the F-LPF for graphs with cycles.

Example B: A network where fading structure worsens the relative performance of GMS: Consider the graph with 3 links a, b, c as shown above. The interference sets for each link is: $I_a = \{b\}, I_b = \{a, c\}$and $I_c = \{b\}$. We assume each link is either in state 'ON'(1) or 'OFF'(0). So the global channel state $'110'$ denotes that link a and b are in 'ON' state and link c is in 'OFF' state. The fading structure is defined as follows: $\pi('110') = \pi('011') = \pi('111') = 1/3$.

For each global channel state, the possible maximal independent sets are as follows:

$$M_{ab,abc} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix}$$

and

$$M_{bc,abc} = \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}$$

and

$$M_{abc} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{pmatrix}$$

Any vector that belongs to $\Phi(\{abc\})$ can be represented as follows,

$$\vec{\phi} = \frac{1}{3} M_{ab}[\alpha\, 1 - \alpha]' + \frac{1}{3} M_{bc}[\beta\, 1 - \beta]' + \frac{1}{3} M_{abc}[\gamma\, 1 - \gamma]'. \tag{3.10}$$

Let $\vec{\phi}_1$ be obtained using $(\alpha, \beta, \gamma) = (1, 0, 0)$ and $\vec{\phi}_2$ be obtained using $(\alpha, \beta, \gamma) = (1/2, 1/2, 3/4)$. Evaluating the above expression using the above values, we have $\vec{\phi}_1 = \frac{1}{3}[1\,1\,1]'$ and $\vec{\phi}_2 = \frac{5}{12}[1\,1\,1]'$. Observing the fact that $\frac{4}{5}\vec{\phi}_2 = \vec{\phi}_1$, using Theorem 3.3.2, we have that local pooling factor for the wireless network with the above fading structure is less than or equal to $\frac{4}{5}$. But, it is known that the local pooling factor of GMS for tree networks (with no fading) is 1.

This result though sounds counter-intuitive, stems from the fact that we allow the fading to be arbitrary. Thus fading can act as adversary and as demonstrated, can degrade the performance of GMS algorithm.

3.3.3 Characterization in terms of Interference degree

So far, we have characterized the performance of GMS through a single scaling factor of the entire throughput region. Note that each fading state J induces a network defined on the set of edges that are in 'ON' state and GMS can stabilize the network if arrivals are inside the region $\sigma^*(J)\Lambda_J$. It is natural to ask for the fading scenario, i.e. network with distribution $\pi(J)$, *if GMS could stabilize the region* $\sum_J \pi(J)\sigma^*(J)\Lambda_J$? We answer the above question in two parts.

In the first part, we show the interesting result that *GMS cannot stabilize* the above averaged region. In other words, there exists an arrival process with rate outside the region $\Lambda_f(\vec{x})$ for $x(J) = \sigma^*(J)$ (standard LPF) that can make the network unstable under GMS algorithm. We illustrate this using a simple example described below.

Counter Example: Consider the network with 3 nodes as in Example B.

Note that the standard LPF [31] for all the three fading states is 1. Thus the region $\Lambda_f(\sigma^*(J))$ is exactly same as the actual throughput region Λ_f. However, we have shown earlier that F-LPF is strictly less than 0.8. Thus there exists an arrival process with rates outside the region $0.8\Lambda_f$ that cannot be stabilized by the greedy maximal schedule.

Given the previous negative result, in the second part we show that GMS can stabilize the region $\Lambda_f(\frac{1}{d_I(J)})$. Note that this region is strictly inside the region $\Lambda_f(\vec{x})$ with $x(J) = \sigma^*(J)$. More formally, our result is as follows:

Theorem 3.3.4. *Under a given network topology and channel state distribution with Assumption A1 on the arrivals and fading channels, GMS can stabilize the network if the arrival rates are inside the region $\Lambda_f(\vec{x})$, where $x(S) = \frac{1}{d_I(S)}$.*

Implications: The above theorem provides an elegant characterization of the rate region that can be stabilizable by the GMS algorithm. Also, we find that that the above region is *not a subset* of the achievable region stated in Theorem 1b (i.e $\sigma^*_G(\pi)\Lambda_f$). We illustrate the above observation through a simple example described below.

Consider the wireless network with 3 nodes and fading distribution similar to example B. Note that the interference degree for fading state $'110'$ is $d_I('110') = 1$, for state $'011'$ is $d_I('011') = 1$ and for the fading state $'111'$ is $d_I('111') = 0.5$. Any arrival rate vector that belongs to the new region defined using the interference degree can be expressed as below,

$$\vec{\lambda} = \frac{1}{3}M_{ab}[\alpha\, 1-\alpha]' + \frac{1}{3}M_{bc}[\beta\, 1-\beta]' + \frac{1}{3}\frac{1}{2}M_{abc}[\gamma\, 1-\gamma]', \tag{3.11}$$

where α, β and γ are positive constants that are bounded by 1. Using $(\alpha, \beta, \gamma) = (0, 1, 0)$, we have that rate vector $(0, \frac{5}{6}, 0)$ is inside the new region characterized by the interference degree. However, note that we have shown the F-LPF is upper bounded by $\frac{4}{5}$ for example B network. Thus, all arrival rates that are inside the region $\frac{4}{5}\Lambda_f$ satisfy the constraint that $\lambda_2 < \frac{4}{5}$ and hence rate vector $(0, \frac{5}{6}, 0)$ belongs to the new region and not the region characterized by F-LPF.

Proof Discussion: We consider the continuous time model with deterministic arrival and channel state processes. We then exhibit a Lyapunov function, sum of squares of queue lengths, whose derivative is strictly less than zero under the GMS policy whenever the arrival rate is strictly inside the new region. Therefore, the fluid model is stable and thus using the results from [32] we conclude that the original network model is stable.

3.4 Extensions to Multiple Fading States

We now extend our results for 'ON/OFF' channels to channel models where each link capacity is time-varying and takes values from a finite state space. Let us denote the set of values in the state space by $\{0, c_1, c_2, ..., c_m\}$. The global state $GS(t)$ of the system now refers to the exact channel state of each link. Let $\pi(X_1, X_2, ..., X_K)$ denote the fraction of time the network is in global channel state $(X_1, X_2, X_3, ...X_K)$. Let us denote the state

$(X_1, X_2, X_3, ..., X_K)$ by X.

Let M_X denote the matrix consisting of K rows one for each link. Each column now represents a possible maximal independent set on the set of links with non-zero channel states. For a given column, the entries of a given row is set to zero if link l (corresponding to row) does not belong to independent set, or is set to equal to channel value X_l if it belongs to independent set. For example, consider the Interference graph in Figure 3.1 with each link taking 3 channel states $\{0, 1, 2\}$. Then $M_{(1,2,1,0)}$ is given by,

$$M_{(1,2,1,0)} = \begin{pmatrix} 1 & 0 \\ 0 & 2 \\ 1 & 0 \\ 0 & 0 \end{pmatrix}$$

The throughput region Λ_f for the above general network model with fading pattern $\pi(X)$ is given by:

$$\Lambda_f^g = \{\vec{\lambda} : \vec{\lambda} > 0 \quad , \quad \vec{\lambda} \leq \sum_X \pi(X)\vec{\eta}_X \text{ where } \vec{\eta}_X \in CH(M_X)\}.$$

We now define the F-LPF for a set of links L as follows:

$$\sigma_L^*(\pi) = \inf\{\sigma : \exists\, \vec{\phi}_1, \vec{\phi}_2 \in \Phi^g(L) \text{ such that } \sigma\vec{\phi}_1 \geq \vec{\phi}_2\}, \tag{3.12}$$

where,

$$\Phi^g(L) = \{\vec{\phi} : \vec{\phi} = \sum_X \pi(X)\vec{\eta}_X \text{ where } \vec{\eta}_X \in CH(M_{X_L})\}, \tag{3.13}$$

X_L is constructed from X by setting the values of links that do not belong to set L in X to zero.

Theorem 1 can be shown to hold for the general model with the above modified definition of F-LPF. The proof of Theorem 1 for the 'ON/OFF' channels can be easily modified to above system with general channels and is therefore omitted.

3.5 Conclusion & Discussion

In this chapter, we studied the problem of scheduling in wireless networks with interference constraints where the capacity of links changes over time. We have analyzed the performance of a well-known algorithm, Greedy-Maximal Scheduling (GMS), to the case of general wireless networks with fading structure. We defined Fading-Local pooling factor for graphs with fading and showed that it characterizes the fraction of throughput that can be achieved by GMS. We have derived useful yet easily computable bounds on F-LPF through alternate formulations.

Chapter 4

Distributed scheduling in wireless networks with cut-through routing

4.1 Introduction

Over the last few years, new radio technology has emerged enabling nodes to simultaneously transmit and receive [44] over the same frequency band. The key idea is that the wireless node has knowledge of the transmitted signal – this information is used to *cancel* the self-interference on the receive side, and thus successfully decode a packet while also simultaneously transmitting a packet. This technology, commonly referred to as full-duplex wireless, has been enabled through a combination of advances in antenna/RF circuits along with advanced digital processing, for self-interference cancellation [44, 46, 55, 58].

From a networking perspective, this technology is exciting for two reasons: *(i)* the fact that nodes can transmit and receive dramatically increases capacity by relaxing the MAC constraints in scheduling, and *(ii)* the fact that this gain occurs due to self-interference cancellation implies that *knowledge of the packet matters*. The second point is especially important – this means that at any node, if we somehow have a copy of a packet, this knowledge can be used to potentially extract any other packet that might be "mixed" with the known packet. Another implication is that packet routing paths matter in determining the MAC region capacity (along the route, nodes have copies of the packet), thus creating an "inversion" with respect to the classical network stack. We specifically leverage this effect to perform cut-through switching (CTS, whose feasibility was first demonstrated in [44, 46]), where a node simultaneously receives and re-transmits the same packet by *canceling* interference from downstream nodes along the route that are re-transmitting the same message. This effectively creates a "long jump" for a packet, where it is able to simultaneously move across several nodes in a path within a single time-slot.

As noted in [46], cut-through paths (long jumps) are appealing from an end-to-end delay perspective. Further, we observe in this work (see Section 4.2 for an example) that in fact, the inversion between the MAC and routing can potentially increase the MAC-layer capacity region (and not just delay) of full-duplex wireless systems. However, a key challenge is to manage such long jumps such that the network interference due to a larger "packet footprint" does not degrade the overall throughput-region by blocking cross-flows. The associated algorithmic challenge is to develop routing and scheduling algorithms that result in good performance. **Our contributions** are as follows: For a given multi-hop wireless network with CTS ability,

1. We propose a new low-complexity, local-information based joint routing/scheduling algorithm (that prioritizes CTS routes) for multi-hop

wireless networks. We analyze the throughput performance of the proposed algorithm and show that the efficiency ratio is greater than inverse of interference degree (Theorem 4.4.1).

2. If the arrivals to the wireless system are inside a particular fraction (inverse of the interference degree of underlying interference graph) of throughput region, we provide an upper bound on the total expected hop-delay, defined in Section 4.3.4, of the proposed algorithm (Theorem 4.4.2).

3. We further characterize the derived upper bound on the total expected hop-delay of the proposed algorithm and relate it to the lower bound on the total expected hop-delay of any algorithm that can stabilize the network (see Theorem 4.4.3). We use this result to provide a sufficient condition for *hop-delay optimality* of our proposed algorithm.

4. We also provide simulation results to justify that the proposed algorithm in-fact exploits the benefits of CTS capability of wireless networks both in terms of achievable data rates and packet delays.

4.1.1 Related Work

Over the last decade, the backpressure algorithm [42] has been the focus of intense study for scheduling and routing over wireless multi-hop networks. While this algorithm is throughput-optimal, it is known to suffer from several deficiencies including high computation complexity and centralized control.

The research effort over the last several years has focussed on addressing some of these deficiencies [39, 53, 56]. However, these algorithms (distributed and/or low-complexity) are in general not throughput optimal. A popular approximation approach with a distributed implementation [52] is the Greedy Maximal Scheduling algorithm (GMS) [54] (alternately, a Maximal Weight Independent Set algorithm). In a single-hop network context, this algorithm (or its variants) have been studied to derive throughput guarantees [51], and extended to fading channels [57]. In a multi-hop context, results include throughput guarantees [59, 60] and optimality conditions [62].

Switching tracks, in the context of performance beyond stability for joint routing and scheduling algorithms, work includes [45, 61] where centralized algorithms are developed (that modify back-pressure so as to bias toward short routes) and show that (through simulations) that they result in better end-to-end delay performance without degrading the throughput performance of back-pressure.

In parallel, radio technology has continued to evolve. Over the last few years, an exciting development [44, 58] has been the development of various analog and digital techniques that enable a wireless node to receive and transmit data at the same time. Further, the authors in [46] noted that such full-duplex wireless radios can be used to implement data forwarding via cut-through switching (CTS) in multi-hop wireless networks to potentially reduce end-to-end delays. While the study in [46] demonstrated the feasibility of CTS in full-duplex wireless, the authors did not propose routing/scheduling algo-

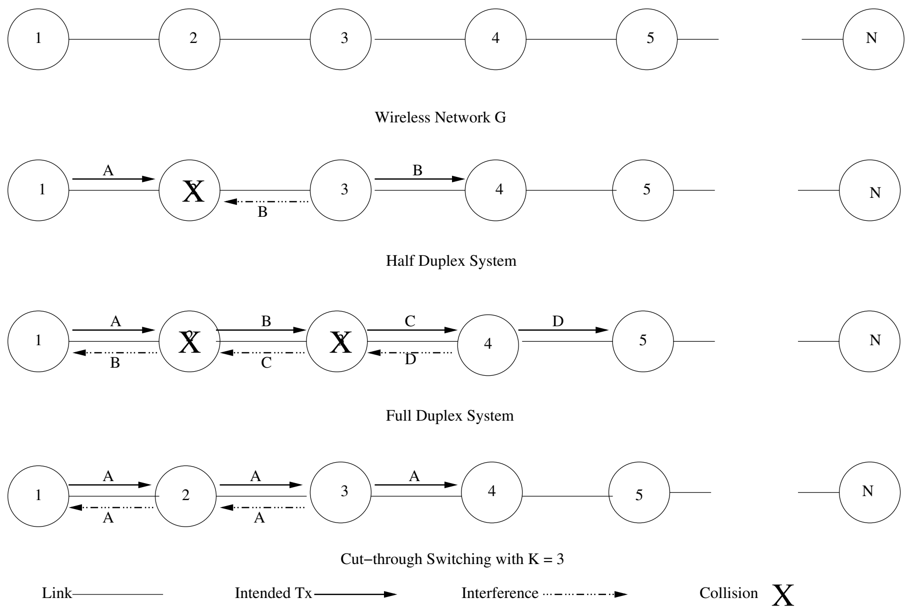

Figure 4.1: Wireless Graph with and with out CTS

rithms (nor address capacity region issues). In the earlier work on multi-hop *wireline* networks, it was shown empirically that [47] significantly helps reduce the latency in the system.

However, it is a priori not clear whether the existing low-complexity, distributed scheduling algorithms perform well (or exploit CTS) in wireless networks with full-duplex/cut-through capability, which is the focus of this work.

4.2 Example and Motivation

Consider the wireless network G (line network) as shown in Figure 4.1. The nodes in the graph represent wireless radios and edges between nodes represent possible pairwise (direct) communication (edges are bi-directional and transmission by nodes result in packets on all links incident on the node). First, let us consider a half-duplex setting (primary conflict) where each node can only either transmit or receive from another node. The objective is to deliver packets from the leftmost (node 1) to the rightmost node (node N). In this setting, we can argue that one every *third* link in the network can simultaneously support transmissions. To see this, suppose node 1 transit packet 'A', and node 2 is the intended receiver. Node 3 cannot transmit simultaneously to node 4, because this transmission would result in a collision at node 2 (recall links are bidirectional, and nodes broadcast packets). Thus, both nodes '2' and '3' cannot transmit, resulting in a long-term throughput of 1/3.

With full-duplex communications but without cut-through capability, two consecutive links can successfully support traffic simultaneously; however, the following two links need to be idle. To see this, in a time-slot suppose that node 1 transmits packet 'A' to node 2, node '2' transmits different packet 'B' to node '3', and node '3' attempts to transmit a packet 'C' to node 4 (see Figure 4.1). This scenario is not feasible because node 2 cannot decode packet 'A' from node 1 due to the interference by packet C (links are bi-directional, hence packet 'C' will collide with packet 'B' at node 2). A similar argument will result in the observation that the closest node that can successfully transmit without causing 'backward' interference is node 5. This implies that along a line, for every four links, only two carry useful packets, resulting in an average throughput of 1/2.

Finally, consider the wireless network G^K, which denotes the wireless network G with cut-through switching capability parameter $K \geq 2$. The CTS-parameter K denotes the number of hops a packet is allowed to cut-through[1]. For the wireless network with parameter K, G^K, it can be shown (the argument is similar to that used in the previous paragraph for full duplex without cut-through; the key point being that two nodes need to not transmit after each cut-through stretch) that the condition $\lambda_f < \frac{K}{K+2}$ (for large enough N) is necessary and sufficient to stabilize the network.

The above example clearly demonstrates that cut-through routing *indeed increases the data rates* beyond that of full-duplex (without CTS) achievable in a wireless network. This is not surprising as CTS essentially reduces the interference observed in the original network, as packet is able to successfully cut across multiple links with out self interference.

Apart from throughput, the other performance metric that is of interest in wireless systems is end-to-end delay. End-to-end delay of a packet consists of two parts: one is queueing delay and other is hop-delay. Queueing delay comprises of sum of the time that the packet spends waiting in the queues (to

[1]We impose a deterministic bound on the number of hops a packet can cut-through to account for propagation delay effects that build-up as the jump length increases.

be served later) and hop-delay comprises of time the packet is being transmitted on a link. It is known that lower hop-delay [61] leads to smaller end-to-end delay at low to moderate traffic loads.

Note that for the above line network G^K with N nodes, the minimum hop-delay of packets (whose source node is 1 and destination node is N) is $\lceil \frac{N-1}{K} \rceil$. Further this can be achieved using a centralized scheduler. The above example also shows that cut-through routing (apart from increasing the throughput region) can help *reduce the hop-delay* in wireless networks.

However, it is a priori not clear if the existing distributed algorithms exploit the CTS ability of wireless networks. In this regard, we develop a distributed algorithm (based on [61]) that prioritizes routes with shortest path length (or in our case CTS links) for multi-hop wireless networks.

In this work, we also provide hop-delay guarantees for the proposed algorithm (apart from throughput guanrantees), which is interesting as this is the first work that bounds the hop-delay component for a distributed algorithm. In particular, using our theoretical results, we show that the proposed algorithm is *hop-delay optimal* for the line network defined above with K-hop cut through switching. Full details will be presented in Section 5.3, Example 1. The throughput and hop-delay performance of the proposed algorithm for the above example is simulated and is presented below in Fig. 4.2 and Fig. 4.3 (for more simulation details, see section 4.5).

From the above simulation results, it can be seen that our algorithm

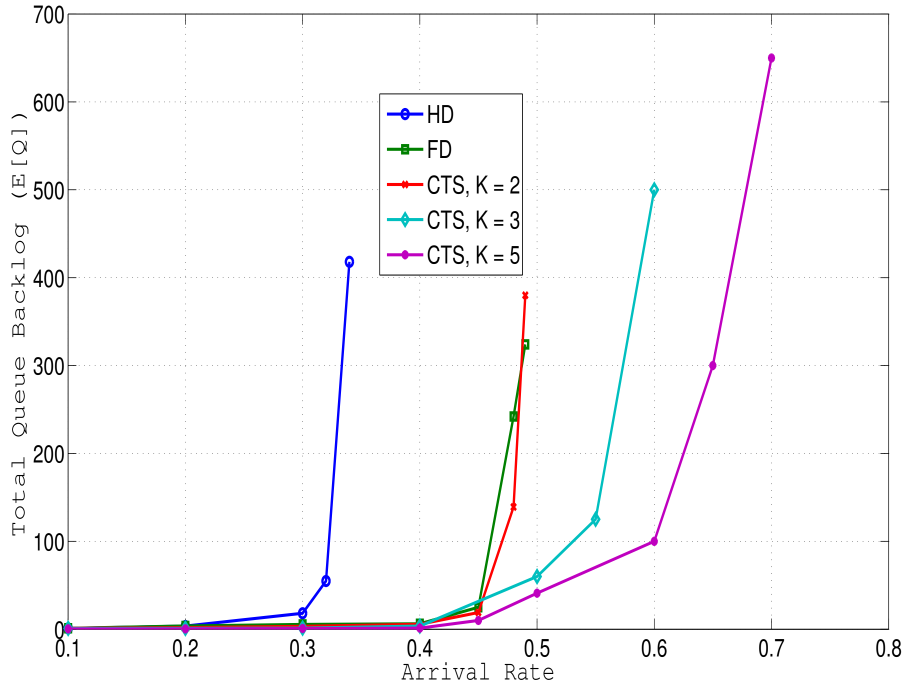

Figure 4.2: Line Network: Performance of proposed algo.

indeed exploits both the *throughput and hop-delay* gains due to cut-through switching in the case of line network.

We now consider a more general spatial network with full-duplex constraints, where we have multiple routes from source node to destination node. For detailed network model, see Section 5.3, Example 3. We use our theoretical results to compute the hop-delay bounds of the proposed algorithm and is plotted in Fig.4.4 . From the plot, it can be seen that the proposed algorithm is hop-optimal at arrival rates less than $C/2$. We also present simulation results on the above described spatial network and observe that the proposed

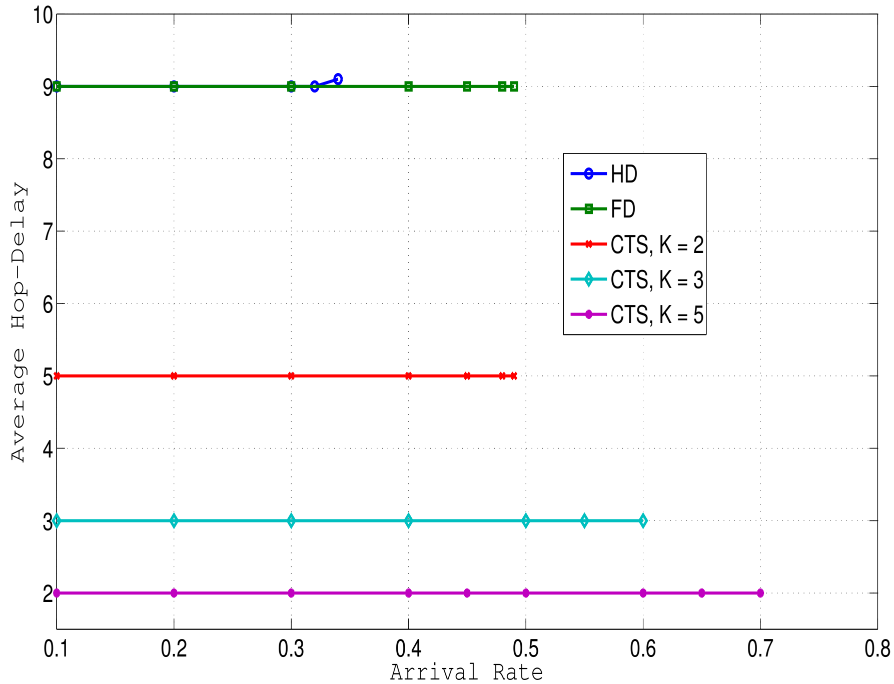

Figure 4.3: Line Network: Expected hop-delay of proposed algo.

algorithm is indeed expected hop-delay optimal at low loads as predicted using our theoretical bounds.

4.3 System Model and Preliminaries

4.3.1 Network and Traffic Model

We model a wireless network via the graph $G = (V, E)$, where V corresponds to the set of nodes (wireless radios) and E corresponds to the set of links. In our model, the links are bi-directional. For every link $l = (m, n)$, let I_l denote the set of links that interfere with its transmission. In other words,

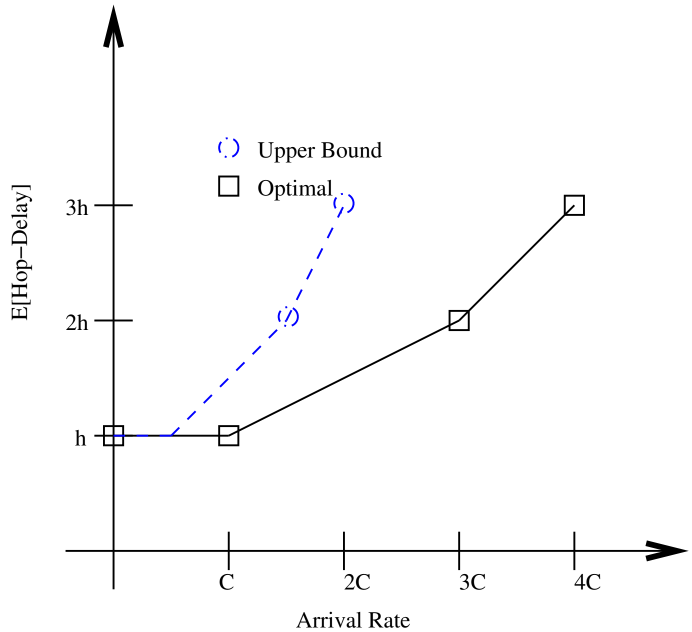

Figure 4.4: Spatial Network: Expected hop-delay performance of proposed algo. for $c_1 = 2C, c_2 = 4C, c_3 = 2C, h_1 = 2h, h_2 = 5h, h_3 = 12h$

for a packet to be successfully sent over link l, all the links in the set I_l need to be turned OFF. As we discuss below, these links need not correspond to the "physical" links between nodes, and can include virtual links (that model CTS capability).

We assume that the time is slotted and packets are of equal size. We further assume that each link's capacity is one, i.e only 1 packet can be successfully transmitted if interfering links are switched OFF. Let $A_f[t]$ denote the number of packets that arrive at the source node of flow f, $s(f)$, and needs to be sent to the destination node $d(f)$. Let F denote the set of all the flows in the network. For simplicity, we assume that the arrival processes $A_f[t]$ are independent and identically distributed across flows and time slots. Further, we assume that the arrivals in each slot are bounded (denote by $A_{\max}$). Let λ_f denote the arrival rate of flow f, i.e. $\lambda_f = E[A_f[t]]$.

4.3.2 Cut through switching

Let us denote the CTS ability by a parameter K, where K denotes the number of hops a packet can be successfully cut-through in a wireless network. Given the above described wireless network $G = \{V, E\}$, K-CTS ability can be considered as adding extra edges between nodes that are with in $K-$hop distance with appropriately defined interference set for these links. However, note that multiple links (in networks with CTS) *may have the exact same* sender node and transmitter node but have different intermediate nodes that were cut through by the links.

Let $G^K = \{V, E^K\}$ denote the new graph obtained from wireless network by adding the K-CTS ability. Note that the link $l = (m, n_1, n_2, n_3, ..., n) \in E^K$ if node n is with in K hops of node m. Further, for link $l \in E^K$, we define I_l to be the union of the set of links that interfere with the original links on its path, i.e $\{(m, n_1), (n_1, n_2), ..., (n_t, n)\}$. Thus, a packet send via cut-through routing across multiple links, from source node of first hop link to destination node of last-hop link in cut-through route, requires all the links that interfere with any of the links on the route to be turned OFF for successful transmission.

4.3.3 Preliminaries

Given the interference constraints, we can construct a *conflict graph* with nodes and directed edges as follows. Each node in the conflict graph corresponds to a link (or CTS link) in the original network and every directed edge in the conflict graph captures the interference between link A and B. In other words, there exist a directed edge from node A to node B, if transmission on link corresponding to node A in the original network interferes with the reception of packets on link corresponding to node B. Note that the conflict graph captures the asymmetric interference in wireless systems, where interference between links may not be mutual.

An *independent set* on the above conflict graph is defined as a set of nodes with no directed edges between them. Further a *maximal independent set* is an independent set to which adding a new node makes it no longer an independent set.

Observe that an independent set on the above conflcit graph can be mapped to a set of links that can successfully transmit with out any interference on the original network. Further it can also be mapped to a rate-vector, that corresponds to above schedule. We denote the set of all such rate-vectors that are generated using the independent sets on the conflict graph as *admissable rate-vectors*. We denote the set of available admissible rate-vectors in the network G^K by $\Pi(G^K)$. Note that $\pi \in \Pi(G^K)$ is a 0-1 column vector of length $|E^K|$.

A *scheduling policy* is an algorithm that decides the set of links to be activated (allowed to transmit) in each time slot. The network is said to be stable under the pair: (scheduling policy, $\{\lambda_f\}$), with the arrival rate $\{\lambda_f\}_{f\in F}$, if the expected vlaue of the sum of queue lengths is bounded. Further, we denote the arrival rate vector $\{\lambda_f\}_{f\in F}$ to be *supportable* if one can construct a scheduling policy that renders the network to be stable.

The *throughput region* Λ_f (where f denotes the set of flows in the network) is the collection of all supportable arrival rate vectors. A *throughput optimal* scheduling policy is defined as a policy that can stabilize the network for any arrival rate inside the throughput region.

For a given scheduling policy, the *efficiency ratio* ([51]) γ^*_{pol} is defined as follows,

$$\begin{aligned}\gamma^*_{pol} = \sup\{\gamma \quad : \quad & \text{the policy can stabilize for all} \\ & \text{the arrival rate vectors}\, \lambda \in \gamma\Lambda_f\}\end{aligned}$$

4.3.4 TEH-Delay Metric

For a given scheduling policy, the *Total Expected Hop-Delay (TEH-Delay)* [61] $H(.)$ is defined as follows,

$$H(\lambda_f) = \sum_f \sum_h h\lambda_{f,h}, \tag{4.1}$$

where $\lambda_{f,h}$ is the rate of flow f that reach the destination in h hops. As described in [61], this has an alternate interpretation as the expected total number of transmissions required to support the load by the given scheduling policy. Further note that $\sum_h \lambda_{f,h} = \lambda_f$ and the Expected Hop-Delay of the network is simply equal to $\frac{1}{\sum \lambda_f} H(\lambda_f)$.

4.3.5 Interference Degree ([51])

For a link l, the associated interference degree (denoted by $d_I(l)$) is defined as the largest number of links in $\{l\} \cup I_l$ that can be concurrently active without mutual interference. Further, for a graph $G = (V, E)$, we have $d_I(G) = \max_{l \in E} d_I(l)$. We have the following inequality in networks with CTS.

Lemma 1. $d_I(G) \leq d_I(G^K) \leq K d_I(G)$

Proof. The first inequality $d_I(G) \leq d_I(G^K)$ follows from the fact that every link in G is also present in G^K. For the second inequality, note that a cut-through link can be no longer than K hops, and the number of links that can be active when those K links are inactive is no more then $K d_I(G)$. □

4.3.6 Greedy/Maximal Weight Independent Set (MWIS) Algorithm [54]

Given a wireless network with an interference graph and the associated node weights, the algorithm results in a schedule consisting of a collection of nodes (links in the original graph). The node selection procedure is greedy: the node with the maximum weight is selected (if more than one, any one is chosen) and added to the schedule. Then, all its neighboring nodes are eliminated. This two-step procedure is repeated until no more nodes remain.

Further, in [52], the authors present a simple decentralized algorithm that can implement MWIS in a wireless network.

4.3.7 Backpressure Routing and Maximal Scheduling (BRMS) [62]

At each time slot, the BRMS algorithm [62] first assigns weights to each link $e = (m, n)$ in the network similar to the back-pressure algorithm [42] (i.e $max_j \left(Q_{mj}[t] - Q_{nj}[t]\right)$), where $Q_{ij}[t]$ denotes the number of packets queued at wireless node i that are destined for node j at time t. The algorithm then implements the MWIS algorithm with these assigned weights and schedules/routes accordingly.

4.4 Algorithm and Performance

In this section, we build on the shortest-path aided back pressure (SP-BP) algorithm and notation in [61] to propose a local information based greedy algorithm, and provide performance guarantees both in terms of throughput and hop delay. At each time step, the proposed algorithm essentially assigns weights for each link similar to SP-BP algorithm, and finds the route/schedule to be used in a greedy fashion.

We now describe the queue structure used for our algorithm. We assume that each node in the wireless network maintains a queue for each possible destination and possible hop budget (which our algorithm assigns to each packet). In other words, at node n, we have queues labelled n, d, h for all $d \in V$ and $h \geq H_{min}^{n \to d}(K)$., where $H_{min}^{n \to d}(K)$ denotes the minimum number of time slots needed to transmit the packet from node n to node d with no queueing delay in network graph G^K. Let Q_{ndh} denote the number of packets in the queue n, d, h that are destined for node d and have a hop budget of h hops. Similarly let Q_{nd} denote the number of packets at node n that are destined for node d.

Let us denote the *service activation vector* by $S = (S_{edh}, e \in E^K, d \in V, 0 < h < N)$. Note that S_{edh} takes only one of two values from the set $\{0, 1\}$. We say that a link activation vector is *feasible* only if the underlying link activation vector belongs to $\Pi(G^K)$, i.e $\{\pi_e = \sum_{d,h} S_{edh}\} \in \Pi(G^K)$. Let us denote the set of feasible service activation vectors by S. Also, a packet, queued at node n and destined for node d with hop budget h, that is successfully sent over link $l = (n.m)$ is assumed to be placed in to the queue labelled $m, d, h-1$.

Let $d_{ndh}(S)$ denote the net amount of service applied to queue n, d, h under the service activation vector S. Let $F_S(t)$ denote the number of slots in which service activation vector S was used during the time interval $[0, t]$.

Observe that we have the following queue dynamics,

$$Q_{ndh}[t] = Q_{ndh}[t-1] + A_{f,h}[t] I_{s(f)=n, d(f)=d} - D_{ndh}[t],$$

where $A_{f,h}[t]$ is obtained using the traffic splitting algorithm and $D_{ndh}[t]$ is given by the following,

$$D_{ndh}[t] = \sum_{S \in S} d_{ndh}(S) F_S(t).$$

We now describe the proposed routing/scheduling algorithm, Greedy SP-BP by modifying [61] (which is a centralized algorithm).

PROPOSED ALGORITHM: GREEDY SP-BP

---

1. **Traffic Splitting:** At time t, packets that arrive in to system $\lambda_f[t]$ are placed in the queue $\{s(f), d(f), h_f^*[t]\}$, where h_f^* is found using the following optimization problem:

$$h_f^*[t] \in \arg\min_{h \geq H_{min}^{s(f)\to d}(K)} \beta h + Q_{s(f),d(f),h}, \tag{4.2}$$

2. Assign weight to each link (including cut-through links) as follows:

$$W_{m,n}[t] = \max\left(0, \max_{d,h} W_{m,d,h+1}^{n,d,h}\right), \text{where} \tag{4.3}$$

$$W_{m,d,h+1}^{n,d,h} = Q_{m,d,h+1}[t] - Q_{n,d,h}[t]. \tag{4.4}$$

3. Obtain the link activation vector π_e^* using the Maximal Weight Independent Set (MWIS) algorithm with the assigned weights for each link from step 2.

4. For each link $e = (m, n)$, obtain the corresponding service activation vector S_{edh}^* from π_e^*,

$$S_{exy}^* = 1 \,\text{if}\, \pi_e^* = 1 \,\text{and}\, x, y = \arg\max W_{m,d,h+1}^{n,d,h} \tag{4.5}$$

Theorem 4.4.1. *For a given wireless network graph G with cut-through ability K, the efficiency ratio of the proposed Greedy SP-BP algorithm is greater than $\frac{1}{d_I(G^K)}$.*

Proof Ideas & Implications The proof technique we use is now standard in literature, and is as follows: we consider the set of arrivals inside the $\frac{1}{d_I(G^K)}$ fraction of the throughput region. We then define a Lyapunov function and show that under the considered set of arrivals and the proposed Greedy SP-BP algorithm, the drift is negative whenever the maximum queue exceeds a fixed threshold value. Using the Foster-Lyapunov condition, we have that the queue lengths in the system are bounded. The key idea that is used is the fact that weight of the schedule generated using maximal weight independent set is always greater than $\frac{1}{d_I(G^K)}$ fraction of any other possible schedule. The proof details are omitted due to space constraints.

Theorem 4.4.2. *For a given wireless network with arrival rate vector λ_f inside $\gamma\Lambda$, where $\gamma \in [0, \frac{1}{d_I(G^K)}]$, and given $\epsilon > 0$, the average hop delay under the proposed Greedy SP-BP algorithm (for large β) is upper bounded by $\frac{1}{\sum \lambda_f} H^{UB}(\lambda_f, \frac{1}{d_I(G^K)}) + \epsilon$, where the total hop-delay function $H^{UB}(\lambda_f, \gamma)$ is* *defined using the below optimization OPT1,*

$$
\begin{aligned}
H^{UB}(\lambda_f, \gamma) \quad &= min \sum_f \sum_{0<h<N} h\lambda_{f,h} \\
&s.t. \ \sum_f \lambda_{f,h} I_{s(f)=n,d(f)=d} + \mu^{in}_{n,d,h} \\
&\qquad \leq \mu^{out}_{n,d,h} \ \forall(n,d,h), \\
&\mu^{n,d,h}_{m,d,h+1} = 0, if h < H^{min}_{n\to d}, \\
&\left(\sum_d \sum_h \mu^{n,d,h-1}_{m,d,h}\right) \in \gamma CH(\Pi(G^K)), \\
&\sum_h \lambda_{f,h} = \lambda_f \\
&\lambda_{f,h} \geq 0, \quad \mu^{n,d,h}_{m,d,h+1} \geq 0.
\end{aligned} \tag{4.6}
$$

where $\mu^{in}_{n,d,h} = \sum_{m:(m,n)\in L} \mu^{n,d,h}_{m,d,h+1}[t]$ and $\mu^{out}_{n,d,h} = \sum_{i:(n,i)\in L} \mu^{i,d,h-1}_{n,d,h}[t]$ denote the average number of packets entering and leaving the queue n, d, h respectively.

Proof Ideas & Implications The proof technique used is similar to one used in [61], however appropriately modified to take in to account of the sub-optimality of the proposed Greedy SP-BP algorithm. The proof details are omitted due to space constraints.

This result is interesting as this is the first (to the best of our knowledge) one to characterize the hop count performance (a surrogate for delay) of a distributed algorithm in context of multi-hop wireless networks.

A natural question that arises is as follows: *How does the hop-delay performance of the proposed algorithm compare to the optimal hop-delay that can be achieved with any centralized algorithm ?.* The next result of our answers the above question and provides a sufficient condition that guarantees the *hop-delay optimality* of the proposed Greedy SP-BP algorithm.

Theorem 4.4.3. *Given the arrival rates inside the throughput region, i.e,* $\{\lambda_f\} \in \Lambda_f$ *and* $0 < \gamma < 1$*, we have the following,*

$$\hat{H}(\gamma\lambda_f) \leq H^{UB}(\gamma\lambda_f, \gamma) \leq \gamma\hat{H}(\lambda_f), \textit{where} \tag{4.7}$$

$\hat{H}(\lambda_f)$ *is given by the following optimization OPT2,*

$$\begin{aligned} \hat{H}(\lambda_f) = \quad & min \sum_f \sum_{0<h<N} h\lambda_{f,h} \\ & s.t. \;\; \sum_f \lambda_{f,h} I_{s(f)=n,d(f)=d} + \mu^{in}_{n,d,h} \\ & \qquad\qquad \leq \mu^{out}_{n,d,h} \; \forall(n,d,h), \\ & \qquad \mu^{n,d,h}_{m,d,h+1} = 0, \textit{if} h < H^{min}_{n \to d}, \\ & \qquad \left(\sum_d \sum_h \mu^{n,d,h-1}_{m,d,h} \right)_{m,n} \in CH(\Pi(G^K)), \\ & \qquad \sum_h \lambda_{f,h} = \lambda_f \\ & \qquad \lambda_{f,h} \geq 0, \;\; \mu^{n,d,h}_{m,d,h+1} \geq 0. \end{aligned} \tag{4.8}$$

Proof Ideas & Implications The key ideas used were to identify the feasible sets for the defined optimization programs OPT1 and OPT2. The details are omitted due to space constraints.

The above theorem allows us to compare the average number of hops taken by the proposed Greedy SP-BP algorithm to the minimal number of hops required by any centralized/distributed, online/offline algorithm. Hence, this can be used to evaluate how far the Greedy SP-BP algorithm is from the optimal algorithm that takes the minimal number of hops to transmit the packet from source to destination.

A sufficient condition for *hop-delay optimality* that comes out of the above inequality is as follows: *For a given* λ_f *and wireless network with defined intereference structure, the proposed algorithm is hop-delay optimal if* $\hat{H}(\lambda_f) = \gamma \hat{H}(\frac{1}{\gamma}\lambda_f)$. In other words, the proposed algorithm achieves hop-delay optimality at a given arrival rate λ_f if the optimal average hop delay at λ_f and $\frac{\lambda_f}{\gamma}$ are equal. We now describe a few applications of the above result, where we can explicitly compute the upper bounds on the hop-dealy of the proposed algorithm.

Applications of Theorem 3: Example 1 Consider the line network example (described in section II) with K-hop cut-through switching. Let us assume that we only have a single flow in the network, whose source is node 1 and destination is node N. We will now use the above result (Theorem 3) to find the upper bound on the number of hops taken by the proposed algorithm, $\frac{1}{\lambda_f} H^{UB}(\lambda_f, \gamma)$.

For the above simple line network with single flow, it is easy to see that the optimal hop-delay (over all centralized algorithms), $\frac{1}{\lambda_f}\hat{H}(\lambda_f)$, is constant and is independent of the arrival rate. Further it is also equal to the minimum hop distance between the source and destination $H_{min}^{s(f)\to d}(K)$ (which in this case is $\lceil \frac{N-1}{K} \rceil$). Using Theorem 3, we have that the upper bound on the total hop-delay on the proposed algorithm (for large β) can be bounded by the following inequality,

$$H_{min}^{s(f)\to d}(K)\gamma\lambda_f \leq H^{UB}(\gamma\lambda_f, \gamma) \leq \gamma H_{min}^{s(f)\to d}(K)\lambda_f. \quad (4.9)$$

From the above inequalities, we observe that $H^{UB}(\lambda_f, \gamma)$ is equal to $\hat{H}(\lambda_f)$. In other words, the upper bound on the average number of hops taken by the packets using the proposed algorithm is equal to $\lceil \frac{N-1}{K} \rceil$. As the minimum number of hops required for a packet to travel from source to destination is greater than $\lceil \frac{N-1}{K} \rceil$, we have that the proposed algorithm (for large values of parameter β) is indeed *hop-optimal*. The same result can be observed from our simulation results in the next section.

Applications of Theorem 3: Example 2 Consider a ring network with $2N$ wireless nodes as shown in the below figure. Let nodes be labelled as $1, 2, ...2N$. Assume we have bi-directional links and full-duplex intereference constraints and further allow cut-through switching up to $K = 2$ hops. Let us assume again, we only have a single flow in the network from node 1 to node $2M + 1$. Let the capacity of all the links be 1. In other words, we can successfully transmit one packet across the link if the transmission is interference free.

Note that the minimum hop-distance between the source node and destination node via 'route 1' (as shown in Fig. 4.5) is M and via 'route 2' is $N - M$. Also note that the interference degree for the above network $(d_I(G^2)$ is 2. Observe that for the above ring network with single flow, the total hop-delay is lower bounded as follows:

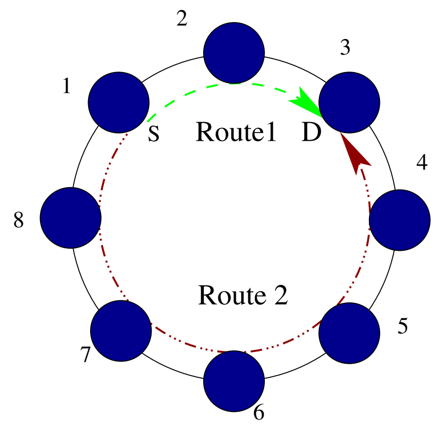

Figure 4.5: Ring network with N=4, M=1.

$$H(\lambda_f) \geq \lambda_f M \,\text{if}\, \lambda_f \leq 0.5 \tag{4.10}$$

and

$$H(\lambda_f) \geq \frac{1}{2}M + (\lambda_f - \frac{1}{2})(N - M) \,\text{if}\, \lambda_f > 0.5 \tag{4.11}$$

Also, it is easy to see that the above lower bound can be achieved using an arrival-rate aware centralized algorithm (that essentially splits the traffic across the two routes and uses the 'route 2' only when $\lambda_f > 0.5$). Given optimal total hop-delay function $\hat{H}(\lambda_f)$, we now use our result in Theorem 3, to compute the upper bound on the average hop-delay performance of the proposed Greedy SP-BP algorithm and is plotted in Fig. 4.6.

From the Fig. 4.6, observe that the proposed algorithm is *hop-delay optimal*, for all arrival rate less than $\frac{1}{4}$.

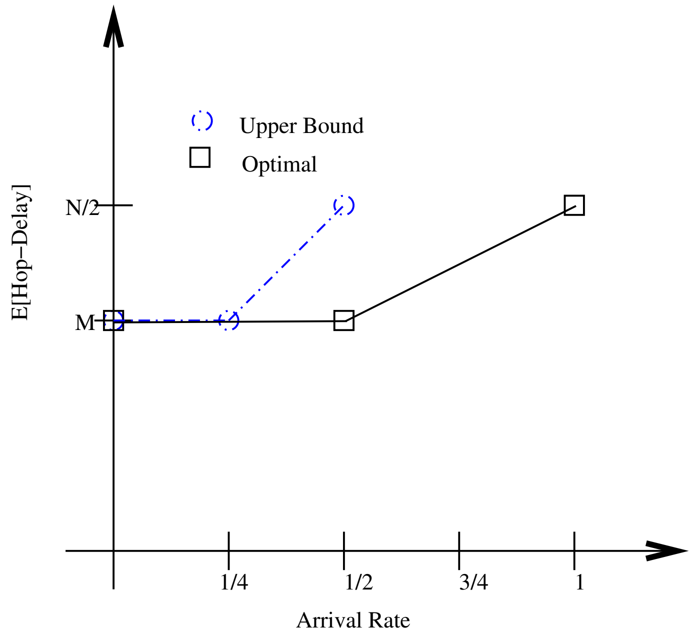

Figure 4.6: Avg. Hop-Delay performance of proposed algorithm (Upper Bound) Vs. Optimal Hop-Delay

Applications of Theorem 3: Example 3 Conisder a spatial network with multiple paths from source to destination as shown in the Fig. 4.7. Let R denote the total number of routes in the network. Let also assume that $c_i < c_j$ if $i < j$. Also assume that the number of nodes in route 1 (other than S and D) be $h_1 - 1$, number of nodes on route 2 be $h_2 - 1$ and so on. Assume that transmissions on path i do not interfere with transmissions on path j. Let us assume that we have full-duplex constraints for transmissions on links on the same path.

Note that minimum number of hops (hop-delay) a packet needs to take to reach destination on route 1 is $\lceil \frac{h_1}{2} \rceil$, on route 2 is $\lceil \frac{h_2}{2} \rceil$ and so on.

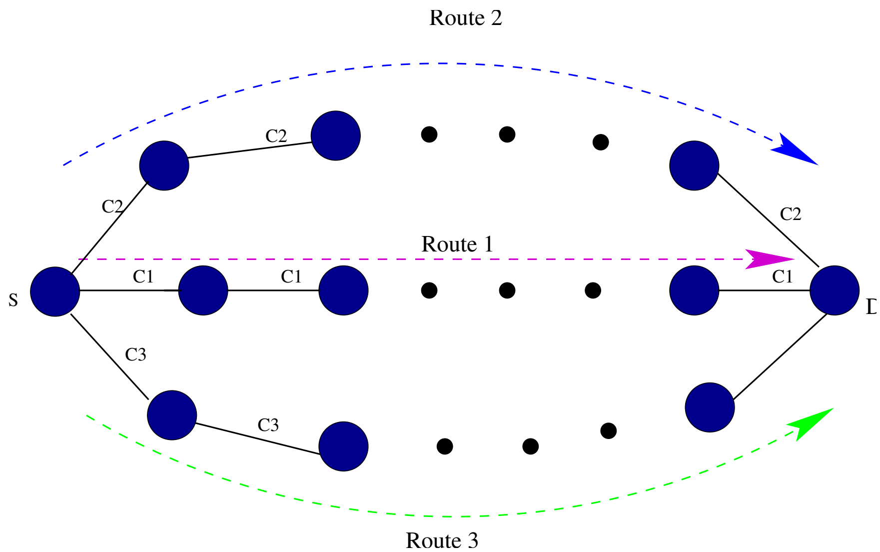

Figure 4.7: Spatial Network with multiple routes from source to destination, R =3

Observe that a *hop-delay* optimal algorithm uses the routes with the smaller path lengths and uses the longer paths only when it cannot support the arrival rates on shorter paths. Let C_k denote the quantity $\sum_{i=1}^{k} c_i$. For $R = 3$, the optimal total hop-delay function $\hat{H}(\lambda_f)$ can be expressed as follows,

$$\hat{H}(\lambda_f) = \begin{cases} \lambda_f \frac{h_1}{2} & \text{if } \lambda_f \leq \frac{C_1}{2} \\ \frac{c_1 h_1}{4} + (\lambda_f - \frac{C_1}{2})\frac{h_2}{2} & \text{if } \frac{C_1}{2} < \lambda_f \leq \frac{C_2}{2} \\ \frac{c_1 h_1 + c_2 h_2}{4} + (\lambda_f - \frac{C_2}{2})\frac{h_3}{2} & \text{if } \frac{C_2}{2} < \lambda_f \leq \frac{C_3}{2} \end{cases} \quad (4.12)$$

Using the above optimal hop-delay function $\hat{H}(\lambda_f)$, we now use our result in Theorem 3, to compute the upper bound on the average hop-delay performance of the proposed Greedy SP-BP algorithm and is plotted in Fig.4.4.

4.5 Simulations

In this section, we simulate the performance of the proposed Greedy SP-BP algorithm on simple wireless networks with and with out CTS ability. We compare the performance (achievable throughput, delay and hop delay) of our algorithm against the existing Back pressure based Routing with Maximal Independent Set (BRMS) Scheduling algorithm [62]. We now describe the simulation set up.

4.5.1 Line Network with K-CTS (Example 1)

We consider a line network G with 10 nodes. We assume that we only have a single flow f in the network, whose source node is left most node and destination node is right most node on the line network. Let λ denote the arrival rate of flow f. We consider the three following interference constraints: half-duplex, full-duplex and K-hop cut-through switching.

We run the simulations for $T = 20,000$ time slots and the total queue-backlog in the network is calculated by averaging the backlog in the last 2000 slots. The below plots show the performance of BRMS algorithm and Greedy SP-BP algorithm (with large $\beta = 100$) in Half-Duplex, Full-Duplex, CTS(K=2) and CTS(K=3) interference constraints. In particular, we plot the total queue-backlog in the network (also proportional to average delay by Little's law) and the average hop-delay for the considered arrival process.

The Figures 4.11 and 4.12 shows the performance of the proposed Greedy SP-BP algorithm for above described line network (with K=3) with

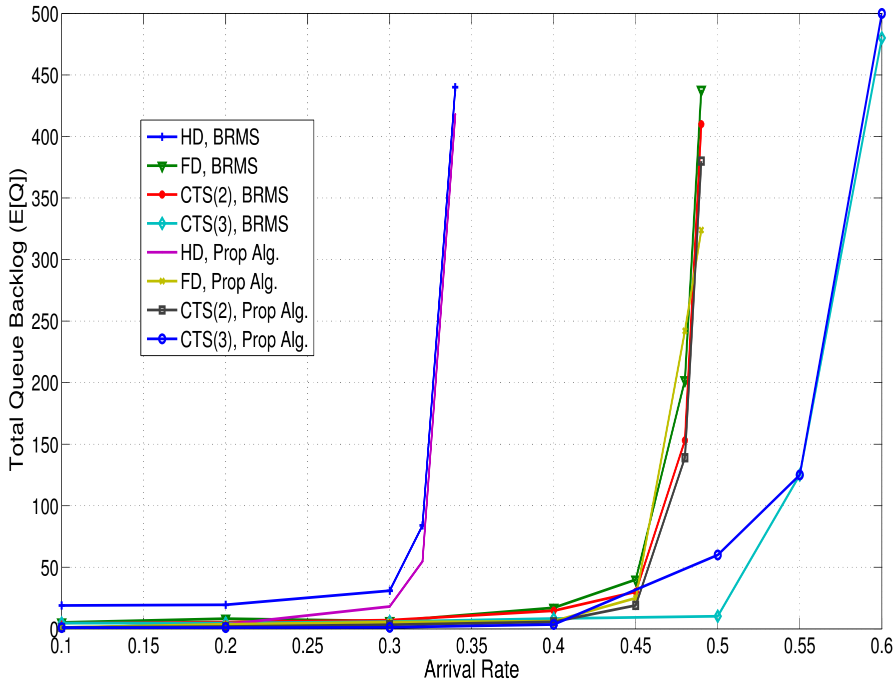

Figure 4.8: Line Network: Performance of Proposed algorithm vs. BRMS. Observe that both proposed alg. and BRMS achieve maximum possible throughput.

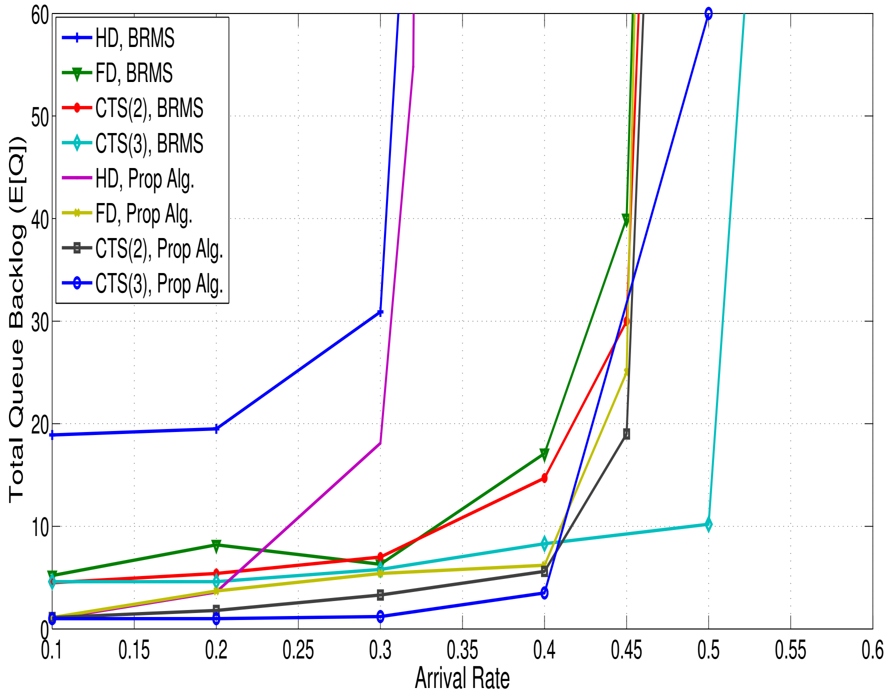

Figure 4.9: Line Network: Performance of Proposed algorithm vs. BRMS at low loads. Observe that proposed alg. has smaller backlog (smaller delay) compared to BRMS.

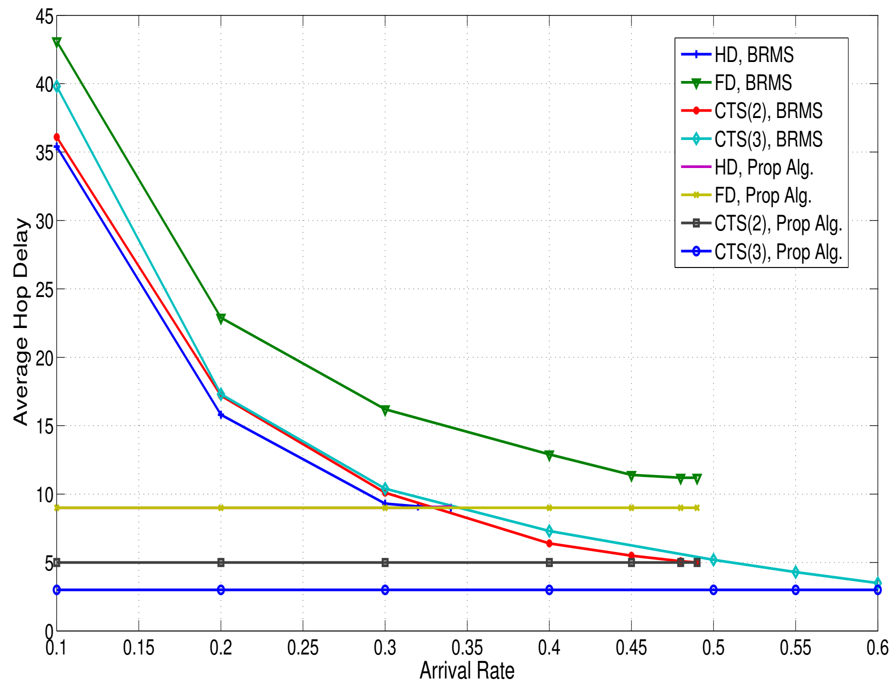

Figure 4.10: Line Network: Average Hop Delay of Proposed algorithm vs. BRMS algorithm for various interference constraints. Observe that proposed alg. is expected hop-delay optimal.

different values of algorithmic parameter β.

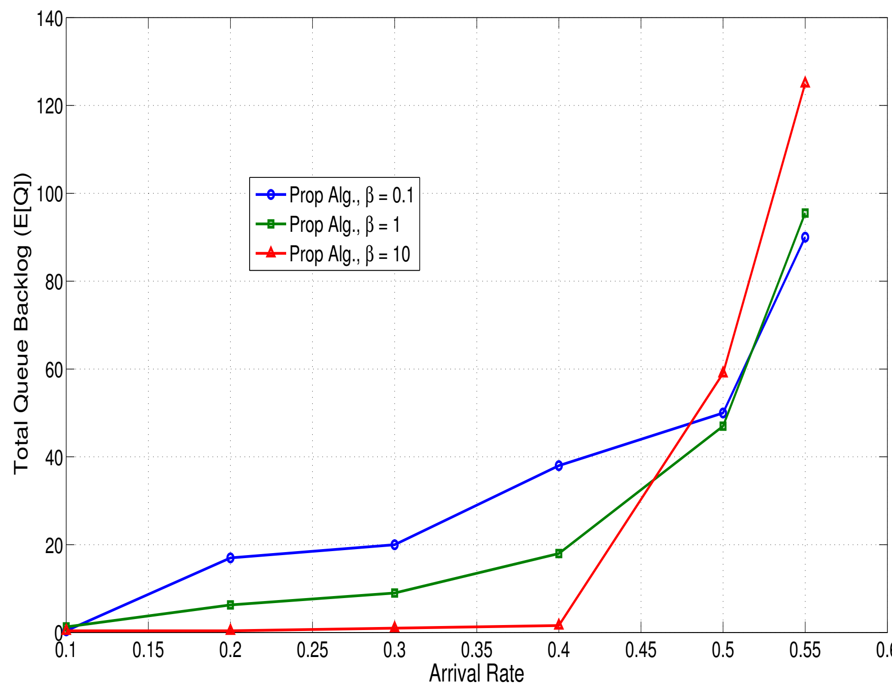

Figure 4.11: Line Network: Average total queue backlog of Proposed Greedy SP-BP algorithm with various values of parameter β

4.5.2 Spatial Network (Example 3)

We consider the spatial network (as defined in Example 3, Section 5.3) with the following parameters $c1 = 2, c2 = 4, c3 = 2, h1 = 4, h2 = 10, h3 = 24, C = 0.5, h = 2$. The performance of the proposed Greedy SP-BP (with $\beta = 10$) versus BRMS is shown below.

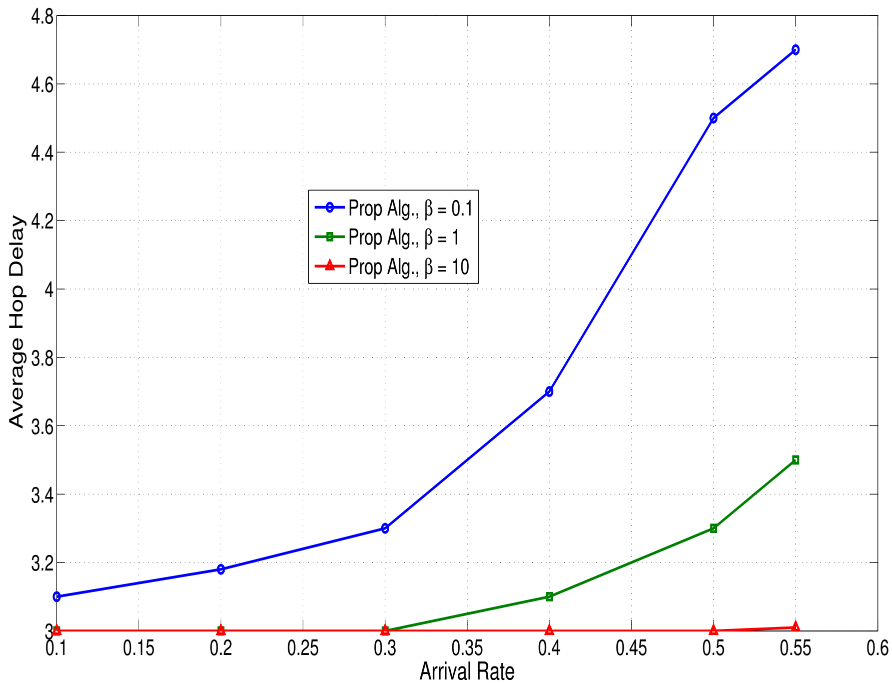

Figure 4.12: Line Network: Average Hop Delay of Proposed Greedy SP-BP algorithm with various values of parameter β

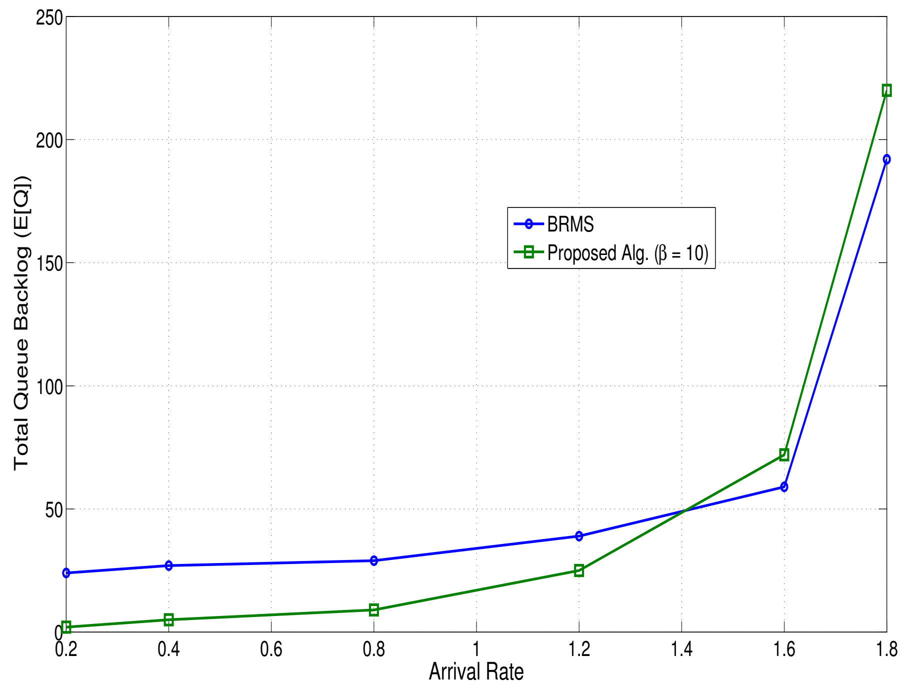

Figure 4.13: Spatial Network: Average queue backlog of proposed alg. vs. BRMS.

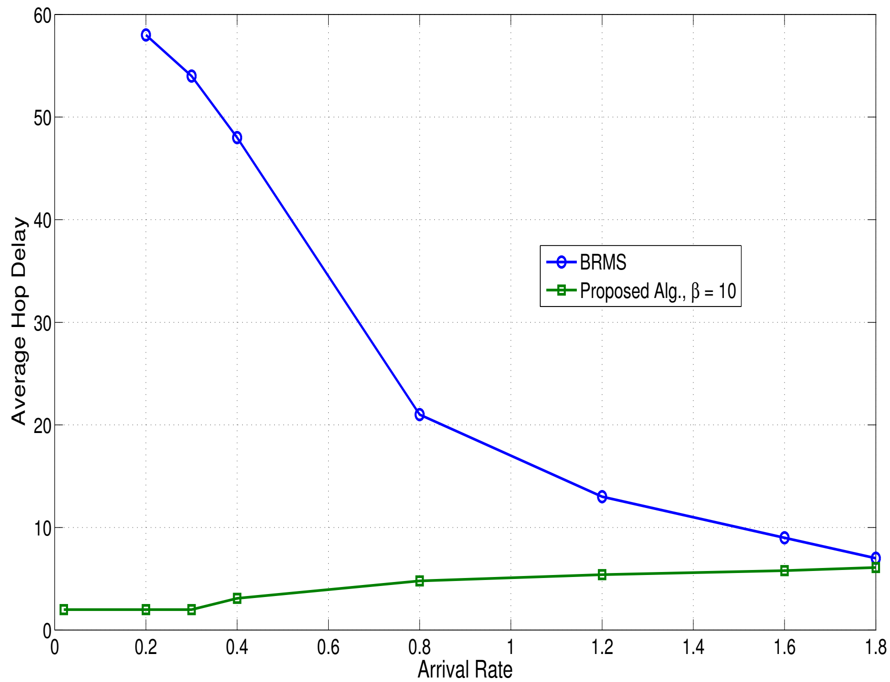

Figure 4.14: Spatial Network: Average Hop Delay of proposed algorithm vs. BRMS

4.5.3 Analysis and Discussion

From Fig. 4.8 and Fig. 4.9, we make the following observations: 1) CTS significantly increases the throughput supportable by both algorithms, 2) the proposed Greedy SP-BP significantly performs better (lower queue backlog and therefore delay by littles result) than BRMS at low-arrival rates and 3) the proposed *Greedy SP-BP stabilizes* the network *whenever BRMS stabilizes* the network.

Fig. 4.10 shows the average hop-delay of packets of flow f for both the proposed algorithm and existing BRMS algorithms. Observe that the average hop-delay of the proposed algorithm (with large $\beta = 100$) for the simulated network is constant and does not vary with the arrival rate.

Fig. 4.11 and Fig. 4.12 show the average total queue backlog and average hop-delay of proposed Greedy SP-BP algorithm on network (CTS, K=3) with various values of algorithmic parameter β. From Fig. 4.11, we observe that at low-arrival rates the proposed alg. performs (delay) better for large values of β and at higher rates, the proposed alg. performs better with small values of β. This behavior is not surprising due to the fact that at low arrival rates, the queueing delay component (of the total delay) is insignificant.

From the Fig. 4.13, we observe that the proposed algorithm (with $\beta = 10$) performs significantly better (backlog of 5 vs 27 at rate 0.4) than BRMS at low-loads. Note that, from Fig. 4.14 the proposed algorithm is hop-delay optimal for arrival rate less than $C/2$ as predicted using our theoretical bounds.

Also, note that the bounds derived in Fig. 4.4 matches well with the simulated average hop-delay in Fig. 4.14.

4.6 Conclusion

In this work, we propose a low-complexity scheduling algorithm that exploit the CTS ability of wireless networks and analyse the performance of the proposed algorithm both in terms of throughput and hop-delay. We provide a sufficient condition that guarantees the expected hop-delay optimality of the proposed algorithm. We further provide simulations to show that the proposed algorithm does exploit the CTS gains (both in terms of throughput and delay) and performs extremely well compared to existing BRMS algorithm in the low-arrival rate regime.

Chapter 5

Scheduling in wireless networks with interference alignment

5.1 Introduction

In wireless networks with multiple source-destinations pairs, it is now clear that interference mitigation is the key to achieving high data rates. Traditionally, most scheduling algorithms, [42, 67] are designed for the case where interference is avoided as in protocol interference model or interference is treated as noise as in physical interference model.

It is widely known that orthogonalization, in general, is not optimal and will not achieve the entire information-theoretic capacity region. Though characterizing the info-theoretic region itself is not known for simple networks, recently there has been progress [63–66, 68] in finding schemes that are optimal in degrees of freedom (DoF-wise), i.e., they can achieve optimal data rates at high SNR values.

In their seminal work, Cadambe et.al [65] proposed the popular linear interference alignment technique to mitigate the interference from other links. The idea is to restrict the interference space by aligning the interference from various unintended senders using linear beam-forming and there by achieving optimal number of degrees of freedom. In particular, for the $K \times K$ Gaussian interference channel with time varying channel coefficients, the authors in show that it is possible to achieve $K/2$- degrees of freedom, when certain rank constraints on the channel state matrices are satisfied. However, this technique needs to deal with symbol extension across multiple slots T and achieves $K/2$- degrees of freedom only when T tends to infinity. Recently, the authors in [69], extended the above technique to multi-hop networks with two hops.

Recently, the authors in [64], propose a new interference alignment technique, called as ergodic interference alignment which is also shown to achieve optimal degrees of freedom. The idea is to pair up channel states (in a time varying environment) and encode across these states to mitigate the interference. However, the above technique relies on the time-varying nature of the wireless network, in the sense that for every channel state H, it assumes that there exists another channel state $\hat{H}$ such that $H+\hat{H}$ is a interference free channel. Though, this condition restricts the class of channel state distributions (where ergodic IA is optimal), it is shown that with appropriate quantization, we can pair up channel states such that we almost have an interference-free channel between K source-destination pairs.

In this work, we address the following question: *Is there an online-scheduling policy, without the knowledge of channel/arrival statistics, that can extract the benefits of ergodic interference alignment technique and achieve higher data rates?*. We propose two such scheduling policies that can extract the full-benefits of ergodic IA. However, the proposed algorithms require a huge number of queues to be maintained at source and destination nodes. In this regard, we also show that the complexity (queueing) of the proposed algorithms can be reduced however with a loss in throughput. In detail, our main contributions in this context are as follows:

1. We provide a throughput characterization for general networks with ergodic interference alignment

2. We propose two scheduling policies, with new queue structure (to capture coding across different time slots), that are throughput optimal. The proposed algorithms are online (i.e only require current queue and channel state information) and do not require the knowledge of channel state/arrival rate statistics.

3. We show that our scheduling algorithms can be modified to reduce the complexity of the queueing structure required to implement and further characterize the throughput achievable using these low-complexity (in terms of queues to be maintained at source and destination) algorithms.

5.2 System Model

5.2.1 Network Model

Consider a wireless network with K links and let us associate each link with a source-destination pair. In other words, we have single hop flows and each link has packets arriving at its transmitter node (source) that needs to be transmitted over the wireless channel successfully to its corresponding receiver node (destination). Let us label the links as $\{1, 2, 3, ..., K\}$. Let s_l denote the source node of link l and d_l denote the destination node of link l.

5.2.2 Channel Model

Assume a slotted discrete time model. Each time slot is composed of two parts - control part and data part. The control part of each time slot is used for making the transmission decisions and data part to transmit the packets arriving at source node. We assume that data slot consists of n mini slots, where each mini slot is used to transmit symbols. The relation between the transmitter symbol and receiver symbol (in every mini slot) is given by the following equation,

$$y_i = h_{ii}x_i + \sum h_{ki}x_k + n_i, \tag{5.1}$$

where n_i is Guassian noise with variance N_o.

The wireless channel between the K (for K=2) source-destination pairs is shown in the below figure.

We assume that channel coefficients vary from slot to slot, however remain constant in each time slot and therefore remains constant across all the mini slots . Let channel state belong to a finite set of states denoted by $H := \{H_1, H_2, ..., H_N\}$. Assume that $H[t]$ is i.i.d. across time slots and is characterized using its stationary distribution denoted by $\pi(H)$, where H denotes the global channel state (for $K = 2, H = (h_{11}, h_{12}, h_{21}, h_{22})$). Note

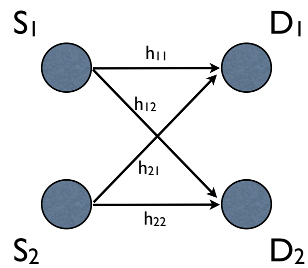

Figure 5.1: Interference model for a 2-link network

that the above model captures a wide-variety of interference models.

5.2.3 Rate Expressions using Ergodic IA

Assume that each packet consists of b bits. The number of bits that can be transmitted on link l, $r_l(.)$, in single time slot (in the absence of interference from other links) is given by the following expression,

$$r_l(h_{ll}) = TW \log\left(1 + \frac{P|h_{ll}|^2}{N_o}\right), \tag{5.2}$$

where P denotes the average power available in each slot, T denotes the time slot period, W denotes the bandwidth used and N_o is the Guassian noise variance. Equivalently, the number of packets that can be successfully transmitted on link l is $\lfloor \frac{r_l}{b} \rfloor$. Let us denote the number of packets by $R_l(.)$.

The ergodic interference alignment technique proposed in [64] combines the channel states at two different time slots (in other words transmitter sends the same signal on two different channel states and receiver combines them before decoding) so as to mitigate the interference. In particular (for K = 2), if same signal is sent on states H_1 and H_2, the number of bits that can be transmitted successfully is given by the following,

$$r_1(H_1, H_2) = TW \log\left(1 + \frac{P|h_{11}(H)|^2}{N_o + P|h_{21}(H)|^2}\right) \quad (5.3)$$

$$r_2(H_1, H_2) = TW \log\left(1 + \frac{P|h_{22}(H)|^2}{N_o + P|h_{12}(H)|^2}\right) \quad (5.4)$$

where $H = H_1 + H_2$. Let $R_1(.,.)$ denote the number of packets (i.e., $\lfloor \frac{r_1}{b} \rfloor$.

5.2.4 Throughput-Region Characterization

Let $f(H_i, H_j)$ denote the fraction of time we use (H_i, H_j) pair for alignment and let $f(H_i)$ denote the fraction of time we do not use alignment and do orthogonalization in time domain (i.e., do independent set scheduling). Let us denote the set of all independent sets when the channel is in state H_i by $IS(H_i)$. Further, let $\alpha(S, H_i)$ denote the fraction of time we schedule independent set, $S \in IS(H_i)$, when the channel state is H_i and alignment is not used. Let $\sum_{i,j}$ denote double summation over $1 \leq i, j \leq N$ and $i \neq j$. Let $A_l[t]$

denote the number of packets that arrive at the sender node s_l. We assume that $A_l[t]$ is bounded and let us denote $E[A_l[t]]$ by λ_l.

The throughput region, denoted by Λ, can be characterized as follows: an arrival rate vector $\vec{\lambda} \in \Lambda$, if there exists $\{f(H_i, H_j), f(H_i), \alpha(S, H_i)\}$ such that the following inequalities are satisfied

$$
\begin{aligned}
\lambda_l &\leq \sum_{i,j} f(H_i, H_j) R_l(H_i, H_j) + \\
&\quad \sum_i f(H_i) \sum_S \alpha(S, H_i) R_l(H_i) I_{l \in S} \,\forall\, l \\
\pi(H_i) &= \sum_{j, j \neq i} \big(f(H_i, H_j) + f(H_j, H_i)\big) + f(H_i) \,\forall i \\
1 &= \sum_{S \in IS(H_i)} \alpha(S, H_i) \\
0 &\leq f(H_i, H_j) \\
0 &\leq f(H_i) \\
0 &\leq \alpha(S, H_i).
\end{aligned}
$$

Note that the above described region is a generalization of the throughput region characterized using independent sets and fully captures the benefits of Ergodic IA.

5.3 Results

We next describe two new queuing structures that takes in to account the possibility of using ergodic interference alignment. We then describe our proposed online-scheduling algorithms that can stabilize the queues, whenever arrival rate vector is inside the throughput region. The proposed algorithms, similar to the celebrated Back-Pressure algorithm, makes the scheduling decisions only based on the current channel state and queue state information and does not require arrival-side or channel-side statistics. *Interestingly, the proposed policies naturally find the right way to combine the channel states so as to extract the benefits of ergodic interference alignment to achieve higher data rates.*

5.3.1 Proposed Queue Structure and Algorithm I

Let the source node of each link maintain the following queues. When packets arrive at the source node of link l (or user l), they are placed in queue labeled Q_l. Each link also maintains a queue at its source node for every pair of global channel state (H_i, H_j).

The queue $Q_l^{H_i,H_j}$ contains packets that were transmitted when the channel state was in state H_i and need to be re-transmitted once again in the future when the channel state is H_j, so that they can be successfully decoded at the destination node. Further the packets in the queue $Q_l^{H_i,H_j}$ are time-stamped, so that the same set of packets can be re-transmitted and be combined at the receiver while decoding the packets. Contrast to the conventional queue structures, we need to maintain a queue at the destination node and needs to store the received signal along with the time stamp. We only store packets at the receiver if they were transmitted as a part of alignment scheme. With the help of the time-stamp, the receivers can rightly combine the respective signals that were sent at two different time slots and successfully decode the packets.

We now describe the proposed scheduling algorithm,

PROPOSED ALGORITHM I

---

1. At time instant t with channel state $H[t] = H_k$, compute the independent-set scheduling weight, $W_{is}[t]$ as follows

$$W_{is}[t] = \max_{S \in IS(H_k)} \sum_{l \in S} W^I(Q_l[t], h_{ll}(H_k)), \tag{5.5}$$

where $W^I(q,h) = qR_l(h)$. Let us denote the independent sets that achieve the maximum by IS^*,

$$IS^*[t] = \left\{ S : W_{is}[t] = \sum_{l \in S} W^I(Q_l[t], h_{ll}(H[t])) \right\} \tag{5.6}$$

2. Compute the interference alignmnet weight, $W_{ia}[t]$, defined as follows,

$$W_{ia}[t] = \max \left\{ W_{ia}^I[t], W_{ia}^{II}[t] \right\}, \tag{5.7}$$

where,

$$W_{ia}^I[t] = \max_{H_i} \sum_l \left(Q_l[t] - Q_l^{H[t],H_i}[t] \right)^+ R_l(H[t], H_i) \tag{5.8}$$

$$\text{and } W_{ia}^{II}[t] = \max_{H_i} \sum_l \left(Q_l^{H_i, H[t]}[t] \right) R_l(H_i, H[t]) \tag{5.9}$$

Also, compute the following,

$$H_I^*[t] = \left\{ H_i : W_{ia}^I[t] = \sum_l \delta[t] R_l(H[t], H_i) \right\} \tag{5.10}$$

where $\delta_l[t] = \left(Q_l[t] - Q_l^{H[t], H_i}[t] \right)^+$ and

$$H_{II}^*[t] =$$
$$\left\{ H_i : W_{ia}^{II}[t] = \sum_l \left(Q_l^{H_i, H[t]}[t] \right) R_l(H_i, H[t]) \right\}$$

3. **if** $W_{is}[t] > W_{ia}[t]$, i.e., independent-set scheduling weight is greater than alignment weight, pick any $S^* \in IS^*$ and transmit $\min\{Q_l[t], R_l(H[t])\}$ packets encoded in time slot t for all $l \in S^*$. Also, discard those packets from queue Q_l and update the queue lengths.

   else

   if $W_{ia}^{II}[t] \geq W_{ia}^I[t]$

   pick any $H^* \in H_{II}^*[t]$ and each link l transmits the oldest (with respect to time-stamp) packets from the queue $Q_l(H^*, H[t])$. Further discard those packets from the respective queues.

   else

   pick any $\hat{H} \in H_I^*[t]$ and each link l transmits $\min\{Q_l[t], R_l(H[t], \hat{H})\}$ packets from the queue Q_l. Also, add these packets to queue $Q_l^{H[t], \hat{H}}$ along with the time stamp.

   end

end

Theorem 5.3.1. *If there exists* $\epsilon > 0$ *such that the arrival rate vector* $\vec{\lambda} \in (1-\epsilon)\Lambda$*, the proposed algorithm-I can stabilize the network.*

Proof. The idea is standard in the literature. We consider a Lyapunov function and show that, under the proposed algorithm the considered Lyapunov function has negative drift. Full proof with details is presented in the proofs section. □

Observation: Note that the number of queues required at each sender node, to implement the algorithm, is 1+ N^2, where N denotes the number of global channel states. For simple 2×2 network with h_{ij} taking values in a finite set of size M, we have $N = M^4$. In a wireless network with K links, N can be of size M^{K^2}. Thus making the implementation of the above proposed algorithm-I highly impractical.

Next, we will present simple ways to reduce the queueing complexity (i.e., the number of queues required to be maintained at the source node). However, the above reduction comes at the cost of sub-optimal throughput.

5.3.2 Proposed Algorithm - I : Low-complexity reductions

We now describe the procedure to reduce the queuing complexity: let A denote the set of allowable pairs that are used for ergodic IA. In other words, each node maintains queues for a pair of channel states, only if the pair belongs to set A. In other words, we limit the alignment queues at the source node. The proposed algorithm is respectively modified to take in to account that we have limited queue pairs. For example, in step 3 of algorithm, the maximum for W^I_{ia} is now found over only those channel states H_i, such that $(H[t], H_i) \in A$ and maximum for W^{II}_{ia} is found over those channel states, such that $(H_i, H[t]) \in A$.

The next theorem characterizes the data rates that can be achieved with the reduction in the queuing complexity as proposed above.

Theorem 5.3.2. *Given a set A, the proposed algorithm-I with the above mentioned queue structure restricted to set A can stabilize arrival rates that satisfy the below constraints.*

$$
\begin{aligned}
\lambda_l &\leq \sum_{i,j} f(H_i, H_j) R_l(H_i, H_j) + \\
&\quad \sum_i f(H_i) \sum_S \alpha(S, H_i) R_l(H_i) I_{l\in S} \, \forall \, l \\
\pi(H_i) &= \sum_{j, j\neq i} \big(f(H_i, H_j) + f(H_j, H_i) \big) + f(H_i) \, \forall i \\
1 &= \sum_{S \in IS(H_i)} \alpha(S, H_i) \\
0 &\leq f(H_i, H_j) \\
0 &= f(H_i, H_j) \, if (H_i, H_j) \quad A \\
0 &\leq f(H_i) \\
0 &\leq \alpha(S, H_i).
\end{aligned}
$$

Proof. The proof is similar to proof of Theorem 5.3.1 and is therefore omitted. □

We now describe our second scheduling policy that uses a different queue structure. *Essentially, we now maintain queues at the source node for each channel state, instead of pair of channel states.*

5.3.3 Proposed Queue Structure and Algorithm II

Let the source node of each link maintain the following queues. When packets arrive at the source node of link l (or user l), they are placed in queue labeled Q_l. Each link also maintains a queue at its source node for every global channel state H_i (instead of every pair as in the previous structure). The queue $Q_l^{H_i}$ contains packets that were once transmitted and need to be transmitted in the future when the channel state is H_i. Further the packets in the queue $Q_l^{H_i}$ are time-stamped, so that the packets with same time-stamp are transmitted together and thereby enabling the receiver to combine rightly the packets while decoding. Note that the destination node also needs to store the previous received signal (corresponding to packets transmitted along with the time stamp) and will use this information to combine the respective signals to successfully decode the packets. Another difference from the queue structure one is as follows : the length of queue $Q_l^{H_i}$ is counted by the number of different types (time-stamp) of packets. In other words, it is equal to the number of transmissions/slots required to empty the queue $Q_l^{H_i}$.

We now describe the second proposed algorithm,

PROPOSED ALGORITHM II

1. At time instant t with channel state $H[t]$, compute the independent set weight, $W_{is}[t]$ as follows

$$W_{is}[t] = \max_{S \in IS(H[t])} \sum_{l \in S} W^{I}(Q_l[t], H_{ll}[t]), \tag{5.11}$$

where $W^{I}(q,h) = qR_l(h)$. Let us denote the independent sets that achieve the maximum by IS^{*},

$$IS^{*}[t] = \left\{ S : W_{is}[t] = \sum_{l \in S} W^{I}(Q_l[t], H_{ll}[t]) \right\} \tag{5.12}$$

2. Compute the interference alignment weight, $W_{ia}[t]$, defined as follows,

$$W_{ia}[t] = \max \left\{ W_{ia}^{I}[t], W_{ia}^{II}[t] \right\}, \tag{5.13}$$

where,

$$W_{ia}^{I}[t] = \max_{H_i} \sum_{l} \left(Q_l[t] - Q_l^{H_i}[t] \right)^{+} R_l(H[t], H_i) \tag{5.14}$$

$$\text{and}\, W_{ia}^{II}[t] = \sum_{l} \left(Q_l^{H[t]}[t] \right) \tag{5.15}$$

Also, compute the following,

$$H_I^{*}[t] = \left\{ H_i : W_{ia}^{I}[t] = \sum_{l} \delta_l[t] R_l(H[t], H_i) \right\} \tag{5.16}$$

where $\delta_l[t] = \left(Q_l[t] - Q_l^{H_i}[t] \right)^{+}$.

3. **if** $W_{is}[t] > W_{ia}[t]$, i.e., independent-set scheduling weight is greater than alignment weight, pick any $S^* \in IS^*$ and transmit $\min\{Q_l[t], R_l(H[t])\}$ packets encoded in time slot t for all $l \in S^*$. Also, discard those packets from queue Q_l and update the queue lengths.

   else

   if $W_{ia}^{II}[t] \geq W_{ia}^{I}[t]$

   Each sender node s_l transmits oldest time stamped packets from the queue $Q_l^{H[t]}$. Further discard those packets from the respective queues.

   else

   pick any $\hat{H} \in H_I^*[t]$ and each link l transmits $\min\{Q_l[t], R_l(H[t], \hat{H})\}$ packets from the queue Q_l. Also, add these packets to queue $Q_l^{\hat{H}}$ along with time stamp.

   end

   end

Theorem 5.3.3. *The above proposed algorithm is optimal and number of queues to be maintained scales linearly in the number of channel states.*

Proof. The proof is similar to proof of Theorem 5.3.1and is therefore omitted. □

5.3.4 Proposed Algorithm - II : Low-complexity reductions

We now describe the procedure to reduce the queuing complexity : let B denote the set of channel states that are allowed to have queues at the source node. In other words, each node maintains queues for a given channel state, only if it belongs to the set B. and thereby limit the alignment possibilities at the source node. The proposed algorithm-II is appropriately modified to take in to account that we have limited queues at the source. In step 3 of algorithm, the maximum for W_{ia}^{I} is now found over only those channel states H_i, such that $H_i \in B$ and maximum for W_{ia}^{II} is assigned a weight equal to zero if $H[t]$ B. We now describe the next result, that characterizes the set of data rates that can be supportable by the above modified proposed algorithm-II with defined set B.

Theorem 5.3.4. *Given a set B, the proposed algorithm-II with the above mentioned queue structure restricted to set B can stabilize arrival rates that* *satisfy the below constraints.*

$$
\begin{aligned}
\lambda_l &\leq \sum_{i,j} f(H_i, H_j) R_l(H_i, H_j) + \\
&\quad \sum_{i} f(H_i) \sum_{S} \alpha(S, H_i) R_l(H_i) I_{l \in S} \, \forall \, l \\
\pi(H_i) &= \sum_{j, j \neq i} \big(f(H_i, H_j) + f(H_j, H_i) \big) + f(H_i) \, \forall i \\
1 &= \sum_{S \in IS(H_i)} \alpha(S, H_i) \\
0 &\leq f(H_i, H_j) \\
0 &= f(H_i, H_j) \, \textit{if} \, H_j \quad B \\
0 &\leq f(H_i) \\
0 &\leq \alpha(S, H_i).
\end{aligned}
$$

Proof. The proof is similar to proof of Theorem 5.3.1 and is therefore omitted.

□

5.4 Conclusion

In this work, we have extended the celebrated BP algorithm to wireless networks, where advanced interference technique, Ergodic IA, can be implemented. In particular, we have proposed new queue structure that can exploit the recent interference cancelation techniques to achieve high data rates. Though alignment techniques provide with increase throughput, we observe that the proposed algorithm needs huge number of queues to be maintained at the source and destination. We, therefore present low-complexity reductions to be algorithm and characterize the loss in throughput.

Observe that the above work still needs current NSI which is either infeasible or hard to get in real networks. Therefore, it is of interest to find new algorithms that can implement IA techniques with delayed NSI or no NSI.

Chapter 6

Conclusion & Future Directions

In this thesis, we have addressed some of the main drawbacks of celebrated scheduling algorithm, BP, for wireless networks. In particular, our contributions are in the research area of distributed scheduling, where each node in the network makes scheduling decisions individually. In chapter 2, we provide new scheduling algorithm (of threshold nature) for wireless networks, when each node in the network has delayed heterogeneous NSI. We also characterize the degradation in throughput with delayed NSI. In chapter 3, we analyzed the performance of popular GMS algorithm (low-complexity greedy version of BP) in wireless networks with fading structure. We show that fading can help or hurt the performance of GMS. In chapter 4, we analyzed the greedy version of SPBP algorithm (a modified BP algorithm that is hop-optimal) and identified wireless networks, for example, line network with cut-through switching, where the greedy version is hop-delay optimal. Finally, in chapter 5, we proposed a new queue structure and proposed a modified BP algorithm that can extract the benefits of ergodic interference alignment technique.

Our work opens a lot of new questions that need to be answered, both from algorithmic and analytical point of view. The questions are as follows:

1. To characterize the performance of GMS algorithm more precisely and to identify properties of fading that can always help (or hurt) the performance of GMS.

2. To find new distributed algorithms for wireless networks that can extract the benefits of Full-Duplex/Cut-through switching capabilities

3. To find scheduling algorithms that use only delayed NSI, but can incorporate new interference mitigation techniques.

Appendices

Appendix A

Proofs for Chapter 2

A.1 Proof of Lemma 2.4.1

First, assume that the arrival rates $E[\vec{A}[t]]$ are such that $\tilde{\lambda} := (1+\epsilon)E[\vec{A}[t]] \in \Lambda$ 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 $\vec{A}[t]$ is supportable, by definition, there exists a scheduling algorithm F which makes the network stable. Since the system state Markov chain $Y^F[t]$ is positive recurrent, it exhibits a stationary distribution. Let us denote the scheduling decision under policy F as $S^F(Y[t])$. We will now construct a time-sharing scheduling policy F_s that depends on the steady state distribution of queue lengths and channel states (denoted as $\pi(y), y = \{q(0:\tau_{max}), c(0:\tau_{max})\}$) under policy F. Let $r(y) = \Pr(q|c)$, computed using $\pi(y)$.

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

A.2 Proof of Theorem 2.4.2

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 (2.3),

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

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

To find $F_k^*(C_k[t] = c_i, \vec{r}), F_k^*(C_k[t] = c_j, \vec{r})$, we can solve the optimization (2.3) with other variables being fixed to the optimal solution. Consider the function that needs to be optimized:

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

Expanding this out, we can write this as

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

Note that $\vec{z} \in C^{L\tau_{max}}$ 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 $F_k(C_k[t] = c_i, \vec{r})$ and $F_k(C_k[t] = c_j, \vec{r})$, 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 $\{\vec{z} : \vec{z} \in C^{L\tau_{max}}, P_k(\vec{z}) = (c_i, \vec{r})\}$. The new function we now have is:

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

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

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

From the above equation, we observe that the optimization function is *linear* in the variable $F_k(c_i, \vec{r})$. Using the fact that channels are independent across links, we have the above function of the form $Pr(C[t] = c_i|\vec{r})(ac_iF_k(c_i, \vec{r}) + b(1 - F_k(c_i, \vec{r})))$, 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 $F_k(c_j, \vec{r})$ is of form $ac_jF_k(c_j, \vec{r}) + b(1 - F_k(c_i, \vec{r}))$. Thus the optimal solution is of the form

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

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

(Proof: Part 2) Let us consider the original function that needs to be optimized (2.3)

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

Expanding the above expression, we have

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

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 $\vec{z} \in C^{L\tau_{max}}$ is denoted by $F_l(\vec{z})$ and more specifically $F_l(P_k(\vec{z}))$. Consider a $\tau(\neq \tau_1(l) \forall l)$ and let the set $B(\tau) = \{\vec{z} \in C^{L\tau_{max}} : C_1[\tau] = c1 \,\text{or}\, C_1[\tau] = c2\}$ 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 $B1 = \{\vec{z} \in C^{L\tau_{max}} : C_1[\tau] = c1\}$ and $B2 = \{\vec{z} \in C^{L\tau_{max}} : C_1[\tau] = c2\}$. 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.

A.3 Proof of Lemma 2.4.3

Consider the following Lyapunov function $V[t]$, of the system state $Y^F[t]$, as follows,

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

We thus have,

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

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

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

Using the queue update equation, we have

$$\begin{aligned} &E\big[V[t+1] - V[t] | (Q[t-\tau_{max}], C[t-\tau_{max}])\big] \leq K+ \\ &E\big[\sum_{l \in L} (R_{l,\tau_{max}}(F^*(.)))(2Q_l[t-\tau_{max}]) | (Q[t-\tau_{max}], C[t-\tau_{max}])\big]. \end{aligned} \tag{A.1}$$

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

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

From the scheduling algorithm optimization, we also have that

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

Taking the expectation on both sides of inequality (A.1) over $C[t-\tau_{max}]$, we have that

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

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

A.4 Proof of Lemma 2.5.3

From the equation (2.5), we have

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

Let us denote the summation $\sum_{i=1}^M c_i P_{.i}^{\tau_1} 1_{c_i \geq T_{2,l}^*}$ 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

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

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

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

Using the triangle inequality, we have the following inequality,

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

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

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

Using the triangle inequality results in

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

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

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

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

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

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

A.5 Proof of Corollary 2.5.4

From equation (2.4), 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

$$|P_{ij}^{\tau} - \pi_j| \leq (1 - M\delta)^{\tau}. \tag{A.2}$$

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 (A.2) in the above inequality, we have the desired corollary.

Appendix B

Proofs for Chapter 3

B.1 Proof of Theorem 3.3.1

The proof follows the method developed by the authors in [33, 51] for the non-fading case; however we have extended it to take in to account the fading structure. First, for the converse (to show instability for arrivals outside the stability region), we explicitly construct an adversarial channel variations pattern that satisfies the time-averages imposed by the fading assumption, and this is used in conjunction with the adversarial arrival process. The achievability part is more straightforward – we augment the analysis in [33, 51] to include the fluid limit of the channel fading process. We now provide the proof more detail:

Proof of Theorem part (a): The result follows from the following general lemma.

Lemma 2. *If there exists a subset of links $L(\subseteq K)$, a positive number σ and two vectors $\vec{\mu}, \vec{\nu} \in \Phi(L)$ such that $\sigma\vec{\mu} > \vec{\nu}$, then for arbitrary small $\epsilon > 0$, there exists a traffic pattern with offered load $\vec{\nu} + \epsilon\vec{e}_L$ and a fading pattern, such that system is unstable under greedy maximal schedule.*

Proof (Lemma 2): The idea of the proof is as follows – we construct a traffic pattern and channel variations pattern with offered load $\vec{\nu}+\epsilon\vec{e}_L$ and show that under this traffic/channel fading pattern, the queue lengths go to infinity under GMS, thus making the system unstable.

As remarked earlier, this proof technique was introduced in [51], where authors only needed to construct adversarial arrival process that makes the queues in the system to overflow. However, in our setting, we need to account for the fading process and construct both arrival and channel fading pattern that makes the network unstable.

Since $\vec{\nu} \in \Phi(L)$, there exist vectors $\vec{w}^J$ such that $\vec{\nu}$ can be expressed as,

$$\vec{\nu} = \sum_{J \subseteq L} \pi_L(J)\Big(M_{J,L}\vec{w}^J\Big). \tag{B.1}$$

Fix $\delta > 0$, we then find a vector $\vec{r}^J$ in the set of rational numbers, $\mathbb{Q}$, such that $\|\vec{r}^J - \vec{w}^J\| < \delta$.

Assume packets arrive to a link at beginning of the time slot. Let the queues of all the links in L are empty at $t = 0$. Let T_J be the smallest integer such that for all i, $r_i^J T_J$ is an integer. Let $t_i^J = r_i^J T_J$. Also, there exists integers $n_1, n_2, ...n_{2^L}$ such that

$$\Big|\frac{n_J T_J}{\sum_{S:S\subseteq L} n_S T_S} - \pi_L(J)\Big| \le \frac{\delta}{2^L}. \tag{B.2}$$

Let us define $\tilde{\pi}_L(J) \in \mathbb{Q}$ as follows,

$$\tilde{\pi}_L(J) := \frac{n_J T_J}{\sum_{S\subseteq L} n_S T_S}. \tag{B.3}$$

Using the rational quantities $\tilde{\pi}_L(J)$ and $\vec{r}^J$, we define $\vec{\nu}^r$ as follows,

$$\vec{\nu}^r = \sum_{J:J\subseteq L} \tilde{\pi}_L(J)\big(M_{J,L}\vec{r}^J\big). \tag{B.4}$$

Consider a total time period of $\sum_J n_J T_J$. We assume that channel state remains in J state for T_J time slots (denoted as a time frame). It is easy to observe that with the above described fading pattern, we achieve the same channel state distribution as $\tilde{\pi}_L(J)$ on links of set L. We now describe the arrival pattern for T_J time slots when the channel is in state J.

Assume that all the queue lengths (of links in L) are equal at the beginning of T_J time slots. We now construct arrival pattern that keeps the queue lengths of all links in set L equal at the end of T_J time slots under the GMS policy. The arrival process is as follows:

1. The time frame of T_J slots is further divided in to $t_1^J, t_2^J, \ldots . t_{|IS^J|}$ time slots, where $t_i^J = r_i^J T_J$ and $|IS^J|$ denotes the number of columns in M_J.

2. During the $t_i^J, i \neq |IS^J|$ time slots, apply one packet to each link that is 'ON' in the i^{th} column of M_J. For the last $t_{|IS^J|}^J$ time slots, apply one packet to each link that is ON in the last column of M_J at the beginning of the time slot except for the last one time slot. For the last one time slot, with probability $1-\epsilon$ we do the same as described before and with probability ϵ, we apply two packets to each link that is ON in the last column of M_J and 1 packet to rest of links in L.

Note that the arrival process is modified compared to one proposed in [51] so as to ensure that all queues remain equal after T_J time slots.

It is now easy to see that at the end of T_J time slots, all the queue lengths are equal and increase by 1 with probability ϵ. Thus the above arrival and channel variation pattern make the system unstable under GMS schedule. We now show that the arrival rate is same as $\vec{\nu} + \epsilon\vec{e}_L$.

Let $\vec{e}_i$ denote the vector of all zeros except for i th position which is set to one. Let $\sum_J = \sum_{J \subseteq L}$ for the remaining part of the proof. For the constructed adversarial arrival process, the arrival rate is given by the following,

$$\vec{\lambda}_{\text{adv}} = \frac{\sum_J n_J(\sum_{i=1}^{|IS^J|} t_i^J M_J \vec{e}_i + \epsilon\vec{e})}{\sum_J n_J(\sum_{i=1}^{|IS^J|} t_i^J)} \tag{B.5}$$

Rewriting the above expression in terms of $\tilde{\pi}_L(J)$, we have that

$$\vec{\lambda}_{\text{adv}} = \sum_J \tilde{\pi}_L(J)(\sum_{i=1}^{|IS^J|} r_i^J M_J \vec{e}_i) + \epsilon\big(\sum_J \frac{\tilde{\pi}_L(J)}{T_J}\big)\vec{e} \tag{B.6}$$

Thus we have,

$$\vec{\lambda}_{\text{adv}} = \sum_J \tilde{\pi}_L(J)\big(M_{J,L}\vec{r}^J\big) + \epsilon(\sum_J \frac{\tilde{\pi}_L(J)}{T_J})\vec{e} \tag{B.7}$$

We choose small enough δ so that the arrival rate is strictly less than $\vec{\nu} + \epsilon\vec{e}_L$.

Proof (Theorem 1. b): *This proof is a simple extension of that in [33, 51], however modified to include the fluid limit arising due to the channel* *fading process. Thus, we have provided a detailed sketch and refer to [33, 51] for full details.*

We consider the fluid limit of the queuing process and we provide a Lyapunov function and show negative drift under GMS schedule whenever arrival rate $\vec{\lambda} \in (\sigma^*_G(\pi) - \epsilon)\Lambda_f$.

Consider a sequence of systems $\frac{1}{n}\vec{Q}^n(nt)$ (scaled in time and space by a factor of n), where $\vec{Q}^n(.)$ denotes the queue lengths of original system, satisfying $\sum Q^n_l(0) \leq n$ at time $t = 0$. Let us index the sequence of systems by $n = \{1, 2, ...\}$. We apply the same arrival processes to all the above defined systems (i.e $\vec{A}^n(.) = \vec{A}(.)$) and assume that queues are served according to greedy maximal schedule. Let $\vec{A}^n(t)$ and $\vec{D}^n(t)$ denote the cumulative arrival and departure process of system n up to time t.

Using the results from [32], it can be shown that the sequence of processes $(\vec{Q}^n(.), \vec{A}^n(.), \vec{D}^n(.))$ as $n \to \infty$ converges to a fluid limit almost surely along a subsequence $\{n_k\}$ in the topology of uniform convergence over compact sets,

$$\frac{1}{n_k} A^{n_k}_l(n_k t) \;\to\; \lambda_l t, \tag{B.8}$$

$$\frac{1}{n_k} D^{n_k}_l(n_k t) \;\to\; \sum_J \pi(J)\Big(\int_0^t \mu^J_l(s)ds\Big), \tag{B.9}$$

$$\frac{1}{n_k} Q^{n_k}_l(n_k t) \;\to\; q_l(t). \tag{B.10}$$

Also, the fluid limits $(q_l(t), \mu_l^J(t))$ satisfy the following equality:

$$q_l(t) = q_l(0) + \lambda_l t - \sum_J \pi(J)\big(\int_0^t \mu_l^J(s)ds\big). \tag{B.11}$$

Moreover, fluid limits are absolutely continuous, and at regular times t (i.e., those points in time where the derivatives exist) we have the following condition satisfied:

$$\frac{d}{dt}q_l(t) = \begin{cases} \lambda_l - \mu_l(t) & \text{if} \quad q_l(t) > 0 \\ (\lambda_l - \mu_l(t))^+ & \text{if} \quad q_l(t) = 0, \end{cases}$$

where $\mu_l(t) = \sum_J \pi(J)\mu_l^J(t)$ satisfies the GMS properties. Let L_0 denote the set of links with the longest queues at time t,

$$L_0(t) = \big\{i \in K | q_i(t) = \max_{j \in K} q_j(t)\big\} \tag{B.12}$$

Let $L(t)$ denote the set of links with the largest derivative of queue length among the links in $L_0(t)$,

$$L(t) = \big\{i \in L_0(t) | \frac{d}{dt}q_i(t) = \max_{i \in L_0(t)} \frac{d}{dt}q_i(t)\big\} \tag{B.13}$$

Lemma B.1.1. *Under the greedy maximal schedule, the service rate satisfies* $\vec{\mu}(t)|_{L(t)} \in \Phi(L(t))$, *where* $\vec{u}|_L$ *denotes the projection of vector on* u *on to set of links* L.

The proof of the above lemma is similar to one in [33, 51] and is presented in appendix. The idea, roughly is that, queues in the set $L(t)$ will remain the longest for small enough amount of time past t and GMS picks the maximal schedule restricted to links in $L(t)$ that are in 'ON' state.

Since the arrival rates are strictly with in $\sigma_L^*(\pi)\Lambda_f$, there exists a service vector $\vec{\nu} \in \Phi(L)$ and $\vec{\nu} < \sigma_L^*(\pi)\Lambda_f$ such that $\vec{\lambda}(L) \leq \vec{\nu}$, where $\vec{\lambda}(L)$ is projection of arrival vector on to the set L. Given any two vectors in set $\Phi(L)$, note that one vector never dominates the other one in all the dimensions by a factor more than $\sigma_L^*(\pi)$. Therefore we have that $\frac{d}{dt}\max_{i\in L(t)} q_i(t)$ is strictly negative when ever $\max q_i(t) > 0$.

Let $V(t) = \max q_l(t)$ denote the Lyapunov function used for the fluid system. Since we have a negative drift for the Lyapunov function, using the results from [32], we have that fluid system is stable (i.e there exists $t_0 > 0$ such that $q_l(t) = 0 \, \forall t > t_0$). Therefore from [32], we have that the queues in the original queuing system are stable.

B.2 Proof of Theorem 3.3.2

Since $(\vec{\mu}_J, \vec{\nu}_J, H_J)$ satisfy the inequality,

$$\vec{\nu}_J \leq H_J \vec{\mu}_J \tag{B.14}$$

Summing over all subsets with positive scaling constants $\pi(J)$,

$$\sum_J \pi(J)\nu_J(l) \leq \sum_J \pi(J)\big(H_J \mu_J(l)\big) \tag{B.15}$$

Using the maximum constant over all the inequalities, we have the following,

$$\sum_J \pi(J)\vec{\nu}_J \leq \Big(\max_l \frac{\sum_J \pi(J) H_J \mu_J(l)}{\sum_J \pi(J)\mu_J(l)}\Big) \sum_J \pi(J)\vec{\mu}_J \qquad \text{(B.16)}$$

By observing the fact that $(\sum_J \pi(J)\vec{\nu}_J, \sum_J \pi(J)\vec{\mu}_J)$ belong to the $\Phi(K)$, we have the result.

B.3 Proof of Theorem 3.3.3

We first state a lemma that describes the dual problem that finds the fading Local Pooling Factor as the optimal solution. The dual characterization of Local Pooling Factor was presented previously in [33, 38]. We now provide such characterization for F-LPF in Lemma B.3.1 by generalizing the arguments in [38]. In particular, the multiple global channel states due to fading each induce a different constraint – combining all of these appropriately while satisfying the long-term average fractions $\{\pi_L(J)\}$ results in a max min problem, as detailed below. This result is used to derive the lower bound.

Lemma B.3.1. *The following optimization problem characterizes $\sigma_L^*(\pi)$:*

$$\sigma_L^*(\pi) = \max \sum_{J: J \subseteq L} \pi_L(J) a(J)$$

$$s.t: \; x' M_{J,L} \geq a(J) e' \quad \forall J \subseteq L$$

$$x' M_{J,L} \leq b(J) e' \quad \forall J \subseteq L$$

$$\sum_{J \subseteq L} \pi_L(J) b(J) = 1$$

From the above Lemma B.3.1, we have that $\sigma_L^*(\pi)$ is equal to,

$$\begin{aligned}
&\max_{x,a(J),b(J)} \sum_{J:J\subseteq L} \pi_L(J)a(J) \\
\text{s.t} :\ & x' M_{J,L} \geq a(J)e' \quad \forall J \subseteq L \\
& x' M_{J,L} \leq b(J)e' \quad \forall J \subseteq L \\
& \sum_{J\subseteq L} \pi_L(J)b(J) = 1
\end{aligned}$$

Observe that $(\frac{1}{\sum \pi_L(J)N(M_J)}e, \frac{n(M_J)}{\sum \pi_L(J)N(M_J)}, 1)$ is a valid point in the search space. Substituting the point in the above function, we have the desired inequality.

B.4 Proof of Lemma B.3.1

Consider the definition of $\sigma_L^*(\pi)$ in (3.4). The corresponding optimization problem is given by:

$$\begin{aligned}
&\inf \qquad \sigma \\
\text{s.t} :\ & \sigma \sum_{J\subseteq L} \pi_L(J) M_{J,L} \vec{\alpha}(J) \geq \sum_{J\subseteq L} \pi_L(J) M_{J,L} \vec{\beta}(J) \\
& \|\vec{\alpha}(J)\| = 1 \qquad \forall \quad J \subseteq L \\
& \|\vec{\beta}(J)\| = 1 \qquad \forall \quad J \subseteq L \\
& \vec{\alpha}(J), \vec{\beta}(J) \geq 0
\end{aligned}$$

where $\|.\|$ is defined as the sum of all the elements of the vector. Let us define a new variable $\vec{\gamma}(J) = \sigma\vec{\alpha}(J)$. Thus, we have:

$$\begin{aligned}
&\inf \qquad \sigma \\
\text{s.t :} \quad & \sum_{J \subseteq L} \pi_L(J) M_{J,L}(\vec{\beta}(J) - \vec{\gamma}(J)) \leq 0 \\
& \|\vec{\gamma}(J)\| = \sigma \qquad \forall \quad J \subseteq L \\
& \|\vec{\beta}(J)\| = 1 \qquad \forall \quad J \subseteq L \\
& \vec{\gamma}(J), \vec{\beta}(J) \geq 0
\end{aligned}$$

For the above LP, let $(\vec{x}, \{y(J)\}, \{z(J)\})$ denote the dual variables associated with the constraints. The dual is given by

$$\begin{aligned}
&\max_{\vec{x},\{y(J)\},\{z(J)\}} \min_{\sigma,\vec{\alpha}(J),\vec{\beta}(J)} \sigma + \\
&\qquad \sum_{i=1}^{L} x_i \Big(\sum_{J \subseteq L} \pi_L(J) [\sum_{j=1}^{|IS_J|} M_{ij}^J (\beta_j^J - \gamma_j^J)] \Big) + \\
&\qquad \sum_{J \subset L} y(J) \big(\vec{\gamma}(J)' e - \sigma \big) + \\
&\qquad \sum_{J \subset L} z(J) \big(\vec{\beta}(J)' e - 1 \big) \\
&\text{s.t:} \vec{\gamma}(J), \vec{\beta}(J) \geq 0
\end{aligned}$$

Rewriting the above dual optimization problem, we have

$$\begin{aligned}
&\max_{\vec{x},\{y(J)\},\{z(J)\}} \min_{\sigma,\vec{\alpha}(J),\vec{\beta}(J)} - \sum_J z(J) + \sigma(1 - \sum_J y(J)) + \\
&\qquad \sum_{j=1}^{|IS_J|} \beta_j^J \big[\pi_L(J) \sum_{i=1}^{L} x_i M_{ij}^J + z(J) \big] + \\
&\qquad \sum_{j=1}^{|IS_J|} -\gamma_j^J \big[\pi_L(J) \sum_{i=1}^{L} x_i M_{ij}^J + y(J) \big] \\
&\text{s.t:} \vec{\gamma}(J), \vec{\beta}(J) \geq 0
\end{aligned}$$

Equivalently, the above program can be reduced to

$$\begin{aligned}
&\max \sum_{J:J\subseteq L} -z(J) \\
\text{s.t : } &\pi_L(J)x'M_{J,L} + z(J)e' \geq 0 \quad \forall J \subseteq L \\
&-\pi_L(J)x'M_{J,L} + y(J)e' \geq 0 \quad \forall J \subseteq L \\
&\sum_{J\subseteq L} y(J) = 1
\end{aligned}$$

Denoting $\frac{-z(J)}{\pi(J)}$ by $a(J)$ and $\frac{y(J)}{\pi(J)}$ by $b(J)$ we have the desired result.

Corollary 1: $\sigma^*_G(\pi) \geq \frac{1}{d_I(G)}$

Proof. Observing the fact that $n(M_J) \geq \frac{1}{d_I(G)} N(M_J)$ and using the above lemma B.3.1, we have the desired inequality. □

B.5 Proof of Theorem 3.3.4

We consider a continuous model similar to the one described in the proof of Theorem 1b. In this model, the queuing system evolves according to the following equation,

$$\frac{d}{dt} q_l(t) = \begin{cases} \lambda_l - \mu_l(t) & \text{if} \quad q_l(t) > 0 \\ (\lambda_l - \mu_l(t))^+ & \text{if} \quad q_l(t) = 0, \end{cases}$$

where $\mu_l(t) = \sum_J \pi(J)\mu_l^J(t)$ satisfies the GMS properties. In the original system with fading channels note that the weight of GMS schedule is always greater than $\frac{1}{d_I(S)}$ of the weight of the max-weight schedule where S is the set of links that are in 'ON' state. Therefore in the fluid model, we can show that $\mu_l^J(t)$ satisfies the following condition

$$\sum_l q_l(t)\mu_l^J(t) \geq \frac{1}{d_I(J)} \max_{\vec{\eta}_J \in CH(M_{J,K})} \sum_l q_l(t)\eta_J(l).$$

Let us the consider the following Lyapunov function,

$$V(\vec{q}(t)) = \sum_l q_l^2(t). \tag{B.17}$$

Taking the derivate of the Lyapunov function, we have that

$$\dot{V}(\vec{q}(t)) \leq 2\sum_l q_l(t)(\lambda_l) - \mu_l(t)). \tag{B.18}$$

Using the GMS properties of $\mu_l(t)$, we have

$$\dot{V}(\vec{q}(t)) \leq \left(2\sum_l q_l(t)\lambda_l - \sum_J \frac{2}{d_I(J)}\pi(J) \max_{\vec{\eta}_J \in CH(M_{J,K})} \sum_l q_l(t)\eta_J(l) \right) \tag{B.19}$$

As $\vec{\lambda}$ is assumed to lie inside the region $\Lambda_f(\vec{x})$, there exists $\vec{\eta}_J \in CH(M_{J,K})$ such that

$$\lambda_l < \sum_J \frac{1}{d_I(J)}\pi(J)\eta_J(l). \tag{B.20}$$

Using the above inequality, we have that

$$\dot{V}(\vec{q}(t)) < \left(2\sum_l q_l(t) \sum_J \frac{1}{d_I(J)}\pi(J)\eta_J(l) - \sum_J \frac{2}{d_I(J)}\pi(J) \max_{\vec{\eta}_J \in CH(M_{J,K})} \sum_l q_l(t)\eta_J(l) \right) \tag{B.21}$$

Thus from the above inequality we have that $\dot{V}(q(t)) < 0$ whenever $q(t) > 0$.

We can now use the results from [32] to argue that the original system is stable under the assumed arrival process as the fluid model is stable.

B.6 Proof of Lemma B.1.1

The proof is similar to the one presented in [51] however taking in to account the channel fading. From the definition of set $L_0(t)$ in Eqn (B.12), there exists $\epsilon_1 > 0$ such that

$$q_i(t) > q_j(t) + \epsilon_1 \quad \forall\, i \in L_0(t) \,\text{and}\, j \in K \;\; L_0(t).$$

Using the continuous property of $q_l(t)$, we further have that, there exists $\epsilon_2 > 0, \delta_1 > 0$ such that

$$\min_{i \in L_0(t)} q_i(t+\delta) > \max_{j \in K L_0(t)} q_j(t+\delta) + \epsilon_2 \,\forall\, \delta \in [0, \delta_1].$$

Since $L(t)$ is contained inside $L_0(t)$, we have that, there exists $\epsilon_2 > 0, \delta_1 > 0$ such that

$$\min_{i \in L_(t)} q_i(t+\delta) > \max_{j \in K L_0(t)} q_j(t+\delta) + \epsilon_2 \,\forall\, \delta \in [0, \delta_1]. \tag{B.22}$$

Also, from the definition of set $L(t)$ in Eqn (B.13), there exists $\epsilon_3 > 0$ such that

$$\frac{d}{dt} q_i(t) > \frac{d}{dt} q_j(t) + \epsilon_3 \quad \forall\, i \in L(t) \,\text{and}\, j \in L_0(t) \;\; L(t).$$

Further, using the definition of derivative $\frac{d}{dt}q(t) \approx \frac{q(t+\delta)-q(t)}{\delta}$, there exists $\epsilon_4 > 0, \delta_2 > 0$ such that the following holds. For all $i \in L(t)$ and $j \in L_0(t)\ \ L(t)$, we have

$$\frac{q_i(t+\delta) - q_i(t)}{\delta} > \frac{q_j(t+\delta) - q_j(t)}{\delta} + \epsilon_4 \, \forall \, \delta \in (0, \delta_2]$$

Using the fact that queues $q_l(t)$ in set $L_0(t)$ are equal, the above inequality can be rewritten as follows. For all $i \in L(t)$ and $j \in L_0(t)\ \ L(t)$, we have

$$\frac{q_i(t+\delta)}{\delta} > \frac{q_j(t+\delta)}{\delta} + \epsilon_4 \, \forall \, \delta \in (0, \delta_2].$$

Thus we have,

$$\min_{i \in L(t)} q_i(t+\delta) > \max_{j \in L_0(t) L(t)} q_j(t+\delta) + \epsilon_5 \, \forall \, \delta \in (0, \delta_2]. \tag{B.23}$$

From the inequalities (B.22) and (B.23), we have the following inequality, there exists $\delta_0, \delta_3 > 0$ such that for all $\delta \in [\delta_0, \delta_3]$ we have

$$\min_{i \in L(t)} q_i(n(t+\delta)) > \max_{j \in K L(t)} q_j(n(t+\delta)) + \epsilon_6. \tag{B.24}$$

From the definition of fluid limit $q_l(t)$, there exists n_0 large enough such that $\forall n > n_0$ and $\delta \in [\delta_0, \delta_3]$, we have that

$$\min_{i \in L(t)} Q_i(n(t+\delta)) > \max_{j \in K L(t)} Q_j(n(t+\delta)) + n\epsilon_7. \tag{B.25}$$

The above inequality ensures that the links in the set $L(t)$ have larger queue lengths compared to other links in the network for all the time slots in $[n(t+\delta_0), n(t+\delta_3)]$. Therefore, depending up on global channel state $GS(\tau)$, at each time slot $\tau \in [n(t+\delta_0), n(t+\delta_3)]$, GMS schedule picks a maximal schedule from the set of links $L(t)$ that are in 'ON' state. Let $Z_l^n(\tau)$ denote the scheduling decision picked by the GMS algorithm for link l at time slot τ. We thus have

$$\vec{Z}^n(\tau)|_{L(\tau)} \in M_{GS(\tau)\cap L(t),L(t)}. \tag{B.26}$$

Computing the total service provided by the GMS algorithm in time slots $[n(t+\delta_0), n(t+\delta_3)]$, we have

$$D_l^n(nt+n\delta_3) - D_l^n(nt+n\delta_0) = \int_{nt+n\delta_0}^{nt+n\delta_3} Z_l^n(\tau)d\tau.$$

Let us denote the quantity $\frac{D_l^n(nt+n\delta_3)-D_l^n(nt+n\delta_0)}{n(\delta_3-\delta_0)}$ by $\mu_l^n(t)$. From the above equality, we have that $\vec{\mu}^n(t)|_{L(t)} \in \Phi(L(t))$. As δ_0 can be made arbitrarily small, we have the result.

Appendix C

Proofs for Chapter 4

C.1 Proof of Thoerem 4.4.1

For any $\epsilon > 0$, we will show that for all arrivals inside $\frac{1-\epsilon}{d_I(G^K)}\Lambda_f$, the proposed algorithm can stabilize the system. Observe that $Q[t]$ is a Markov chain. We define a quadratic Lyapunov function, $L(Q[t])$ as follows,

$$L(Q[t]) = \sum_{n,d,h} \left(Q_{n,d,h}[t]\right)^2 . \tag{C.1}$$

Consider the drift in $L(.)$,

$$\begin{aligned}
E[\Delta L[t]|Q[t]] &:= E[L(Q[t+1]) - L(Q[t]) \mid Q[t]] \\
&= E\left[\sum_{n,d,h} \left(\left(Q_{n,d,h}[t+1]\right)^2 - \left(Q_{n,d,h}[t]\right)^2\right) |Q[t]\right] \\
&= E\left[\sum_{n,d,h} \left(\left(Q_{n,d,h}[t] + \Delta Q_{n,d,h}[t]\right)^2\right) \mid Q[t]\right] \\
&\quad - E\left[\sum_{n,d,h} \left(Q_{n,d,h}[t]\right)^2\right] \\
&= E\left[\sum_{n,d,h} \left(\left(\Delta Q_{n,d,h}[t]\right)^2\right) \mid Q[t]\right] \\
&\quad + 2E\left[\sum_{n,d,h} Q_{n,d,h}[t]\Delta Q_{n,d,h}[t] \mid Q[t]\right]
\end{aligned}$$

We will next prove that $E[\Delta L[t]|Q[t]] < 0$ whenever $maxQ_{n,d,h}[t]$ exceeds a certain fixed value, which will imply the positive recurrence of the Markov chain.

The difference $\Delta Q_{n,d,h}[t]$ is given by,

$$\Delta Q_{n,d,h}[t] := Q_{n,d,h}[t+1] - Q_{n,d,h}[t] \tag{C.2}$$

$$= A_f[t] I_{s(f)=n,d(f)=d,h_f^*[t])=h} + \nu_{n,d,h}^{in} - \nu_{n,d,h}^{out}, \tag{C.3}$$

where

$$\nu_{n,d,h}^{in} = \sum_{m:(m,n)\in L} \nu_{m,d,h+1}^{n,d,h}[t] \quad \text{and} \tag{C.4}$$

$$\nu_{n,d,h}^{out} = \sum_{i:(n,i)\in L} \nu_{n,d,h}^{i,d,h-1}[t] \tag{C.5}$$

and $\nu_{m,d,h+1}^{n,d,h}$ denotes the actual number of packets transferred on link (m,n) from queue $\{m,d,h+1\}$ in to queue $\{n,d,h\}$.

Given that $A_f[t]$ satisfies these bounds, we have that $\Delta Q_{n,d,h}[t]$ is bounded. Also using the inequalities on $\nu < \mu$, we have that

$$E[\Delta L[t])|Q[t]] \le M+$$
$$2E\left[\sum_{n,d,h} Q_{n,d,h}[t] A_f[t] I_{s(f)=n,d(f)=d,h_f^*[t])=h} \mid Q[t]\right] +$$
$$2E\left[\sum_{n,d,h} Q_{n,d,h}[t] \left(\mu_{n,d,h}^{in} - \mu_{n,d,h}^{out}\right) \mid Q[t]\right]$$

Rewriting the above quantities inside the summation, we have that

$$E\left[\Delta L[t] \mid Q[t]\right] \leq M+$$
$$2\sum_{f}\sum_{h} Q_{s(f),d(f),h}[t]\left(E\left[A_{f,h}[t] \mid Q[t]\right]\right) -$$
$$2\sum_{(m,n)}\sum_{d,h} \mu_{m,d,h+1}^{n,d,h}[t]\left(Q_{m,d,h+1}[t] - Q_{n,d,h}[t]\right)$$

Since $\vec{A}_f \in \frac{1-\epsilon}{d_I(G^K)}\Lambda_f$, we have that $\frac{d_I(G^K)}{1+\epsilon}\vec{A}_f \in \Lambda_f$. Consider the quantity $\hat{H}(d_I(G^K)\vec{A}_f)$ OPT2 defined in 5.3. Let us denote the optimizer to the below optimization by $A^*_{f,h}, \mu^*$.

$$\begin{aligned}
\hat{H}(d_I(G^K)A_f) = \quad & \min \sum_{f}\sum_{0<h<N} hA_{f,h} \\
& \text{s.t.} \;\; \sum_{f} A_{f,h} I_{s(f)=n,d(f)=d} + \mu_{n,d,h}^{in} \\
& \qquad\qquad \leq \mu_{n,d,h}^{out} \;\forall (n,d,h), \\
& \quad \mu_{m,d,h+1}^{n,d,h} = 0, \text{if}\, h < H_{n\to d}^{min}, \\
& \quad \left(\sum_{d}\sum_{h} \mu_{m,d,h}^{n,d,h-1}\right) \in CH(\Pi(G^K)), \\
& \quad \sum_{h} A_{f,h} = d_I(G^K)A_f \\
& \quad A_{f,h} \geq 0, \quad \mu_{m,d,h+1}^{n,d,h} \geq 0.
\end{aligned} \tag{C.6}$$

Note that we have $\sum_h A^*_{f,h} = d_I(G^K)A_f$. Let us now consider the above drift inequality with the term $\frac{1-\epsilon}{d_I(G^K)}\sum_{f}\sum_{0<h<N} Q_{s(f),d(f),h}A^*_{f,h}$ added and subtracted,

$$
\begin{aligned}
&E\left[\Delta L[t] \mid Q[t]\right] \leq M+ \\
&\quad 2\sum_{f}\sum_{h} Q_{s(f),d(f),h}[t]\left(E\left[A_{f,h}[t] \mid Q[t]\right] - \frac{1-\epsilon}{d_I(G^K)} A^*_{f,h}\right) \\
&\quad + \frac{2(1-\epsilon)}{d_I(G^K)} \sum_{f}\sum_{h} Q_{s(f),d(f),h}[t] A^*_{f,h} \\
&\quad - 2\sum_{(m,n)}\sum_{d,h} \mu^{n,d,h}_{m,d,h+1}[t]\left(Q_{m,d,h+1}[t] - Q_{n,d,h}[t]\right)
\end{aligned}
$$

As optimizer $A^*_{f,h}, \mu^*$ satisfies the all the constraints in the above optimization, we have that $\sum_f A^*_{f,h} I_{s(f)=n,d(f)=d} + \mu^{*,in}_{n,d,h} < \mu^{*,out}_{n,d,h}$. Using this inequality in the above drift inequality, we have that,

$$
\begin{aligned}
&E\left[\Delta L[t] \mid Q[t]\right] \leq M+ \\
&\quad 2\sum_{f}\sum_{h} Q_{s(f),d(f),h}[t]\left(E\left[A_{f,h}[t] \mid Q[t]\right] - \frac{1}{d_I(G^K)} A^*_{f,h}\right) \\
&\quad + \frac{2(1+\epsilon)}{d_I(G^K)} \sum_{(m,n)}\sum_{d,h} \mu^{*,n,d,h}_{m,d,h+1}[t]\left(Q_{m,d,h+1}[t] - Q_{n,d,h}[t]\right) \\
&\quad - 2\sum_{(m,n)}\sum_{d,h} \mu^{n,d,h}_{m,d,h+1}[t]\left(Q_{m,d,h+1}[t] - Q_{n,d,h}[t]\right)
\end{aligned}
$$

We can show that (similar to [52]) the weight of the schedule given by the proposed Greedy SP-BP algorithm is greater than the $\frac{1}{d_I(G^K)}$ fraction of the maximum weight attainable by any other schedule in that time slot. Thus, using this fact, we have that

$$
\begin{aligned}
&E\left[\Delta L[t] \mid Q[t]\right] \leq M+ \\
&\quad 2\sum_{f}\sum_{h} Q_{s(f),d(f),h}[t]\left(E\left[A_{f,h}[t] \mid Q[t]\right] - \frac{1}{d_I(G^K)}A^*_{f,h}\right)
\end{aligned}
$$

Further, the above inequality can be reduced to,

$$
\begin{aligned}
&E\left[\Delta L[t] \mid Q[t]\right] \leq M+ \\
&\quad 2\sum_{f}\sum_{h} Q_{s(f),d(f),h}[t]\left(E\left[A_{f,h}[t] \mid Q[t]\right] - \frac{1}{d_I(G^K)}A^*_{f,h}\right)
\end{aligned}
$$

We thus have the result.

C.2 Proof of Theorem 4.4.2

We define a quadratic Lyapunov function, $L\left(Q[t]\right)$ as follows,

$$
L\left(Q[t]\right) = \sum_{n,d,h}\left(Q_{n,d,h}[t]\right)^2. \tag{C.7}
$$

Similar to the analysis of theorem 1, we have the below bound on the drift,

$$
\begin{aligned}
&E\left[\Delta L[t] \mid Q[t]\right] \leq M + \\
&\quad 2\sum_{f}\sum_{h} Q_{s(f),d(f),h}[t]\left(E\left[A_{f,h}[t] \mid Q[t]\right]\right) - \\
&\quad 2\sum_{(m,n)}\sum_{d,h} \mu_{m,d,h+1}^{n,d,h}[t]\left(Q_{m,d,h+1}[t] - Q_{n,d,h}[t]\right)
\end{aligned}
$$

Let $\hat{A}_{f,h}$ and $\hat{\mu}_{n,d,h}$ denote the optimal solution to the optimization problem in OPT1. Adding and subtracting the quantity $\sum_{f,h}\hat{A}_{f,h}$ to the drift, we have the following,

$$
\begin{aligned}
&E\left[\Delta L[t] \mid Q[t]\right] \leq M + \\
&\quad 2\sum_{f}\sum_{h} Q_{s(f),d(f),h}[t]\left(E\left[A_{f,h}[t] - \hat{A}_{f,h} \mid Q[t]\right]\right) - \\
&\quad 2\sum_{(m,n)}\sum_{d,h} \mu_{m,d,h+1}^{n,d,h}[t]\left(Q_{m,d,h+1}[t] - Q_{n,d,h}[t]\right) + \\
&\quad 2\sum_{f}\sum_{h} Q_{s(f),d(f),h}[t]\left(\hat{A}_{f,h}\right)
\end{aligned}
$$

Since $\hat{A}_{f,h}$ and $\hat{\mu}_{n,d,h}$ is a feasible point in optimization OPT1, we have that

$$
\sum_{f}\hat{A}_{f,h} I_{s(f)=n,d(f)=d} + \sum_{m:(m,n)\in L}\hat{\mu}_{m,d,h+1}^{n,d,h} \leq \sum_{i:(n,i)\in L}\hat{\mu}_{n,d,h}^{i,d,h-1}
$$

Multiplying both sides by $Q_{n,d,h}$ and taking the summation over n, d, h

on both sides of the above inequality, we have that

$$\sum_{n,d,h} Q_{n,d,h} \sum_{f} \hat{A}_{f,h} I_{s(f)=n,d(f)=d} \leq$$
$$\sum_{n,d,h} Q_{n,d,h} \left(\sum_{i:(n,i)\in L} \hat{\mu}_{n,d,h}^{i,d,h-1} - \sum_{m:(m,n)\in L} \hat{\mu}_{m,d,h+1}^{n,d,h} \right)$$

Rearranging the summations and rewriting the above inequality, we have the following,

$$\sum_{f} \sum_{h} Q_{s(f),d(f),h} \hat{A}_{f,h} \leq$$
$$\sum_{(m,n)\in L} \sum_{d,h} \hat{\mu}_{m,d,h+1}^{n,d,h} \left(Q_{m,d,h+1} - Q_{n,d,h} \right)$$

Using the above inequality, we have the drift to be bounded by,

$$E\left[\Delta L[t] \mid Q[t]\right] \leq M+$$
$$2 \sum_{f} \sum_{h} Q_{s(f),d(f),h}[t] \left(E\left[A_{f,h}[t] - \hat{A}_{f,h} \mid Q[t] \right] \right) -$$
$$2 \sum_{(m,n)} \sum_{d,h} \mu_{m,d,h+1}^{n,d,h}[t] \left(Q_{m,d,h+1}[t] - Q_{n,d,h}[t] \right) +$$
$$2 \sum_{(m,n)\in L} \sum_{d,h} \hat{\mu}_{m,d,h+1}^{n,d,h} \left(Q_{m,d,h+1}[t] - Q_{n,d,h[t]} \right)$$

It can be shown that the service vector found using Greedy SP-BP, $\mu[t]$, satisfies the below inequality, $\forall \eta \in \Pi(G^K)$,

$$\sum_{(m,n)} \sum_{d,h} \mu_{m,d,h+1}^{n,d,h}[t] \left(Q_{m,d,h+1}[t] - Q_{n,d,h}[t] \right) \geq$$
$$\frac{1}{d_G} \sum_{(m,n)\in L} \sum_{d,h} \eta_{m,d,h+1}^{n,d,h} \left(Q_{m,d,h+1}[t] - Q_{n,d,h[t]} \right).$$

As $\hat{\mu}$ is the optimizer for OPT1, we have that $\hat{\mu} \in \frac{1}{d_I(G)} CH(\Pi(G^K))$ (i.e $d_I(G)\hat{\mu} \in CH\Pi(G^K)$), and using the above inequality, we have that

$$\sum_{(m,n)} \sum_{d,h} \mu_{m,d,h+1}^{n,d,h}[t] \left(Q_{m,d,h+1}[t] - Q_{n,d,h}[t] \right) \geq$$
$$\sum_{(m,n)\in L} \sum_{d,h} d_I(G) \hat{\mu}_{m,d,h+1}^{n,d,h} \left(Q_{m,d,h+1}[t] - Q_{n,d,h[t]} \right)$$

Using the above inequality, we have the drift in the Lyapunov function bounded by,

$$E\left[\Delta L[t] \mid Q[t]\right] \leq M +$$
$$2 \sum_{f} \sum_{h} Q_{s(f),d(f),h}[t] \left(E\left[A_{f,h}[t] - \hat{A}_{f,h} \mid Q[t] \right] \right)$$

Observe that our proposed algorithm has the following rate-splitting property,

$$\sum_{f} \sum_{h} \left(Q_{s(f),d(f),h}[t] + \beta h \right) E\left[A_{f,h}[t] \mid Q[t] \right] \leq$$
$$\sum_{f} \sum_{h} \left(Q_{s(f),d(f),h}[t] + \beta h \right) \hat{A}_{f,h}$$

Substituting the above inequality in to drift inequality, we have that

$$E\left[\Delta L[t] \mid Q[t]\right] \leq M +$$
$$2\beta \sum_{f} \sum_{h} \left(h\hat{A}_{f,h} - hE\left[A_{f,h}[t] \mid Q[t] \right] \right)$$

Taking one more expectation on both sides, we have the following

$$E\left[\Delta L[t]\right] \leq M +$$
$$2\beta \sum_{f} \sum_{h} \left(h\hat{A}_{f,h} - hE\left[A_{f,h}[t] \right] \right)$$

Summing the above inequality over time, we have the following inequality

$$\frac{1}{T}\sum_{t=0}^{T-1} E\left[\Delta L[t]\right] \leq M+ \\ 2\beta\sum_{f}\sum_{h} h\hat{A}_{f,h} - 2\beta\frac{1}{T}\sum_{t=0}^{T-1}\sum_{f}\sum_{h} hE\left[A_{f,h}[t]\right]$$

Further, by rearranging the above inequality, we have the following,

$$\frac{1}{T}\sum_{t=0}^{T-1}\sum_{f}\sum_{h} hE\left[A_{f,h}[t]\right] \leq \frac{M_1}{\beta}+ \\ \sum_{f}\sum_{h} h\hat{A}_{f,h}$$

Thus, for any $\epsilon > 0$, there exists large enough β such that

$$\sum_{f}\sum_{h} hA_{f,h}^{\text{Greedy SP-BP}} \leq \sum_{f}\sum_{h} h\hat{A}_{f,h} + \epsilon. \tag{C.8}$$

Hence we have the result that the average hop-delay of our proposed algorithm is upper bounded by $H^{UB}(A_f, \frac{1}{d_I(G)})$.

C.3 Proof of Theorem 4.4.3

(Lower Bound) Note that optimization used to calculate the quantities $\hat{H}(\gamma A_f)$ and $H^{UB}(\gamma A_f, \gamma)$ have the same objective and same constraints except one. Observing the fact that the every feasible point for OPT1 is also a feasible point for OPT2. We have that ,

$$\hat{H}(\gamma A_f) \leq H^{UB}(\gamma A_f, \gamma).$$

(Upper Bound) Let $\{A^*_{f,h}, \mu^*\}$ denote the optimizer for the convex problem OPT2. It can be shown that the scaled point $\gamma * (\{A^*_{f,h}, \mu^*\})$ satisfies all the constraints of the optimization problem to find $H^{UB}(\gamma A_f, \gamma)$, and therefore a feasible point in the search space of optimization problem of $H^{UB}(\gamma A_f, \gamma)$. We thus have the following upper bound,

$$H^{UB}(\gamma A_f, \gamma) \leq \sum_f \sum_{0<h<N} h\gamma A^*_{f,h}.$$

As $\{A^*_{f,h}, \mu^*\}$ was the optimizer for OPT2, we have that

$$\hat{H}(A_f) = \sum_f \sum_{0<h<N} hA^*_{f,h}$$

Using the above equality, we have that

$$H^{UB}(\gamma A_f, \gamma) \leq \gamma \hat{H}(A_f),$$

Appendix D

Proofs for Chapter 5

D.1 Proof of Theorem 5.3.1

For any given $\epsilon > 0$, we will show that for all the arrival rates that belong to $(1-\epsilon)\Lambda$, the proposed algorithm - I can stabilize the network. Let us define a quadratic Lyapunov function, $L(\vec{Q}[t])$ as follows,

$$L(\vec{Q}[t]) = \sum_{l} (Q_l[t])^2 + \sum_{l,i,j} (Q_l^{H_i,H_j}[t])^2 \tag{D.1}$$

Consider the drift in the Lyapunov function, $\Delta L(.)$,

$$\begin{aligned}
E\left[\Delta L(\vec{Q}[t])|\vec{Q}[t]\right] &:= E\left[L(\vec{Q}[t+1]) - L(\vec{Q}[t])\middle|\vec{Q}[t]\right] \\
&= E\left[\sum_{l} \left((Q_l[t+1])^2 - (Q_l[t])^2\right)\middle|\vec{Q}[t]\right] \\
&\quad + E\left[\sum_{l,i,j} \left((Q_l^{H_i,H_j}[t+1])^2 - (Q_l^{H_i,H_j}[t])^2\right)\middle|\vec{Q}[t]\right] \\
&\le M + E\left[\sum_{l} (2Q_l[t]\Delta Q_l[t])\middle|\vec{Q}[t]\right] \\
&\quad + E\left[\sum_{l,i,j} \left(2Q_l^{H_i,H_j}[t]\Delta Q_l^{H_i,H_j}[t]\right)\middle|\vec{Q}[t]\right],
\end{aligned}$$

where the last inequality follows from our assumption that the number of arrivals and departures in any time slot are bounded. Using the definition of conditional expectation $E[X] = \sum_y P(Y = y)E[X|Y = y]$, we have the following,

$$
\begin{aligned}
&E\left[\Delta L(\vec{Q}[t])|\vec{Q}[t]\right] \leq M \\
&\quad + \sum_k \pi(H_k) E\left[\sum_l (2Q_l[t]\Delta Q_l[t]) \middle| \vec{Q}[t], H[t] = H_k\right] \\
&\quad + \sum_k \pi(H_k) E\left[\sum_{l,i,j} \left(2Q_l^{H_i,H_j}[t]\Delta Q_l^{H_i,H_j}[t]\right) \middle| \vec{Q}[t], H_k\right].
\end{aligned}
$$

Let us define $W^{alg}(\vec{Q}[t], H[t]) := \max\{W_{is}[t], W_{ia}[t]\}$. Using this definition, we can rewrite the above inequality as follows,

$$
\begin{aligned}
&E\left[\Delta L(\vec{Q}[t])|\vec{Q}[t]\right] \leq M \\
&\quad + 2\sum_l Q_l[t]\lambda_l - 2\sum_k \pi(H_k) W^{alg}(\vec{Q}[t], H_k).
\end{aligned}
$$

Since arrival rate vector lies inside Λ (i.e., $\exists\, \epsilon > 0$ such that $\vec{\lambda} \in (1-\epsilon)\Lambda$), we can find the pair $\left\{f(H_i, H_j), f(H_i), \alpha(S, H_i)\right\}$ such that the conditions in Theorem 1 hold. Utilizing the conditions, we have

$$
\begin{aligned}
E\left[\Delta L(\vec{Q}[t])|\vec{Q}[t]\right] &\leq M - \epsilon \sum_l Q_l[t] \\
&+ 2\sum_l Q_l[t]\left(\sum f(H_i, H_j) R_l(H_i, H_j)\right) \\
&+ 2\sum_l Q_l[t]\left(\sum f(H_i) \sum_S \alpha(S, H_i) I_{l\in S} R_l(H_i)\right) \\
&- 2\sum_k \pi(H_k) W^{alg}(\vec{Q}[t], H_k).
\end{aligned}
$$

Using the inequality that relates the $\pi(H_k)$ and $f(.,.)$, we have the following,

$$
\begin{aligned}
E\left[\Delta L(\vec{Q}[t])|\vec{Q}[t]\right] &\leq M - \epsilon \sum_l Q_l[t] \\
&+ 2\sum_l Q_l[t]\left(\sum f(H_i, H_j) R_l(H_i, H_j)\right) \\
&+ 2\sum_l Q_l[t]\left(\sum f(H_i) \sum_S \alpha(S, H_i) I_{l\in S} R_l(H_i)\right) \\
&- 2\sum_k \left(\sum_j f(H_k, H_j) + \sum_j f(H_j, H_k) + f(H_k)\right) \times \\
&W^{alg}(\vec{Q}[t], H_k).
\end{aligned}
$$

Rewriting the above expression, we have

$$E\left[\Delta L(\vec{Q}[t])|\vec{Q}[t]\right] \leq M - \epsilon \sum_l Q_l[t]$$
$$+ 2\sum_{i,j} f(H_i, H_j)\left(\sum_l Q_l[t] R_l(H_i, H_j) - W^{alg}(\vec{Q}[t], H_i) - W^{alg}(\vec{Q}[t], H_j)\right)$$
$$+ 2\sum_i f(H_i)\left(\sum_S \alpha(S, H_i)\sum_{l \in S} Q_l[t] R_l(H_i) - W^{alg}(\vec{Q}[t], H_i)\right)$$

Using the fact that $Q_l[t] \leq \left(Q_l[t] - Q_l^{H_i,H,j}\right)^+ + Q_l^{H_i,H_j}$, we have the following,

$$E\left[\Delta L(\vec{Q}[t])|\vec{Q}[t]\right] \leq M - \epsilon \sum_l Q_l[t]$$
$$+ 2\sum_{i,j} f(H_i, H_j)\left(\sum_l \left(Q_l[t] - Q_l^{H_i,H_j}\right)^+ R_l(H_i, H_j) - W^{alg}(\vec{Q}[t], H_i)\right)$$
$$+ 2\sum_{i,j} f(H_i, H_j)\left(\sum_l Q_l^{H_i,H_j}[t] R_l(H_i, H_j) - W^{alg}(\vec{Q}[t], H_j)\right)$$
$$+ 2\sum_i f(H_i)\left(\sum_S \alpha(S, H_i)\sum_{l \in S} Q_l[t] R_l(H_i) - W^{alg}(\vec{Q}[t], H_i)\right)$$

Since the last three quantities in the above inequality are negative for the proposed algorithm, we have that

$$E\left[\Delta L(\vec{Q}[t])|\vec{Q}[t]\right] \leq M - \epsilon \sum_l Q_l[t]$$

Thus, using the Foster-Lyapunov condition we have that the Markov chain $\vec{Q}$ is positive recurrent.

Bibliography

[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.

[31] B. Birand, M. Chudnovsky, B. Ries, P. Seymour, G. Zussman, and Y. Zwols, " Analyzing the performance of greedy maximal scheduling via local pooling and graph theory," In *Proc. IEEE Infocom.*, San Diego, California, March 2010.

[32] J. G. Dai, " On positive harris recurrence of multiclass queueing networks: A unified approach via fluid limit models," *The Annals of Applied Probability*, 5(1):49–77, 1995.

[33] A. Dimakis and J. Walrand, " Sufficient conditions for stability of longest-queue-first scheduling: Second-order properties using fluid limits," *Advances in Applied Probability*, 38(2):505–521, 2006.

[34] L. Georgiadis, M.J. Neely, and L. Tassiulas, " Resource allocation and cross-layer control in wireless networks," *Foundations and Trends in Networking*, 1(1):1–144, 2006.

[35] C. Joo, X. Lin, and N. B. Shroff, " Understanding the capacity region of the greedy maximal scheduling algorithm in multi-hop wireless networks," *IEEE/ACM Transactions on Networking*, 17(4):1132–1145, August 2009.

[36] C. Joo and N. B. Shroff, " Performance of random access scheduling schemes in multihop wireless networks," In *Proc. IEEE Infocom.*, 2007.

[37] L. Bao Le, E. Modiano, C. Joo, and N. B. Shroff, " Longest-queue-first scheduling under sinr interference model," In *Proceedings of the eleventh ACM international symposium on Mobile ad hoc networking and computing*, MobiHoc '10, pages 41–50, 2010.

[38] B. Li, C. Boyaci, and Y. Xia, " A refined performance characterization of longest-queue-first policy in wireless networks," In *Proceedings of the tenth ACM international symposium on Mobile ad hoc networking and computing*, MobiHoc '09, pages 65–74, New York, NY, USA, 2009.

[39] Q. Li and R. Negi, " Greedy maximal scheduling in wireless networks," In *GLOBECOM'10*, pages 1–5, 2010.

[40] X. Lin and S. Rasool, " Constant-time distributed scheduling policies for ad hocwireless networks," In *Proc. Conf. on Decision and Control*, 2006.

[41] A. Reddy, S. Sanghavi, and S. Shakkottai, " On the effect of channel fading on greedy scheduling," In *Proceedings of IEEE Infocom*, Orlando, FL, 2012.

[42] 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.*, 4:1936–1948, December 1992.

[43] G. Zussman and A. Brzezinski; E. Modiano;, " Multihop local pooling for distributed throughput maximization in wireless networks," In *Proc. IEEE Infocom.*, Phoenix, Arizona, April 2008.

[44] M. Jain, J. I. Choi, T. Kim, D. Bharadia, S. Seth, K. Srinivasan, P. Levis, S. Katti, and P. Sinha, " Achieving single channel, full duplex wireless communication," In *ACM Int. Conf. on Mobile Computing and Networking (MobiCom)*, New York, USA, 2011.

[45] L. X. Bui, R. Srikant, and A. Stolyar, " A novel architecture for reduction of delay and queueing structure complexity in the back-pressure algorithm," *IEEE/ACM Transactions on Networking*, 19:1597–1609, December 2011.

[46] J. I. Choi, M. Jain, K. Srinivasan, P. Levis, and S. Katti, " Achieving single channel, full duplex wireless communication," In *ACM Int. Conf. on Mobile Computing and Networking (MobiCom)*, New York, USA, 2010.

[47] W. J. Dally and C. L. Seitz, "The torus routing chip," *Distributed Computing*, 1986.

[48] A. Gupta, X. Lin, and R. Srikant, "Low-complexity distributed scheduling algorithms for wireless networks," In *Proc. IEEE Infocom.*, 2007.

[49] M. Jain, J. I. Choi, K. Srinivasan, S. Seth, P. Levis, and S. Katti, "Single channel, full-duplex wireless," In *Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems*, pages 357–358, New York, NY, USA, 2011.

[50] L. Jiang and J. Walrand, "A distributed csma algorithm for throughput and utility maximization in wireless networks," In *Proc. Ann. Allerton Conf. Communication, Control and Computing*, 2008.

[51] C. Joo, X. Lin, and N. B. Shroff, "Understanding the capacity region of the greedy maximal scheduling algorithm in multi-hop wireless networks," *IEEE/ACM Transactions on Networking*, 17(4):1132–1145, August 2009.

[52] M. Leconte, J. Ni, and R. Srikant, "Improved bounds on the throughput efficiency of greedy maximal scheduling in wireless networks," In *Proceedings of the tenth ACM international symposium on Mobile ad hoc networking and computing*, MobiHoc '09, 2009.

[53] L. Lin, X. Lin, and N. Shroff, " Low-complexity and distributed energy minimization in multi-hop wireless networks," In *Proc. IEEE Infocom.*, 2007.

[54] N. McKeown, " *Scheduling Algorithms for Input-Queued Cell Switches*," PhD Thesis, University of California at Berkeley,, Berkeley, CA, 1995.

[55] K. Miller, A. Sanne, K. Srinivasan, and S. Vishwanath, “ Enabling real-time interference alignment: promises and challenges,” In *Proceedings of the thirteenth ACM international symposium on Mobile Ad Hoc Networking and Computing*, MobiHoc ’12, New York, NY, USA, 2012.

[56] S. Rajagopalan, J. Shin, and D. Shah, “ Network adiabatic theorem: An efficient randomized protocol for contention resolution,” In *Proc. Ann. ACM SIGMETRICS Conf.*, 2009.

[57] A. Reddy, S. Sanghavi, and S. Shakkottai, “ On the effect of channel fading on greedy scheduling,” In *Proceedings of IEEE Infocom*, Orlando, FL, 2012.

[58] A. Sahai, G. Patel, and A. Sabharwal, “ Pushing the limits of full-duplex: Design and real-time implementation,” *CoRR*, abs/1107.0607, 2011.

[59] X. Wu and R. Srikant, “ Regulated maximal matching: a distributed scheduling algorithm for multi-hop wireless networks with node-exclusive spectrum sharing,” In *Proc. Conf. on Decision and Control*, 2005.

[60] X. Wu and R. Srikant, “ Bounds on the capacity region of multi-hop wireless networks under distributed greedy scheduling,” In *Proc. IEEE Infocom.*, 2006.

[61] L. Ying, S. Shakkottai, A. Reddy, and S. Liu, “ On combining shortest-path and back-pressure routing over multihop wireless networks,” *IEEE/ACM Transactions on Networking*, 19(3):841–854, 2011.

[62] G. Zussman, A. Brzezinski, and E. Modiano, " Multihop local pooling for distributed throughput maximization in wireless networks," In *Proc. IEEE Infocom.*, Phoenix, Arizona, April 2008.

[63] M. Maddah-Ali, A. Motahari, and A. Khandani, "Communication over mimo x channels: interference alignment, decomposition, and performance analysis," *IEEE Trans. Inform. Theory*, vol. 54, pp. 3893–3908, 2008.

[64] B. Nazer, M. Gastapar, S. A. Jafar, and S. Vishwanath, "Ergodic interference alignment," in *Proc. IEEE Int. Symp. Information Theory (ISIT)*, 2009.

[65] V. R. Cadambe and S. A. Jafar, "Interference alignment and degrees of freedom of the k-user interference channel," *IEEE Trans. Inform. Theory*, vol. 54, pp. 3425–3441, 2008.

[66] S. W. Jeon and M. Gastpar, "A survey on interference networks: interference alignment and neutralization," in *Entropy*, sep. 2012.

[67] L. Georgiadis, M. Neely, and L. Tassiulas, "Resource allocation and cross-layer control in wireless networks," *Foundations and Trends in Networking*, vol. 1, no. 1, pp. 1–144, 2006.

[68] C. Geng and S. A. Jafar, "On optimal ergodic interference alignment," in *Proceedings of Globecom*, 2012.

[69] T. Gou, S. A. Jafar, C. Wang, S. W. Jeon, and S. Y. Chung, "Aligned interference neutralization and degrees of freedom of the 2 x 2 x 2 interference channel," *IEEE Trans. Inform. Theory*, vol. 58, pp. 4381–4395, 2012.

[70] L. Bui, R. Srikant and A. Stolyar, "Novel Architectures and Algorithms for Delay Reduction in Back-pressure Scheduling and Routing," *Proc. of IEEE Infocom*, 2009.

Vita

Akula Aneesh Reddy received the Bachelor of Technology degree in Electrical Engineering from the Indian Institute of Technology (IIT) Madras in May 2006 and M. S. E. degree in Electrical and Computer Engineering from The University of Texas at Austin in December 2008. He was awarded Young Engineering Fellowship from Indian Institute of Science (IISc) Bangalore in July 2005. He worked as summer intern at Qualcomm Corp. R&D in 2009 and at Texas Instruments in 2010. He is currently pursuing a Ph.D. under the supervision of Dr. Sanjay Shakkottai.

Permanent address: 3453 Lake Asutin Blvd
Austin, Texas 78703

This dissertation was typeset with LaTeX† by the author.

†LaTeX is a document preparation system developed by Leslie Lamport as a special version of Donald Knuth's TeX Program.